

UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação



Dissertação

**Reconhecimento de Atividades em Casas Inteligentes: Uma abordagem não
Intrusiva explorando Processamento Semântico**

Eduardo Soares de Abreu

Pelotas, 2017

Eduardo Soares de Abreu

Reconhecimento de Atividades em Casas Inteligentes: Uma abordagem não Intrusiva explorando Processamento Semântico

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Ciência da Computação

Orientador: Prof. Dr. Adenauer Corrêa Yamin
Coorientador: Prof. Dr. Ana Marilza Pernas Fleischmann

Pelotas, 2017

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

A162r Abreu, Eduardo Soares de

Reconhecimento de atividades em casas inteligentes :
uma abordagem não intrusiva explorando processamento
semântico / Eduardo Soares de Abreu ; Adenauer Corrêa
Yamin, orientador ; Ana Marilza Pernas Fleischmann,
coorientadora. — Pelotas, 2017.

88 f.

Dissertação (Mestrado) — Programa de Pós-Graduação
em Computação, Centro de Desenvolvimento Tecnológico,
Universidade Federal de Pelotas, 2017.

1. Reconhecimento de atividades. 2. Processamento
semântico. 3. Ontologia. 4. Ciência de contexto. I. Yamin,
Adenauer Corrêa, orient. II. Fleischmann, Ana Marilza
Pernas, coorient. III. Título.

CDD : 005

**Dedico... este trabalho a minha família e minha namorada Mirian,
companheira de todas horas.**

AGRADECIMENTOS

Agradeço à minha família pelo apoio e de minha namorada Mirian por todo carinho, paciência, incentivo. Sem vocês eu não chegaria ao fim desta jornada.

Agradeço também ao meu orientador, Adenauer Corrêa Yamin, por todo apoio, paciência e ensinamentos recebidos, permitindo completar este trabalho.

A minha coorientadora Ana Marilza Pernas Fleischmann, por todas as revisões, sugestões e ensinamentos.

A colega Andrea Gonzales, que me recebeu de braços abertos, sendo uma grande amiga na hora dos estudos e dos trabalhos.

Ao Renato Dilli, pela sua ajuda incansável em várias etapas desse trabalho.

Aos colegas de laboratório Patrícia, Beto, Jeronimo e Rodrigo pela atenção e auxílio.

**Não é bom ficar furioso se você empacou [em uma ideia].
O que eu faço é continuar pensando sobre o problema.
Mas trabalhando em outra coisa.**

— STEPHEN HAWKING

RESUMO

ABREU, Eduardo Soares de. **Reconhecimento de Atividades em Casas Inteligentes: Uma abordagem não Intrusiva explorando Processamento Semântico**. 2017. 88 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2017.

Nos últimos anos, as técnicas de Reconhecimento de Atividades têm atraído atenção crescente. Entre as muitas aplicações, um interesse especial está no domínio ubíquo da e-Health, onde o reconhecimento automático de atividades é usado em sistemas de reabilitação, gerenciamento de doenças crônicas, monitoramento de idosos, bem como em aplicações de bem-estar pessoal.

Esse interesse tem tido como motivação o envelhecimento da população. Este envelhecimento resulta em significativos desafios socioeconômicos no setor da saúde pública, bem como na incidência de doenças crônico-degenerativas, sendo a demência uma das mais preocupantes.

Uma alternativa que vem sendo amplamente proposta é a utilização de casas inteligentes (ambiente assistido de vivência), nos quais as residências das pessoas sob tratamento deverão contemplar serviços computacionais que possam auxiliá-las nas suas práticas diárias, de forma o mais transparente possível.

Este trabalho tem como acrônimo EXEHDA-AR (*EXEHDA-Activity Recognition*), e seu objetivo principal é contribuir com o Subsistema de Reconhecimento de Contexto e Adaptação do *middleware* EXEHDA (*Execution Environment for Highly Distributed Applications*) capacitando-o para o atendimento das demandas do Reconhecimento de Atividades, explorando para isto uma abordagem baseada em Ciência de Contexto.

O EXEHDA-AR explora processamento semântico para prover Reconhecimento de Atividades em Casas Inteligentes, para tanto foram propostos componentes a serem integrados ao *middleware* EXEHDA, bem como um modelo ontológico. Os dados de contexto coletados são agrupados segundo o conceito de janela de tempo deslizante.

Quando avaliado o EXEHDA-AR obteve uma acurácia média de 94,36% no Reconhecimento de Atividades. Estes resultados apontam que métodos baseados em processamento semântico constituem uma alternativa viável para o Reconhecimento de Atividades com baixo nível de intrusão, indicando a continuidade dos esforços de pesquisa.

Palavras-chave: Reconhecimento de Atividades, Processamento Semântico, Ontologia, Ciência de Contexto.

ABSTRACT

ABREU, Eduardo Soares de. **Recognition Activities in Smart Home: A Non-Intrusive Approach Exploring Semantic Processing**. 2017. 88 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2017.

In recent years, the techniques of Activity Recognition have attracted increasing attention. Among the many applications, special interest lies in the ubiquitous domain of e-Health, where automatic activity recognition is used in rehabilitation systems, chronic disease management, elderly monitoring, as well as in personal wellness applications.

This interest has been motivated by the aging of the population. This aging results in significant socioeconomic challenges in the public health sector, as well as in the incidence of chronic-degenerative diseases, with dementia being one of the most worrying.

An alternative that has been widely proposed is the use of intelligent houses (ambient assisted living), in which the residences of people under treatment should contemplate computational services that can assist them in their daily practices, in the most transparent way possible.

This work has the acronym EXEHDA-AR (EXEHDA-Activity Recognition), and its main objective is to contribute with the Context Recognition and Adaptation Subsystem of the EXEHDA (Execution Environment for Highly Distributed Applications) middleware, enabling it to meet the demands Of Activity Recognition, using a Context Awareness approach.

The EXEHDA-AR explores semantic processing to provide Activity Recognition in smart home, for which we have proposed components to be integrated with the EXEHDA middleware as well as an ontological model. The collected context data are grouped according to the concept of time-sliding window.

When evaluated the EXEHDA-AR obtained an average accuracy of 94.36% in the Recognition of Activities. These results indicate that methods based on semantic processing constitute a viable alternative for the Recognition of Activities with low level of intrusion, indicating the continuity of research efforts.

Keywords: Recognition Activity, Semantic Processing, Ontology, Context Awareness.

LISTA DE FIGURAS

Figura 1	Visão geral das diferentes camadas de interpretação de contexto .	23
Figura 2	Exemplo da modelagem CML	24
Figura 3	Visão geral PalSPOT	33
Figura 4	Visão geral CONON	34
Figura 5	Representação das três camadas	38
Figura 6	Visão das classes, relacionamentos e propriedade de dados da ontologia	39
Figura 7	Visão geral do framework	40
Figura 8	Piramide das ontologias utilizadas	40
Figura 9	Visão geral da arquitetura do sistema (OBAR)	42
Figura 10	Ontologia de infraestrutura de contexto (OBAR)	43
Figura 11	Linguagem natural regra varrer	43
Figura 12	Regra da atividade varrer na linguagem da API JENA	44
Figura 13	Relacionamento entre as diferentes camadas	46
Figura 14	Um gráfico RDF descrevendo um sensor de contato de concreto . .	47
Figura 15	Fragmento da ontologia	47
Figura 16	Ambiente Ubíquo gerenciado pelo EXEHDA	49
Figura 17	Visão geral da proposta do EXEHDA-AR	50
Figura 18	Visão geral da arquitetura do EXEHDA Servidor de Contexto expandido	51
Figura 19	Representação de janela de tempo deslizante	53
Figura 20	Visão geral do Módulo de Processamento do EXEHDA-AR	54
Figura 21	Visão geral da ontologia concebida	55
Figura 22	Classes e atributos da ontologia proposta	56
Figura 23	Concepção de regras para o EXEHDA-AR	56
Figura 24	Instância criada quando uma atividade é reconhecida	57
Figura 25	Servidor de Borda	62
Figura 26	Comparação entre datasets produzidos por casas inteligentes . . .	64
Figura 27	Distribuição dos sensores casa Aruba	65
Figura 28	Distribuição dos sensores casa Aruba	65
Figura 29	Visão da casa Aruba em 3D	66
Figura 30	Destacando o sensores do quarto de Dormir	67
Figura 31	Regra Dormir SWRL	67
Figura 32	Consulta SPARQL para atividade Dormir	68
Figura 33	Resultado da consulta atividade Dormir SPARQL	68

Figura 34	Destacando os sensores da atividade Cama para Banheiro	69
Figura 35	Regra Cama para Banheiro SWRL	69
Figura 36	Consulta SPARQL da atividade Cama para Banheiro	70
Figura 37	Resultado da consulta SPARQL da atividade Cama para Banheiro .	70
Figura 38	Sensores de interesse Entrando em Casa	71
Figura 39	Regra da atividade Entrando em Casa	71
Figura 40	Consulta SPARQL da atividade Entrando em Casa	72
Figura 41	Resultado da consulta SPARQL da atividade Entrando em Casa . .	72
Figura 42	Regra Saindo de Casa SWRL	73
Figura 43	Consulta SPARQL da atividade Sair de Casa	73
Figura 44	Resultado da consulta SPARQL da atividade Sair de Casa	73
Figura 45	Sensores do Escritório	74
Figura 46	Regra Trabalhando no Escritório em SWRL	74
Figura 47	Consulta SPARQL atividade Trabalhando no Escritório.	75
Figura 48	Resultado da consulta SPARQL atividade Trabalhando no Escritório	75
Figura 49	Matriz de confusão	76
Figura 50	Visão geral do método	76
Figura 51	Matriz de confusão do EXEHDA-AR	77

LISTA DE TABELAS

Tabela 1	Comparações entre os três tipos de modelagem	30
Tabela 2	Comparação dos Trabalhos Relacionados	60

LISTA DE ABREVIATURAS E SIGLAS

AAL	<i>Ambient Assisted Living</i>
AALISABETH	<i>Ambient-Aware LIfeStyle tutor, Aiming BETter Health</i>
ADL	<i>Activity Daily Live</i>
API	<i>Application Programming Interface</i>
CEP	<i>Complex Event Processing</i>
CML	<i>Context Modelling Language</i>
CONON	<i>CONtext ONtology</i>
CRF	<i>Conditional Random Field</i>
DB	<i>Data Base</i>
EXEHDA	<i>Execution Environment for Highly Distributed Applications</i>
EXEHDA-AR	<i>EXEHDA – AR Activity Recognition</i>
GPS	<i>Global Positioning System</i>
HMM	<i>Hidden Markov Model</i>
LoT	<i>Lab of Things</i>
LUPS	<i>Laboratory of Ubiquitous and Parallel Systems</i>
NB	<i>Network Bayesian</i>
OBAR	<i>Ontology Based Activity Recognition</i>
OntSit	<i>Ontology Situation</i>
ORM	<i>Object-Role Modeling</i>
OWL	<i>Web Ontology Language</i>
PiVOn	<i>Pervasive Information Visualization Ontology</i>
RFID	<i>Radio-Frequency IDentification</i>

SME	<i>Simple Model Event</i>
SPARQL	<i>SPARQL Protocol and RDF Query Language</i>
SWRL	<i>Semantic Web Rule Language</i>
UbiComp	Computação Ubíqua

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Motivação e Objetivos	16
1.2	Estrutura do Texto	18
2	FUNDAMENTOS CONCEITUAIS	19
2.1	Ciência de Contexto na UbiComp	19
2.1.1	Principais Conceitos	20
2.1.2	Modelagem de Contexto	23
2.2	Reconhecimento de Atividades	30
2.3	Métodos Baseados em Aprendizagem	31
2.4	Ontologia para Reconhecimento de Atividades	32
2.5	Considerações sobre o Capítulo	35
3	TRABALHOS RELACIONADOS	37
3.1	A Context-Aware System Infrastructure for Monitoring	37
3.2	AAL Domain Ontology for Event-based Human Activity Recognition	39
3.3	Activity Recognition using Context-Aware Infrastructure Ontology in Smart Home Domain	41
3.4	Semantic Smart Homes: Towards Knowledge Rich Assisted Living Environments	44
3.5	Considerações sobre o Capítulo	47
4	EXEHDA-AR: ARQUITETURA E FUNCIONALIDADES	48
4.1	EXEHDA-AR: Visão Geral da Proposta	48
4.2	Servidor de Contexto do EXEHDA na visão EXEHDA-AR	50
4.2.1	Módulo de Aquisição no EXEHDA-AR	52
4.2.2	Módulo de Processamento no EXEHDA-AR	53
4.2.3	EXEHDA-AR: Servidor de Borda	57
4.3	Discussão dos Trabalhos Relacionados	59
4.4	Considerações sobre o Capítulo	61
5	ESTUDO DE CASO	63
5.1	Datasets Considerados	63
5.2	Avaliação de Acurácia no EXEHDA-AR	76
5.3	Considerações sobre o Capítulo	77

6	CONSIDERAÇÕES FINAIS	79
6.1	Principais conclusões	79
6.2	Publicações Realizadas	80
6.3	Trabalhos Futuros	80
	REFERÊNCIAS	82

1 INTRODUÇÃO

A população mundial está passando por um processo no qual a sua idade média está aumentando. Estima-se que até o ano de 2030 o Brasil, que é considerado até então estando em uma faixa etária adulta, passará para terceira idade rapidamente, devido a redução das taxas de natalidade e ao aumento da expectativa de vida (MARIN; PANES, 2015).

Essa transição para um perfil mais envelhecido resulta em significativos desafios socioeconômicos no setor da saúde pública, bem como na incidência de doenças crônico-degenerativas, sendo a demência uma das mais preocupantes.

Uma alternativa que vem sendo amplamente proposta é a utilização de Casas Inteligentes (ambientes assistidos de vivência), nos quais as residências das pessoas sob tratamento deverão ter serviços computacionais que possam auxiliá-las nas suas práticas diárias, de forma mais transparente possível.

A perspectiva é integrar a tecnologia ao cotidiano do usuário, de modo que seja exigido o menor envolvimento possível no gerenciamento da infraestrutura computacional utilizada, caracterizando assim uma infraestrutura computacional de natureza ubíqua (LOPES et al., 2014). Este aspecto ganha significado em se tratando de usuários de uma Casa Inteligente.

As Casas Inteligentes consideram as características do usuário determinadas pelas necessidades, preferências, ou limitações de forma a proporcionarem respostas personalizadas e adequadas no espaço e no tempo para auxiliar as pessoas na realização de atividades diárias (RÖCKER; ZIEFLE; HOLZINGER, 2014).

Um dos desafios para coleta de informações contextuais em Casas Inteligentes está relacionado com a heterogeneidade decorrente da diversidade de tecnologias de *hardware* e *software* presentes neste ambiente. Esta heterogeneidade exige soluções que permitam a interoperabilidade dos vários componentes presentes no ambiente. Uma alternativa promissora para tratar a heterogeneidade é a utilização de plataformas de *middleware* (YURUR; LIU; MORENO, 2014).

Este trabalho tem como acrônimo EXEHDA-AR (EXEHDA – *AR Activity Recognition*), e contribui com o **Subsistema de Reconhecimento de Contexto e Adaptação**

do *middleware* EXEHDA (*Execution Environment for Highly Distributed Applications*) contemplando uma abordagem que capacite-o para o atendimento das demandas do **Reconhecimento de Atividades** em Casas Inteligentes, explorando para tanto metodologias de Ciência de Contexto.

1.1 Motivação e Objetivos

Considerando as várias naturezas das atividades desempenhadas pelo ser humano, bem como os diferentes níveis de complexidade das mesmas, o Reconhecimento de Atividades impõe muitos desafios, entre eles, o processamento de informações oriundas de diversos sensores localizados em ambientes dispersos. Apesar destes desafios, o fato das tecnologias de sensoriamento estarem em constante avanço e com custos cada vez menores, tem motivado diversas pesquisas por parte da comunidade da Computação Ubíqua (UbiComp) (DAWADI; COOK; SCHMITTER-EDGEcombe, 2015).

O foco do **Reconhecimento de Atividades** consiste, portanto, em identificar atividades de seres humanos em situações reais, interpretadas a partir dos dados coletados por sensores. Estes sensores podem estar presentes em objetos carregados pelo usuário ou, então, incorporados ao ambiente em que o mesmo interage (PERERA et al., 2015).

O reconhecimento e controle de atividades em ambientes instrumentados por sensores e atuadores, auxiliam na resolução de diversos problemas do dia a dia, como por exemplo, o acompanhamento de pessoas idosas (OSMANI; BALASUBRAMANIAM; BOTVICH, 2008), domótica (SINGLA; COOK; SCHMITTER-EDGEcombe, 2010), eficiência energética (LAI et al., 2012), entre outros. Deste modo, com a evolução dos ambientes inteligentes e com o emprego de métodos adequados, o Reconhecimento de Atividades vem gradualmente se consolidando em diferentes áreas, o que pode ser identificado pela revisão de literatura realizada.

A partir do Reconhecimento de Atividades nesses ambientes instrumentados, é possível fornecer informações relevantes para os envolvidos: pacientes e cuidadores. Um indivíduo acometido de uma doença degenerativa, como o Alzheimer, que precisa de cuidados constantes, encontrará em uma Casa Inteligente a possibilidade de ter uma vida mais confortável através do reconhecimento de suas atividades. Este reconhecimento irá dar ao paciente maior autonomia, alertando o mesmo e seus cuidadores sobre possíveis riscos à sua integridade física. O Reconhecimento de Atividades além da produção de alertas, através de registros, permitirá um acompanhamento histórico das atividades do paciente.

Esta dissertação contempla um aspecto que não havia sido trabalhado no grupo de pesquisa *Laboratory of Ubiquitous and Parallel Systems* (LUPS) até momento,

estando alinhada com a área de identificação computacional das atividades humanas (KIM; HELAL; COOK, 2010). A premissa perseguida foi de dotar o *middleware* EXEHDA de recursos arquiteturais que permitam seu emprego nesta área.

A exploração do potencial de *middleware* para reconhecimento não intrusivo de atividades em intervalos de tempo próximos ao momento em que as rotinas estão sendo observadas é uma situação pouco encontrada na literatura da área (AMIRIBESHELLI; BENMANSOUR; BOUCHACHIA, 2015).

Considerando isto, esta pesquisa tem como objetivo geral dotar o *middleware* EXEHDA de suporte para o Reconhecimento de Atividades. O esforço de pesquisa terá como acrônimo EXEHDA-AR (EXEHDA-*Activity Recognition*).

Considerando o objetivo geral, identifica-se como objetivos específicos:

- organizar fundamentos teóricos sobre Reconhecimento de Atividades, caracterizando as técnicas relacionadas a área;
- sistematizar trabalhos relacionados ao Reconhecimento de Atividades, construindo o estado da arte nesta frente pesquisa;
- conceber uma arquitetura denominada EXEHDA-AR, a ser integrada ao Sub-Sistema de Reconhecimento de Contexto e Adaptação do EXEHDA, que viabilize Reconhecimento de Atividades em janelas de tempos personalizáveis por cenário de uso;
- divulgar ante a comunidade científica os resultados atingidos pela pesquisa através de publicações em eventos e/ou periódicos especializados da área.

A concepção do EXEHDA-AR também compreenderá o estudo e a pesquisa de diversos conceitos relacionados a Ciência de Contexto na UbiComp, o qual irá considerar no seu desenvolvimento os seguintes aspectos:

- (i) garantir que o EXEHDA-AR tenha compatibilidade com os recursos de processamento semântico previstos no trabalho (DAVET et al., 2015);
- (ii) oferecer para os usuários um *framework* que facilite a associação de regras de inferência às diferentes informações contextuais oriundas de sensores;
- (iii) dar continuidade aos esforços de dotar o EXEHDA de um padrão de mercado direcionado a interoperabilidade. Este padrão seria utilizado tanto entre os serviços distribuídos, como para comunicação com outros *middleware*;
- (iv) atender as demandas operacionais do EXEHDA, enquanto *middleware* para UbiComp.

Assim sendo, esta dissertação promove o Reconhecimento de Atividades em Casas Inteligentes, por meio de sensores não intrusivos, e considerou a experiência do grupo de pesquisa LUPS na concepção abordagens para provimento de soluções voltadas à Ciência de Contexto.

1.2 Estrutura do Texto

O texto é composto por seis(6) capítulos. No capítulo 1 são apresentados o tema, motivação e objetivos que caracterizam a proposta desta dissertação de mestrado. Tópicos referentes as áreas de interesse como a Computação Ubíqua, Ciência de Contexto e Reconhecimento de Atividades são revisados no Capítulo 2, bem como as principais vertentes e modelos.

Com o intuito de caracterizar o estado da arte, o Capítulo 3 descreve e analisa alguns trabalhos relacionados. Uma visão geral e os principais aspectos quanto a concepção e modelagem do EXEHDA-AR, bem como uma discussão dos trabalhos relacionados em relação as estratégias concebidas são apresentados no Capítulo 4. Um estudo de caso empregando uma Casa Inteligente é discutido no Capítulo 5. Por fim o Capítulo 6 apresenta as principais considerações, publicações realizadas e trabalhos futuros.

2 FUNDAMENTOS CONCEITUAIS

Neste capítulo são sistematizados conceitos entendidos como importantes durante a revisão de literatura em relação ao tema proposto, e que constituem a fundamentação teórica desta dissertação de mestrado. Na primeira seção é tratado o uso a Ciência de Contexto na UbiComp e os seus principais conceitos. Por sua vez, o Reconhecimento de Atividades é contemplado na segunda seção, sendo revisadas as principais técnicas empregadas: tanto aquelas que exploram aprendizagem de máquina, como aquelas que utilizam técnicas baseadas em especificação, particularmente processamento semântico.

2.1 Ciência de Contexto na UbiComp

A **Computação Ubíqua** propõe uma interação entre o usuário e os recursos computacionais de forma natural, a qual requer o mínimo de intervenção humana e acontece de forma autônoma, interativa e relevante. Este cenário, possibilita que os usuários realizem suas atividades livremente, sem que haja preocupação em dar suporte à tecnologia, caracterizando o conceito de tecnologia calma (*Calm Technology*) (KRUMM, 2016).

Um dos preceitos da UbiComp é que os dispositivos devem se adaptar ao cotidiano dos indivíduos e em harmonia com outros objetos presentes. Considerando a escala de recursos desses ambientes, a premissa é automatizar a coleta de informações contextuais, reagindo a possível ocorrência de situações. Nesse sentido, a Ciência de Contexto constitui um dos principais desafios de pesquisa na UbiComp (KRUMM, 2016).

Esse ambiente visualizado na clássica publicação de (WEISER, 1991), assume que (i) é necessário existir um ambiente computacional invisível e transparente, com métodos intuitivos para que o usuário possa interagir com a computação sem precisar de conhecimento específico da área; (ii) o ambiente deve identificar quem está inserido neste, e assim propor recursos para atendê-lo de forma personalizada.

Essa visão de Weiser é considerada por autores como (COOK, 2012), (ROGGEN

et al., 2013), (YE; STEVENSON; DOBSON, 2015) entre outros, como possibilitador de novos horizontes de pesquisa para o Reconhecimento de Atividades, pela infraestrutura computacional oferecida pela UbiComp que combina conjuntos heterogêneos de dispositivos computacionais, como microcomputadores, *smartphones* e combinações de sensores e atuadores, que monitoram e disparam ações sobre um determinado ambiente.

Na perspectiva da UbiComp as atividades cotidianas vem gerando um crescente volume de dados contextuais. A utilização deste grande volume pode propiciar o desenvolvimento de novas aplicações que exploram a Ciência de Contexto para o Reconhecimento de Atividades em ambiente ubíquos (AMIRIBESHELI; BENMANSOUR; BOUCHACHIA, 2015).

Nesse sentido, a revisão das técnicas de reconhecimento pode ser dividida em duas principais vertentes: (i) Baseada em Aprendizagem de Máquina e (ii) Baseada na Modelagem Lógica e Raciocínio. Ambas as vertentes tem preocupações relacionadas ao tratamento do contexto.

2.1.1 Principais Conceitos

A **Computação Ciente de Contexto** refere-se a capacidade que um sistema computacional tem de perceber as características do meio ambiente que sejam de seu interesse, podendo utilizá-las para tomar decisões.

As aplicações Cientes de Contexto precisam monitorar e adaptar-se constantemente ao ambiente, compreendendo o contexto que estão inseridas. Essas aplicações trazem uma nova perspectiva para o desenvolvimento de sistemas mais ricos, que exploram a natureza dinâmica das modernas infraestruturas computacionais e da mobilidade dos usuários.

Uma definição clássica para contexto é a de (DEY and ABOWD, 2001), que definem contexto como qualquer informação que caracteriza a situação de uma entidade, sendo que uma entidade pode ser uma pessoa, um lugar ou um objeto considerado relevante para interação entre usuário e uma aplicação, incluindo o próprio usuário e a aplicação.

Para (ZIMMERMANN; LORENZ; OPPERMANN, 2007), qualquer informação que descreve contexto de uma entidade se enquadra em uma das cinco categorias de informações de contexto a seguir:

- **Individualidade:** está relacionada com as propriedades e atributos que descrevem a própria entidade. Esta informação compreende qualquer coisa que pode ser observada sobre uma entidade, normalmente, o seu estado.
- **Atividade:** abrange todas as tarefas nas quais a entidade está envolvida atualmente ou estará no futuro, e responde à pergunta: "O que a entidade quer

alcançar e como?”.

- **Localização:** descreve modelos de localização física ou virtual de uma entidade, bem como outras informações espaciais como velocidade e orientação.
- **Tempo:** engloba informações sobre o tempo como fuso horário, tempo atual ou qualquer tempo virtual. É considerado um aspecto vital para a classificação do contexto, pois a maioria das informações é relacionada com a dimensão temporal.
- **Relações:** esta categoria de informações de contexto capta as relações que uma entidade pode estabelecer com outras entidades. Potencialmente, uma entidade pode estabelecer qualquer número de relações diferentes com a mesma entidade. Além disso, as relações não são necessariamente estáticas, elas podem surgir e desaparecer dinamicamente.

Na visão de (VIEIRA; TEDESCO; SALGADO, 2009) sobre a definição de contexto, é feita uma distinção entre dois conceitos: (i) o elemento contextual que representa qualquer dado, informação ou conhecimento que possibilite caracterizar uma entidade em um domínio; e (ii) o contexto da interação entre um agente (humano ou de *software*) e uma aplicação, na realização de uma atividade, que corresponde ao conjunto de elementos contextuais instanciados que são necessários para apoiar a atividade atual. O elemento contextual é um tipo de informação estável que pode ser definido em tempo de projeto. O contexto é dinâmico e depende da atividade atual do agente, devendo ser definido em tempo de execução, em que ocorre a interação (VIEIRA; TEDESCO; SALGADO, 2009).

Para (XU et al., 2012) contexto é todo o conhecimento, mesmo que implícito, acerca de uma entidade em um domínio, que possibilite particularizar uma situação. Esta que poderá influenciar ou ativar um comportamento, seja de um agente ou de uma aplicação, durante a interação entre o agente e a aplicação na execução de uma atividade. Entende-se “situação” como uma condição especial, interessante para uma aplicação, que merece resposta desta aplicação em tempo de execução.

Inúmeros desafios são apresentados por (VENEZIAN et al., 2010), para as aplicações cientes de contexto. São eles:

- (i) a caracterização dos elementos de contexto;
- (ii) a aquisição do contexto a partir de fontes heterogêneas, tais como sensores físicos, base de dados, agentes e aplicações;
- (iii) a representação de um modelo semântico formal de contexto;
- (iv) o processamento e interpretação das informações de contexto adquiridas;

- (v) a disseminação do contexto a entidades interessadas de forma distribuída e no momento oportuno;
- (vi) o tratamento da qualidade da informação contextual.

2.1.1.1 Classificações de Informações Contextuais

O contexto em algumas situações pode ser relativamente estável e previsível, mas na maioria dos casos a sua complexidade aumenta devido a dificuldade que o projetista de aplicações cientes de contexto têm de enumerar o conjunto de todos os estados contextuais que podem existir, bem como definir quais ações devem ser executadas para diferentes estados.

A literatura propõe algumas classificações para informações contextuais. Essas informações são fornecidas por meio de sensores físicos, chamados de contextos de baixos nível, as quais são adquiridas sem qualquer interpretação adicional, tornando-as inexpressivas ou vulneráveis e pequenas mudanças podem gerar incertezas (YE et al., 2007).

O contexto de baixo nível possui limitações para modelar interações e comportamentos humanos, assim reduzindo as utilidades de aplicações sensíveis de contexto.

O contexto de alto nível se mostra uma alternativa para contornar esse problema. O mesmo utiliza a derivação de informação a partir dos valores não processados dos sensores, a qual é conhecida como raciocínio de contexto. Este raciocínio consiste na abstração do contexto de baixo nível ao criar uma nova camada que recebe as percepções dos sensores. Essas percepções geram o que se chama de situação que pode ser definida como um conjunto de características de contexto que são invariáveis em um determinado intervalo de tempo (WEISSENBERG; GARTMANN; VOISARD, 2006).

As situações são geradas pelas interpretações semânticas de baixo nível, quando a ocorrência de eventos causadores de mudanças, geram a necessidade da adaptação dos sistemas sensíveis ao contexto aonde essas alterações de estado caracterizam em situações. Uma mudança do valor de contexto que produz um evento pode disparar adaptação se a atualização do contexto gerar uma mudança de situação (YE; STEVENSON; DOBSON, 2015).

Pode-se representar as diferentes camadas de interpretação de contexto por meio de uma pirâmide como ilustra a Figura 1.

Com a interpretação do contexto torna-se possível modelá-lo de uma forma mais eficiente.



Figura 1: Visão geral das diferentes camadas de interpretação de contexto. Fonte: adaptada de (YE; STEVENSON; DOBSON, 2015).

2.1.2 Modelagem de Contexto

A definição do contexto possibilita a especificação do domínio do problema que está sendo tratado. Porém, definir contexto pode não ser suficiente para viabilizar o processamento computacional dos dados contextuais. Assim, a modelagem do contexto visa representar o mesmo em um formato que seja processável por máquina (PERERA et al., 2014).

A modelagem do contexto pode contemplar diversos aspectos que são considerados no processo de construção de *software*, tais como: facilidade de desenvolvimento, manutenção e evolução das aplicações, separação entre código e modelo, reuso e compartilhamento.

Bettini aborda três formas de modelar contexto: **Baseada em Fatos**, **Espacial** e **Baseada em Ontologia** (BETTINI et al., 2010).

2.1.2.1 Modelagem Baseada em Fatos

Abordagem de modelagem baseada em fatos foi inicialmente desenvolvida a partir da tentativa de criar modelos formais de contextos, a fim de suportar o processamento de consultas e raciocínio, assim como prover construções de modelagem adequadas para a sua utilização em etapas da Engenharia de Software, como, análise e projeto (BETTINI et al., 2010).

Tais abordagens de modelagem possuem seus fundamentos baseados em técnicas de modelagem de banco de dados.

Uma das abordagens mais populares consiste na *Context Modelling Language* (CML), que foi definida primeiramente em (HENRICKSEN; INDULSKA; RAKOTONIRAINY, 2002) e, então, refinada em trabalhos posteriores (HENRICKSEN; INDULSKA, 2004) e (HENRICKSEN; INDULSKA, 2006). Tal abordagem é baseada

em *Object-Role Modeling* (ORM) (HALPIN, 2001), que foi desenvolvida para modelagem conceitual de banco de dados.

A CML captura a heterogeneidade das fontes de informações de contexto, histórico de informações contextuais e, ainda, provê um mapeamento claro dos conceitos do mundo real para estruturas de modelagem.

Além disso, tal linguagem fornece um bom balanceamento entre capacidade de expressão e eficiência dos procedimentos de raciocínio. Entretanto, a modelagem baseada em fatos através da CML apresenta uma falta de suporte a descrições hierárquicas de contexto. A Figura 2 apresenta um exemplo de modelagem de contexto utilizando-se CML.

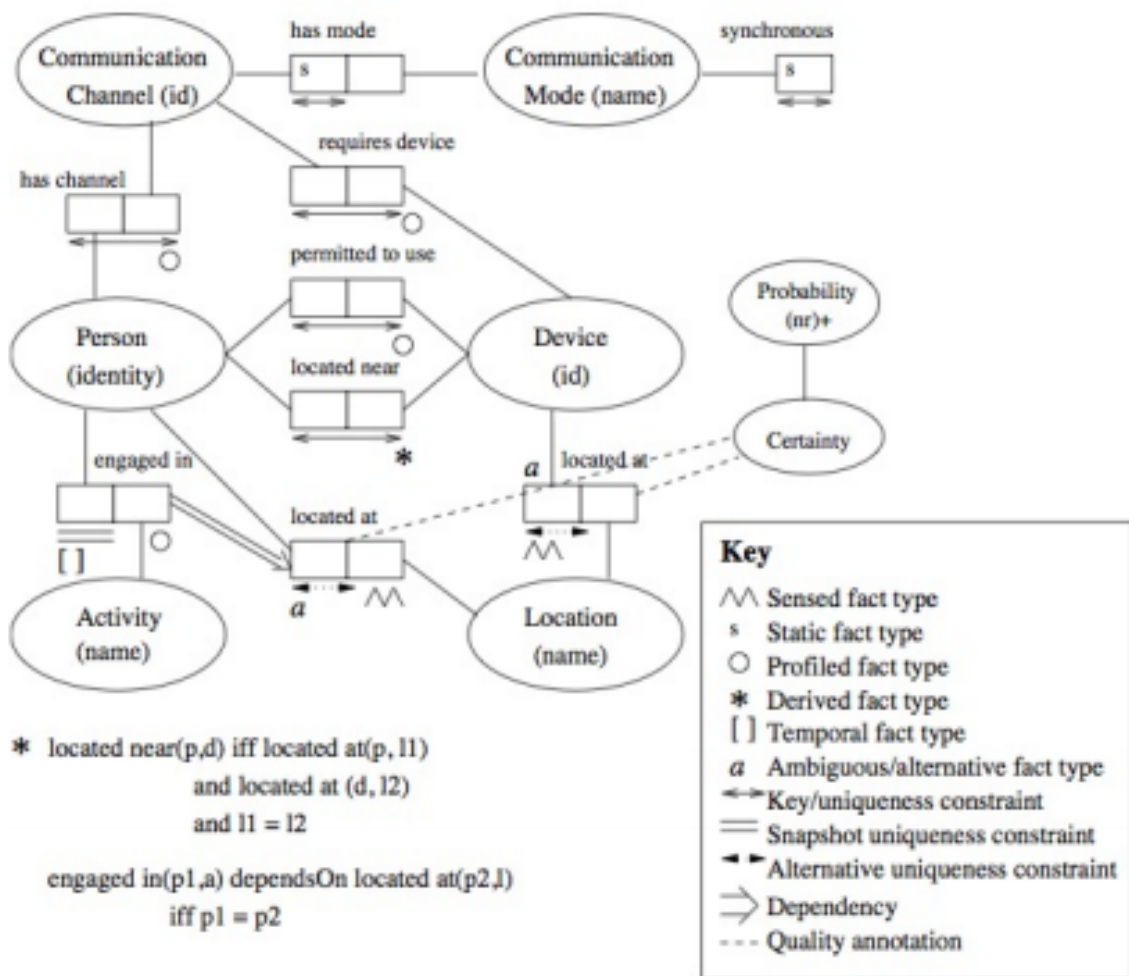


Figura 2: Exemplo da modelagem CML. Fonte: (BETTINI et al., 2010).

2.1.2.2 Modelagem Espacial

Informação espacial desempenha um importante papel em aplicações sensíveis de contexto. Consequentemente, grande parte das definições de contexto citam o espaço como uma propriedade vital. Por exemplo, conforme (DEY, 2001), espaço consiste em

um aspecto central das entidades: "Uma entidade é uma pessoa, lugar ou objeto que é considerado relevante na interação entre um usuário e uma aplicação, incluindo-se o usuário e a aplicação".

Desta forma, lugares são entidades espaciais e, ainda, a interação entre entidades, geralmente, envolve uma determinada proximidade. Portanto, algumas abordagens de modelagem de contexto priorizam fatores espaciais e temporais.

Modelos espaciais permitem raciocínio e interpretação sobre a localização e relacionamentos espaciais entre objetos. Tais relacionamentos compreendem a inclusão de objetos em determinada área e a distância entre objetos.

Conforme (BECKER; DÜRR, 2005), existem três classes de consultas típicas que podem ser realizadas sobre informações espaciais:

- (i) posição: retorna a posição de um objeto;
- (ii) alcance: retorna os objetos que estão localizados em uma determinada área de alcance;
- (iii) vizinho mais próximo: retorna uma lista de um ou mais objetos que estão próximos à posição de um objeto.

Grande parte dos modelos espaciais são modelos baseados em fatos (seção 2.1.2.1) que organizam sua informação contextual através da localização física. Tal localização pode ser pré-definida (se as entidades são estáticas), ou pode ser obtida a partir de sistemas de posicionamento que localizam objetos móveis e reportam suas posições para um sistema de gerenciamento de localizações. Basicamente, duas categorias de sistemas de coordenadas são suportadas por tais sistemas de posicionamento:

- Coordenadas geométricas: representam pontos ou áreas em um espaço métrico, como as coordenadas de Global Positioning System (GPS) (latitude, longitude e elevação acima do nível do mar). Ao utilizar funções geométricas, é possível a execução de consultas relacionadas a localização de objetos, como, distância entre objetos.
- Coordenadas simbólicas: são representadas por um identificador, como, por exemplo, o número de uma sala ou ID de um *access point*. Em contraste com as coordenadas geométricas, coordenadas simbólicas não oferecem relações espaciais. A fim de permitir interpretação espacial, como, distância entre objetos, deve-se prover informações explícitas sobre as relações espaciais entre pares de coordenadas simbólicas.

Modelos geométricos e geográficos de localização oferecem um mapeamento simples entre dados e posições, enquanto modelos simbólicos e relacionais de localização são mais fáceis para a construção e representação de uma percepção simples do espaço (através de relacionamentos, como, parte-de e localizado-próximo).

Tal escolha, determina como a informação contextual deve ser gerenciada (através de um banco de dados espacial, por exemplo), quais os métodos de raciocínio e interpretação são disponibilizados, assim como as classes de consultas que podem ser utilizadas (BETTINI et al., 2010). Porém, deve-se priorizar, cuidadosamente, as informações espaciais relevantes, a fim de priorizar a eficiência do processo de raciocínio e interpretação através da redução do tamanho da base de conhecimento (NICKLAS et al., 2008).

2.1.2.3 Modelagem Baseada em Ontologia

Visto que contexto pode ser considerado um tipo específico de conhecimento, é natural investigar se existe algum *framework* apropriado para representação de conhecimento e raciocínio para a manipulação de contexto (BETTINI et al., 2010).

O contraste entre a capacidade de expressão e complexidade de raciocínio tem motivado grande parte das pesquisas em modelos simbólicos de representação do conhecimento (BAADER, 2003).

Segundo (BETTINI et al., 2010) as ontologias consistem, essencialmente, em descrições de conceitos e seus relacionamentos, modelos contextuais baseados em ontologias exploram o poder de representação e raciocínio por diversos motivos:

- (i) a expressividade de linguagens, como *Web Ontology Language* (OWL) (HORROCKS; PATEL-SCHNEIDER; VAN HARMELEN, 2003), é utilizada para descrever dados contextuais complexos que não podem ser representados, por exemplo, por linguagens mais simples como CC/PP (KLYNE, 2001);
- (ii) possibilidade de compartilhar e/ou integrar informações contextuais entre diferentes fontes, devido ao formalismo semântico fornecido aos dados contextuais;
- (iii) as ferramentas de raciocínio disponíveis podem ser utilizadas, tanto para a verificação de consistência de um conjunto de relacionamentos que descrevem um cenário, quanto para reconhecer que um conjunto particular de instâncias de dados contextuais e seus relacionamentos revelam a presença de uma caracterização de contexto mais abstrata (por exemplo, a atividade do usuário pode ser automaticamente reconhecida).

Tipicamente, o formalismo escolhido para modelagem contextual baseada em ontologia é a OWL-DL (HORROCKS; PATEL-SCHNEIDER; VAN HARMELEN, 2003) ou

algumas de suas variações, visto que essa está se tornando um padrão entre diversos domínios de aplicações, além de ser suportada por diversos serviços de raciocínio (BETTINI et al., 2010).

Através da OWL-DL é possível a modelagem de um domínio específico ao definir classes, indivíduos, características de indivíduos (*datatype properties*) e relações entre indivíduos (*object properties*). Descrições complexas de classes e propriedades podem ser construídas através da composição de descrições elementares utilizando-se os operadores fornecidos pela linguagem.

Além do poder representativo de ontologias, uma vantagem adicional consiste no suporte a procedimentos de raciocínio. Desta forma, através do conhecimento já representado, é possível:

- Derivar novo conhecimento sobre o contexto atual automaticamente. O raciocínio ontológico pode ser executado através da inferência de nova informação contextual baseada nas classes e propriedades definidas, e nos objetos provenientes de sensores e outras fontes de contexto. Por exemplo, é possível derivar o conjunto de objetos que são relacionados a um determinado objeto através de uma propriedade particular (tal como, o conjunto de atividades que ocorrem em um lugar específico);
- Detectar possíveis inconsistências presentes na informação contextual. Assim, a verificação automática de consistência pode ser executada a fim de capturar possíveis inconsistências na definição de classes e propriedades da ontologia (por exemplo, uma classe sendo subclasse de duas classes disjuntas), ou na sua população (por exemplo, uma pessoa estar presente em diferentes lugares simultaneamente).

Além de prover um formalismo para representação de dados contextuais complexos, ontologias também desempenham um importante papel no compartilhamento de conhecimento, visto que estas provêm uma especificação formal da informação contextual.

Tal característica é importante para ambientes móveis e ubíquos, pois diferentes entidades heterogêneas e distribuídas interagem a fim de oferecer/trocar informação dos contextos dos usuários. Consequentemente, diversas ontologias descritas em OWL têm sido propostas para representar descrições de dados contextuais.

Entre os esforços mais destacados, pode-se citar as ontologias propostas para modelagem de contexto reconhecimento de atividades a PalSPOT (RIBONI et al., 2011), e DomoML (RATHNAYAKA; POTDAR; KURUPPU, 2011), proposta para modelagem em ambientes de Casa Inteligente.

2.1.2.4 *Comparações entre Modelagem Baseada em Fatos, Espacial e Baseada em Ontologia*

A fim de explicitar as diferenças entre os três modelos analisados para modelagem de contexto, foi criada uma tabela onde apresenta um comparativo entre os três modelos citados anteriormente com base nos requisitos que serão descritos de acordo com (BETTINI et al., 2010):

- **Mobilidade:** muitas aplicações sensíveis ao contexto são móveis (por exemplo, executadas em dispositivos móveis) ou dependem de fontes móveis para o fornecimento de informações contextuais (por exemplo, sensores móveis). Consequentemente, isso acrescenta problemas à heterogeneidade, visto que o fornecimento de informações contextuais deve ser adaptável à mudança ambiental. Além disso, a localização e o caráter espacial das informações capturadas desempenham um importante papel devido a esse requisito;
- **Relacionamentos e dependências:** existem diversos tipos de relações entre informações de contextos que devem ser capturados, a fim de garantir o comportamento correto das aplicações. Um desses relacionamentos é a dependência que pode existir entre informações contextuais de entidades/fatos. Por exemplo, uma mudança no valor de uma propriedade (como, largura de banda) pode afetar os valores de outras propriedades (como, nível restante de bateria);
- **Usabilidade de modelos formais:** modelos contextuais são criados por projetistas de aplicações sensíveis de contexto. Esses, também, são utilizados por sistemas de gerenciamento de contextos e aplicações sensíveis, a fim de manipular as informações contextuais. Consequentemente, as características importantes dos formalismos de modelagem são: a facilidade com que os projetistas podem traduzir os conceitos do mundo real para as estruturas do modelo e a facilidade com que as aplicações conseguem utilizar e manipular as informações contextuais em tempo de execução;
- **Histórico:** aplicações sensíveis de contexto podem necessitar do acesso a estados passados e estados futuros (prognóstico). Portanto, históricos de contextos e suas características que devem ser capturadas pelos modelos de contexto e gerenciada pelo sistema de gerenciamento de contexto. Alguns trabalhos por exemplo, (PEDDEMORS; EERTINK; NIEMEGEREERS, 2010), (UDEN, 2006), e (CASSENS; KOFOD-PETERSEN, 2006) buscam definir o papel do tempo em sistemas sensíveis de contexto. Em tais esforços, nota-se que o histórico é uma parte importante do contexto como um todo e não apenas de um subconjunto deste, pois o comportamento de uma entidade reflete sua cultura e desenvolvimento histórico em circunstâncias específicas (MCMICHAEL, 1999). Sendo

assim, a informação temporal é crucial para a interpretação do comportamento das entidades que integram o espaço ubíquo. Por outro lado, o gerenciamento de históricos de contextos é penoso, caso o número de atualizações seja muito alto. Assim, pode ser inviável o armazenamento de cada valor para acesso futuro, e, então, técnicas de sumarização, como, a utilização de sinopses históricas dos dados, devem ser aplicadas.

- **Heterogeneidade:** modelos projetados para as informações coletadas devem lidar com uma grande variedade de fontes provedoras de informações que diferem em taxa de atualização de dados e nível semântico. Ao mesmo tempo que alguns sensores podem observar determinadas situações do ambiente físico e prover acesso rápido e em tempo real, outros sensores oferecem dados não processados (como, coordenadas de um GPS ou imagens de uma câmera), que necessitam ser interpretados antes de serem utilizados pelas aplicações. Por outro lado, informações fornecidas pelo usuário como perfis de usuários, são atualizadas raramente e, geralmente, não requisitam interpretações adicionais. Além disso, informações contextuais podem ser derivadas de informações contextuais já existentes. Por exemplo, dados obtidos de banco de dados ou bibliotecas digitais, como dados de mapas geográficos, são frequentemente estáticos. O modelo de contexto deve ser capaz de expressar esses diferentes tipos de informação e o sistema de gerenciamento de contexto deve prover gerenciamento às informações dependendo do seu tipo;
- **Raciocínio:** aplicações sensíveis de contexto utilizam as informações para avaliar se existe uma mudança na situação do contexto do usuário e/ou ambiente computacional. A decisão de adaptação a alguma mudança, geralmente, requer capacidades de raciocínio. Portanto, é importante que técnicas de modelagem de contexto sejam capazes de suportar tanto verificação de consistência do modelo, quanto técnicas de raciocínio sobre o contexto. Estas podem ser utilizadas para derivar novos fatos/situações a partir de fatos/situações já existentes, além de avaliar abstrações de contextos de alto nível que modelam as situações do mundo real. Além disso, as técnicas de raciocínio devem ser computacionalmente eficientes.
- **Acesso eficiente:** ao mesmo tempo que o acesso eficiente às informações contextuais é necessário, este pode ser um requisito difícil de implementar na presença de modelos grandes e numerosos dados. A fim de selecionar os objetos relevantes, atributos que indicam o caminho para acesso adequado devem ser representados na modelagem contextual. Esses caminhos de acesso representam as dimensões em que as aplicações frequentemente selecionam

as informações dos contextos, tipicamente suportadas por índices. Essas dimensões são, geralmente, referenciadas aos contextos primários, em contraste aos contextos secundários, que são acessados utilizando-se o contexto primário. Comumente, os atributos de contextos primários utilizados são a identificação de objetos de contextos, localização, tipo de objeto, tempo ou atividade do usuário.

A Tabela 1 apresenta um comparativo entre os 3 tipos de modelagens citados levando em consideração os requisitos descritos acima a legenda de forma simples aonde (Complemente) refere-se que aquela modelagem atende completamente aquele requisitos, (Parcial) que atende o requisito em parte e (Não Atende) não consegue atender a esse requisito.

Tabela 1: Comparações entre os três tipos de modelagem. Fonte: (BETTINI et al., 2010).

Requisitos	Modelagem Baseada em Fatos	Modelagem Espacial	Modelagem Baseada em Ontologia
Mobilidade	Parcial	Completamente	Parcial
Relacionamentos	Parcial	Parcial	Completamente
Usabilidade	Completamente	Parcial	Parcial
Histórico	Completamente	Completamente	Completamente
Heterogeneidade	Completamente	Parcial	Completamente
Raciocínio	Parcial	Não Atende	Completamente
Eficiência	Parcial	Completamente	Completamente

A modelagem baseada em ontologia foi escolhida para este trabalho, por permitir uso de raciocinadores para inferir sobre as informações contextuais coletadas, pelo emprego de regras e consultas semânticas. Considerando a premissa de envolver o mínimo possível o usuário, uma abordagem baseada em especificação cujas as regras para inferência são construídas a partir da expertise do desenvolvedor se mostram oportunas justificando assim a adoção de processamento semântico nessa dissertação.

2.2 Reconhecimento de Atividades

O ser humano realiza dezenas de atividades por dia, como tomar banho, assistir televisão, cozinhar, dormir, etc. Devido ao avanço das tecnologias de sensoriamento, o reconhecimento das atividades humanas baseado em dados provenientes de sensores tem, recentemente, motivado diversas pesquisas por parte da comunidade de Computação Ubíqua (GU et al., 2009). Tipicamente, esses sensores podem estar presentes em objetos carregados pelo usuário ou, então, incorporados ao ambiente em que ele integra (ATALLAH; YANG, 2009).

O conhecimento e o controle das atividades ajudam a resolver problemas de vários domínios do dia a dia, por exemplo, monitoramento de pessoas idosas (OSMANI; BALASUBRAMANIAM; BOTVICH, 2008), domótica (SINGLA; COOK; SCHMITTER-EDGEcombe, 2010), economia de energia (LAI et al., 2012) entre outras. Portanto, com a evolução dos Ambientes Inteligentes e a possibilidade de aplicação de técnicas de inteligência artificial, permitem que tais atividades sejam reconhecidas automaticamente.

O objetivo do Reconhecimento de Atividades reside, portanto, em reconhecer atividades humanas comuns em situações reais, interpretadas a partir dos dados coletados por sensores.

O Reconhecimento de Atividades humanas é abordado de diferentes formas. (MUNGUIA TAPIA, 2003) afirma que alguns trabalhos fazem análises complexas de sinais de sensores, por exemplo, sensor de presença, vídeos de câmeras e áudios de microfones e outros trabalhos utilizam *Radio-Frequency IDentification* (RFID) para reconhecer atividades. O Reconhecimento de Atividades por meio de sensores ainda é um desafio, pois a extração de características dos sinais e o reconhecimento de padrões por meio de algoritmos de aprendizagem de máquina, ainda é uma tarefa complexa devido à diversidade de cenários.

Especificamente, o processo de Reconhecimento de Atividades impõe alguns desafios relativos à interpretação devido à natureza das atividades humanas, que podem se apresentar de forma sequencial, intervalada e concorrente (GU et al., 2009).

Existem diversos modelos empregados para o Reconhecimento de Atividades, de acordo com (KIM; HELAL; COOK, 2010) podem ser divididos em duas vertentes: modelos baseados em Aprendizagem de Máquina e modelos baseados em Modelagem Lógica.

2.3 Métodos Baseados em Aprendizagem

A ideia por trás da utilização do conceito de aprendizagem não é apenas para saber como atuar, mas também tem como intuito aumentar a habilidade do agente para atuar no futuro. A aprendizagem ocorre através de como o agente observa as suas interações com o mundo e os seus próprios processos de decisão.

A aprendizagem indutiva é uma classe de algoritmos para aprendizagem supervisionada determinística baseada em conjuntos de hipóteses aprendidas com recurso a exemplos de treino. Este tipo de algoritmos fornece uma hipótese que é uma boa aproximação da predição correta e que responde bem quando são problemas generalizados, mas isto levanta o problema da indução.

Dentre os algoritmos de aprendizagem supervisionada os mais utilizados no estado da arte para o Reconhecimento de Atividades: *Hidden Markov Model* (HMM), *Network*

Bayesian (NB) e *Conditional Random Field* (CRF) (AMIRIBESHELI; BENMANSOUR; BOUCHACHIA, 2015).

2.4 Ontologia para Reconhecimento de Atividades

A segunda vertente de algoritmos de Reconhecimento de Atividades é baseada em modelagem lógica e raciocínio. A lógica destas abordagens é explorar a representação do conhecimento para a modelagem de dados de atividade e de sensores e usar o raciocínio lógico para realizar o reconhecimento da atividade.

O procedimento geral de uma abordagem lógica inclui (i) usar um formalismo lógico para definir e descrever explicitamente uma biblioteca de modelos de atividade para todas as atividades possíveis em um domínio, (ii) agregar e transformar dados de sensores em termos e fórmulas lógicas, e (iii) executar o raciocínio lógico, por exemplo, dedução, abdução e subjunção, para extrair um conjunto mínimo de modelos de cobertura de interpretação a partir da biblioteca de modelo de atividade com base em um conjunto de ações observadas, o que poderia explicar as observações.

Existe uma série de métodos de modelagem lógica e algoritmos de raciocínio em termos de teorias lógicas e formalismos de representação. Por exemplo, Kauz (Kautz, 1991) adotou axiomas de primeira ordem para construir uma biblioteca de planos hierárquicos para o reconhecimento do plano. Wobke (Wobke, 2002) estendeu o trabalho de Kauz usando a teoria da situação para abordar as diferentes probabilidades de planos inferidos. Bouchard (Bouchard, 2006) usou a ação de Descrição Lógica (DL) e teoria de retículo para reconhecimento de planos com ênfase particular na modelagem e raciocínio de planos intra-dependências.

A teoria de evento e formalismo lógico, é explorada por (CHEN; NUGENT, 2009), para especificação explícita, manipulação e raciocínio de eventos, para formalizar um domínio de atividade no reconhecimento de atividade e assistência utilizando ontologias.

A ontologia de reconhecimento de atividade (RIBONI; BETTINI, 2011) do projeto PalSPOT modela atividades individuais e sociais. Os tipos de interação são modelados como reconhecimento, pedindo opinião, comentário, opinião negativa / positiva, proposta ou pedido de informação.

A granularidade de atividade é ligeiramente mostrada (basicamente, apenas um nível de atividade); entretanto, uma extensa taxonomia está disponível para atividades pessoais, físicas e profissionais, atividades de viagem, posturas e atividades usando diferentes tipos de dispositivos ou artefatos. Uma representação baseada em intervalos de atividades modela a sobreposição destes no tempo. Outras entidades são corredor interno, elevador calçada e rotas de comunicação. Locais simbólicos (interno, externo) e granularidade de tempo são fornecidos.

A ontologia PalSPOT é usada no *middleware* de agregação de contexto CARE (AGOSTINI; BETTINI; RIBONI, 2009), que mapeia dados de contexto para classes ontológicas e propriedades, e interage com o sistema COSAR (RIBONI; BETTINI, 2011), que recupera informações sobre atividades usando apenas racionalistas ontológicos / estatísticos híbridos.

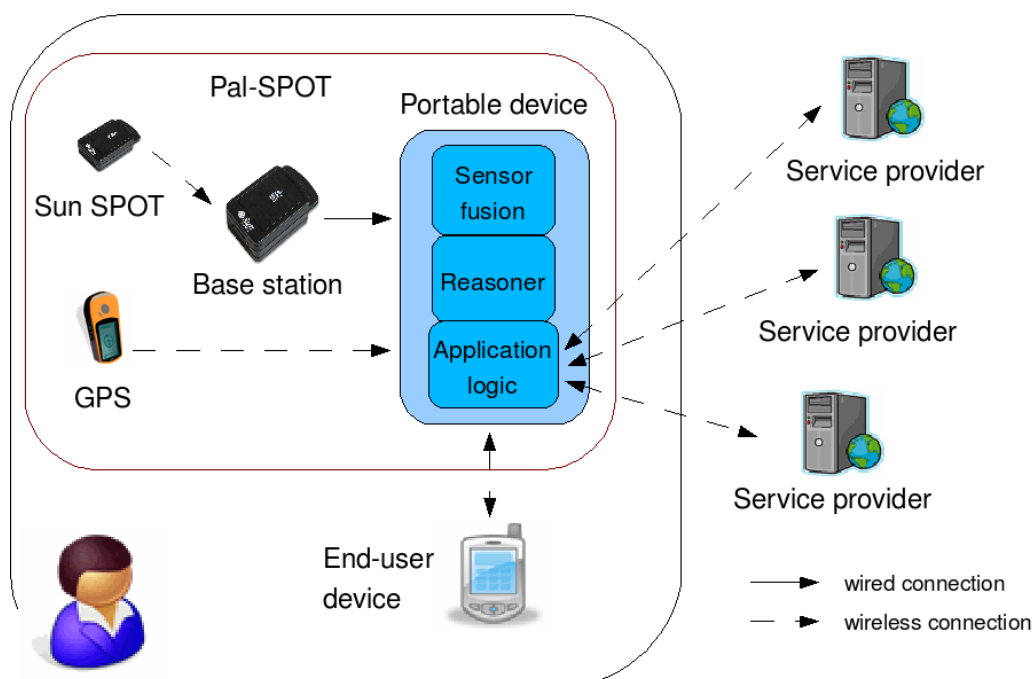


Figura 3: Visão geral PalSPOT. Fonte: (RIBONI; BETTINI, 2011).

A *CONtext ONtology* (CONON) (WANG et al., 2004) define conceitos gerais em uma ontologia superior, como localização (interna, externa, com diferentes características e variáveis ambientais, condições climáticas), atividade, pessoa ou entidade computacional, TV, DVD *player*). CONON permite extensões de forma hierárquica adicionando conceitos específicos de domínio, onde diferentes ambientes inteligentes são modelados (casa, escritório, veículo, etc).

Atividades (com início e hora de fim) são divididos em atividades deduzidas (jantar, filme) e programadas (festa, aniversário). O estado de entidades de espaços interiores, por exemplo, cortina, porta, janela, também é representado. Algumas ontologias específicas de domínio são o domínio doméstico (por exemplo, dormir, tomar banho, cozinhar, assistir TV, jantar) e ontologias de domínio de escritório. As categorias de raciocínio empregadas são raciocínio ontológico DL e raciocínio definido pelo usuário usando lógica de primeira ordem (através de regras customizadas).

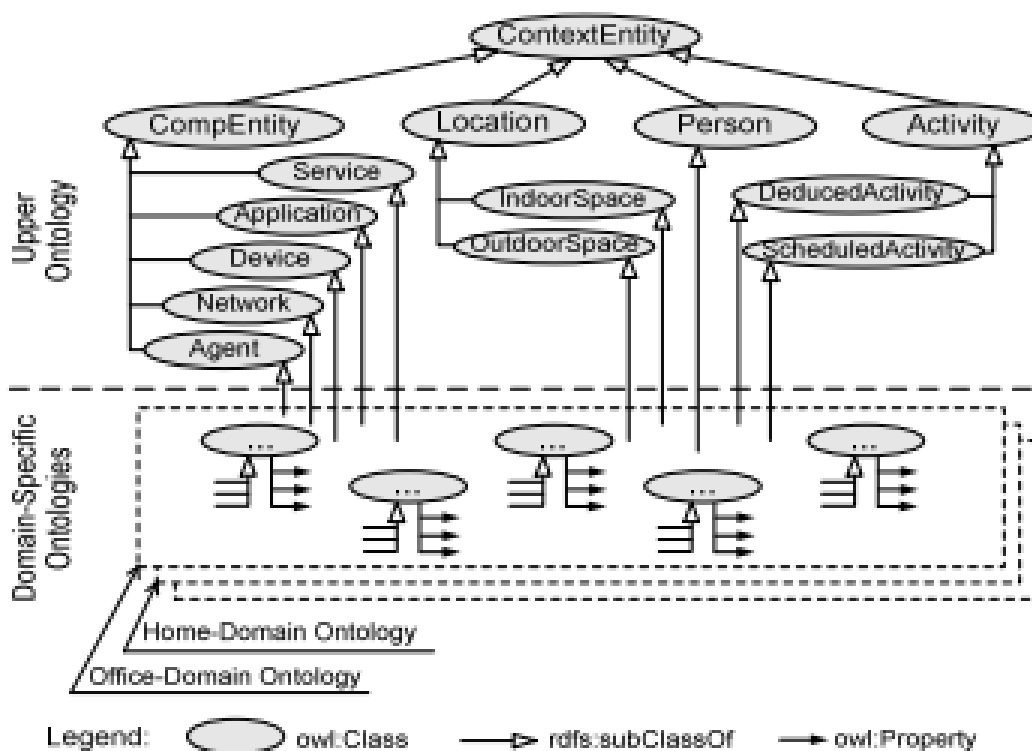


Figura 4: Visão geral CONON. Fonte: (WANG et al., 2004).

A *Pervasive Information Visualization Ontology* (PiVOn) é um modelo de contexto formal composto por quatro ontologias independentes (usuários, ambiente, dispositivos e serviços) que descrevem ambientes inteligentes. Algumas propriedades dos elementos principais na ontologia do usuário são localização, identidade, atividade e tempo (HERVÁS; BRAVO; FONTECHA, 2010).

O contexto é analisado a partir da perspectiva da teoria dos cinco (5) Ws para projetar sistemas sensíveis de contexto. O resultado pode ser resumido em uma taxonomia bidimensional de elementos de contexto: a primeira definida pelas quatro categorias principais do contexto (usuário, ambiente, dispositivo e serviço) e a segunda, pelos cinco (5) Ws.

Os eventos na ontologia têm lembretes, e fazem parte de uma agenda de usuários. Eles envolvem contatos (FOAF) do usuário, que podem estar em situações de usuário, e possivelmente acompanhados por algum outro usuário(s). Os usuários realizam tarefas (que podem ter subtarefas) que têm um objetivo e usam alguns serviços.

As tarefas têm tipos, papéis, níveis de significância, progresso, questões de tempo e espaço e níveis de atenção. Situações de usuário também desempenham um papel e pertencem à disponibilidade do usuário, um estado de espírito ou uma tarefa.

A ontologia do dispositivo determina os **tipos de dispositivos (autônomo, dependente, sensor, atuador)**, os serviços fornecidos por um dispositivo, o proprietário

do dispositivo, *status*, localização, seu *hardware* e perfis de *software*, *hardware* de comunicação, seu uso e compatibilidades, etc. A ontologia de ambiente representa a co-localização de objetos próximos (*inFrontOf*, *on*, *under*, *behind*), *includedIn*, associado a outros.

Os espaços são modelados com a área em que são localizados (prédio, piso), sua finalidade, estrutura e capacidade. A ontologia de serviço inclui uma visualização associada para cada serviço (exibido por dispositivos) que têm conteúdo. O conteúdo têm uma forma visual e é transformado em dados. A forma visual tem diferentes tipos de parâmetros de escalabilidade (filtragem, paginação, complexidade, latência).

A *Ontology Situation* (OntSit) é dividida em camadas de Situação e Contexto. A situação é definida como um conjunto de contextos na aplicação durante um período de tempo que afeta o comportamento futuro do sistema. Um contexto é qualquer propriedade instantânea, detectável e relevante do ambiente, sistema ou usuários, como localização ou largura de banda disponível (YAU; LIU, 2006).

A ontologia considera as situações atômicas e compostas. Compostas são situações temporais, de conjunção, de disjunção e de negação. As situações compostas podem ser integradas por situações atômicas através de operadores de contexto booleano que atuam sobre domínios de valores de contexto.

Em relação ao contexto, a ontologia de Situação é classificada em contexto de dispositivo, usuário e ambiente. Uma entidade pode satisfazer uma situação tendo dados de contexto relacionados a um determinado valor, por exemplo, valor de flutuação de temperatura, dentro de um domínio de contexto (de memória disponível).

Os domínios de valor de contexto são fornecidos com operações de dados. Um exemplo de um cenário de conferência inteligente pode especificar a situação *ReadyForMeeting* como a conjunção de duas situações atômicas: *InConferenceRoom* e *LightOn*, onde o valor do contexto de localização da situação *InConferenceRoom* é o mesmo que *crLocation* e a situação do *LightOn* é representada como "o valor *lightContext* é verdade".

2.5 Considerações sobre o Capítulo

Este capítulo apresentou os esforços realizados na revisão de literatura resumindo aspectos importantes que contribuíram para o desenvolvimento desta dissertação de mestrado.

Foram abordados conceitos de Ciência de Contexto na UbiComp, a classificação e modelagem de contexto, como também a caracterização do Reconhecimento de Atividades, suas principais vertentes de pesquisa e as técnicas mais utilizadas para o Reconhecimento de Atividades.

Considerando que esta dissertação irá explorar técnica baseada em especificação

para o Reconhecimento de Atividades, o próximo capítulo irá discorrer sobre trabalhos que exploram este tipo de técnica.

3 TRABALHOS RELACIONADOS

Tendo como base as premissas que motivaram o desenvolvimento da pesquisa nesta dissertação, foram identificados quando da revisão de literatura diversos trabalhos relacionados na área de Reconhecimento de Atividades. Os trabalhos foram selecionados tendo em vista os seguintes aspectos: (i) ambiente *indoor* de uma Casa Inteligente, (ii) o tratamento das informações contextuais e (iii) Reconhecimento de Atividades utilizando processamento semântico. Este capítulo além de realizar a descrição dos trabalhos relacionados, ao final, apresenta uma comparação entre os mesmos.

3.1 A Context-Aware System Infrastructure for Monitoring

Nesse trabalho, os autores propuseram uma arquitetura com três camadas de contexto para monitorar e reconhecer atividades da vida diária em uma Casa Inteligente. Essa arquitetura prevê a inclusão de funcionalidades que vão desde a coleta de dados de baixo nível até a extração de conhecimento de contexto de alto nível (NI; GARCÍA HERNANDO; CRUZ, 2016).

A seguir estão descritas as três camadas previstas na arquitetura proposta:

- **Camada Sensível ao Contexto:** é a camada onde o contexto é coletado através de sensores heterogêneos e dispositivos embutidos no ambiente doméstico inteligente ou usados pelo usuário, como hardware físico (por exemplo, sensores de luz, temperatura, acelerômetros e interruptores de portas) e componentes de rede. Esta camada é responsável pela obtenção de informações de contexto não processadas de várias fontes.
- **Camada de Gerenciamento de Contexto:** nesta camada os dados capturados provenientes da Camada Sensível ao Contexto são pré-processados, interpretados pelo modelo ontológico de contexto e inferidos pelo raciocinador de contexto.
- **Camada de Aplicação:** o objetivo central desta camada é servir como interface para que os usuários acessem as informações de contexto fornecidas pela camada de gerenciamento.

De maneira mais específica, a camada de gerenciamento de contexto funciona com informações oriundas da camada sensível ao contexto, para assim fornecer abstração dos dados para manipulação de contexto de nível mais alto.

Esta camada é organizada em três componentes: (1) o componente de pré-processamento de dados, o qual processa informações coletadas a partir da camada de detecção para construir uma base de conhecimento; (2) o modelo de ontologia de contexto abstrai dados relacionados a *Activity Daily Live* (ADL) de fontes heterogêneas e os converte em representações formalizadas para facilitar a interoperabilidade e reutilização e; (3) o raciocinador de contexto fornece serviços de raciocínio lógico (por exemplo, derivando contextos de alto a baixo nível) para processamento de contextos relacionados a ADL através do mecanismo de raciocínio.

A Figura 5 representa como as três camadas se relacionam:

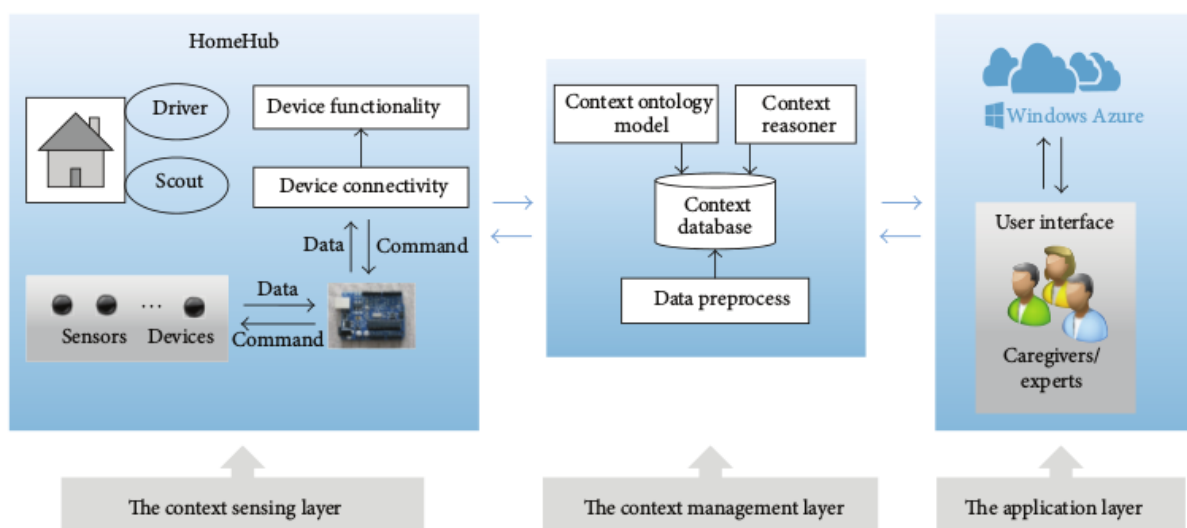


Figura 5: Representação das três camadas. Fonte: (NI; GARCÍA HERNANDO; CRUZ, 2016).

Os resultados obtidos neste trabalho, utilizando a plataforma de pesquisa *Lab of Things* (LoT) (DIXON et al., 2012), desenvolvida pela Microsoft, resultou no gerenciamento de recursos de *hardware* de uma Casa Inteligente e nos testes de coletas dos dados de sensores de presença, para posteriormente aplicar no modelo ontológico o Reconhecimento de Atividades diárias.

A ontologia desenvolvida para modelar o contexto em que as atividades ocorrem foi implementada utilizando o modelo *Simple Model Event* (SME) com o objetivo de representar as atividades básicas e usou como base a ontologia DOLCE + Dns Ultralite (DUL), para representar o domínio (Casa Inteligente).

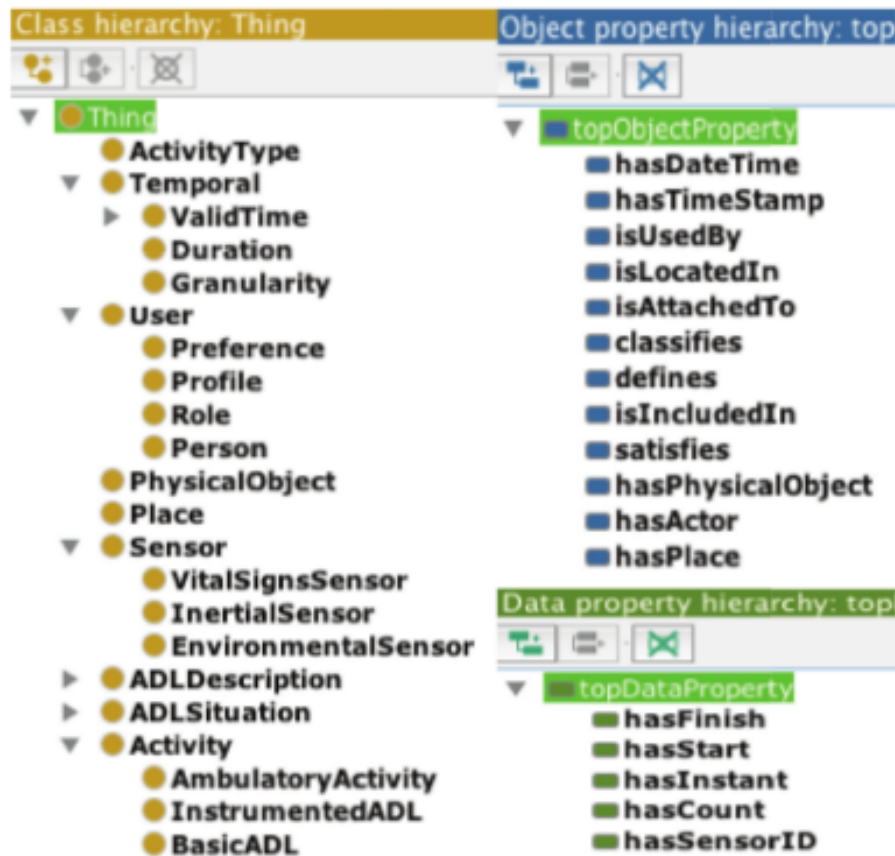


Figura 6: Visão das classes, relacionamentos e propriedade de dados da ontologia. Fonte: (NI; GARCÍA HERNANDO; CRUZ, 2016).

3.2 AAL Domain Ontology for Event-based Human Activity Recognition

Este trabalho propõe um *framework* baseado em ontologias que visa efetuar consultas semânticas num repositório de dados contextuais, com o objetivo de prover Reconhecimento de Atividades, cujos registros são oriundos de uma rede de sensores de fontes heterogêneas (CULMONE et al., 2014).

A arquitetura da estrutura proposta é ilustrada na Figura 7. Em primeiro lugar, os dados disponíveis em um banco de dados MySQL são mapeados na OntoAALISABETH, uma ontologia de domínio para o projeto *Ambient-Aware LifeStyle tutor, Aiming BETter Health* (AALISABETH). Na continuidade é buscada uma correspondência entre os elementos do banco de dados e os da ontologia. Esses dados são reorganizados de acordo com seu significado, onde a ontologia desempenha um papel de pré-processamento.

A ontologia empregada chama-se OnoAALISABETH, e o principal elemento representado define a semântica do domínio considerado e é utilizada como uma base de conhecimento compartilhada para todos os componentes relacionados. Esta ontolo-

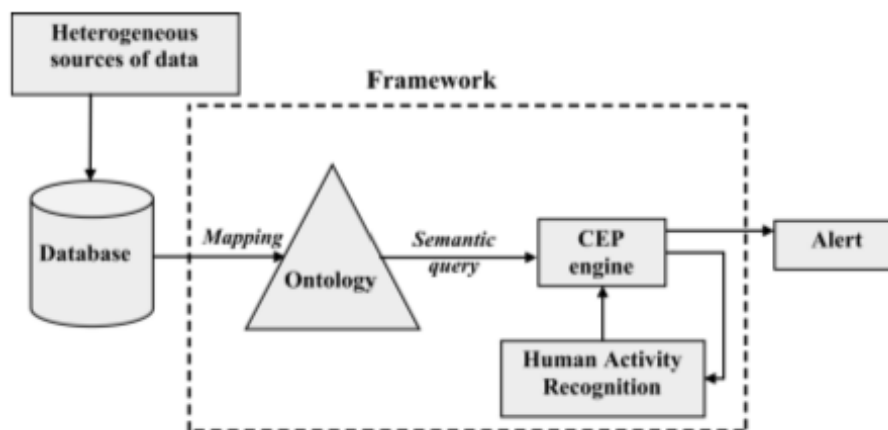


Figura 7: Visão geral do framework. Fonte: (CULMONE et al., 2014).

gia mostra diferentes camadas de abstração que, em conjunto, formam uma estrutura tipo pirâmide (vide figura 8).



Figura 8: Pirâmide das ontologias utilizadas. Fonte: (CULMONE et al., 2014).

A seguir estão descritas as quatro ontologias empregadas na arquitetura:

- **AAL Ontology:** É uma ontologia de domínio superior, onde conceitos gerais de contexto AAL são formalizados o mais genérico possível, transportando informações estáticas.
- **AAL-Building Specific Ontology:** Esta ontologia específica de construção *Ambient Assisted Living* (AAL), formaliza os diversos componentes do ambiente doméstico como a estrutura real do ambiente e disposição das salas, as informações pessoais sobre quem mora na casa, quais sensores são instalados na rede e como eles se comunicam.
- **Data Ontology:** A característica dinâmica da ontologia começa nesta camada. Aqui, as classes anteriores são preenchidas com os indivíduos correspondentes aos registros coletados no *Data Base* (DB). Esse procedimento é permitido por um mapeamento de banco de dados para ontologia. Em geral, esse mapeamento gera a ontologia automaticamente a partir de um DB inteiro através de um sistema de acesso a dados baseado em ontologias (OBDA), onde a estrutura do DB é importada para criar os componentes de ontologia relacionados.
- **View Ontology:** Nesta etapa um conjunto de regras internas, são introduzidas para adquirir novos conhecimentos estabelecendo novas conexões de propriedade de objeto entre entidades não relacionadas. Esta etapa de revisão é a base

do processo de pré-processamento, onde o raciocínio desempenha o papel principal que consiste em criar uma visão alternativa do conhecimento.

A implementação do *framework* pode ser dividida em duas partes: a primeira, foi com relação ao mapeamento e descrição dos dados para as ontologias, testada no Protégé, utilizando o Pellet *Reasoner*; a segunda, utilizou a ferramenta Esper para processamento de *Complex Event Processing* (CEP), por diversas razões, entre elas, o fato do Esper possuir uma biblioteca Java de código aberto para CEP, ser utilizada em diferentes aplicações de fluxo de dados e CEP e, por último, a presença de adaptadores que permitem ao usuário fornecer diferentes formatos de entrada para a representação de eventos.

O caso de estudo simulado para testar essa abordagem, busca reconhecer a atividade de preparação do café da manhã, onde são identificadas como ações atômicas, levantar-se das 5:00 às 11:00; ir à cozinha dentro de uma hora, depois se levantar; pegar utensílios de cozinha e abrir o refrigerador.

Neste estudo de caso são descritos os sensores envolvidos com a atividade, como o de presença na cama e na cozinha e o de abertura de porta. Esses sensores com seus *timestamps* relacionados são armazenados em um banco de dados MySQL, sendo mapeados para a ontologia OntoAALISABETH, antes do processamento semântico.

3.3 Activity Recognition using Context-Aware Infrastructure Ontology in Smart Home Domain

Wongpatikaseree e colaboradores (2012), propõem no seu trabalho de pesquisa uma infra-estrutura baseada em ontologias para reconhecimentos de atividades de uma Casa Inteligente, denominando-o *Ontology Based Activity Recognition* (OBAR) (WONGPATIKASEREE et al., 2012).

Entre as contribuições deste trabalho, destacam-se: (1) o projeto de uma ontologia de infraestrutura de contexto para modelar o contexto do usuário na Casa Inteligente; (2) propõe conceitos para distinguir as atividades por meio de conceitos baseados em objetos, localização e postura do corpo humano; e (3) um sistema de Reconhecimento de Atividades baseado em ontologias (OBAR) para reconhecer a atividade e pesquisar a informação semântica no sistema, chamada sistema de busca de ontologia semântica (SOS).

A arquitetura do sistema OBAR é ilustrada na Figura 9.

Com a coleta de todos os dados de observação dos sensores, o contexto do usuário é armazenado em um banco de dados relacional. Com base nesse contexto, a ontologia de infraestrutura sensível ao contexto é projetada pela aplicação Hozo,

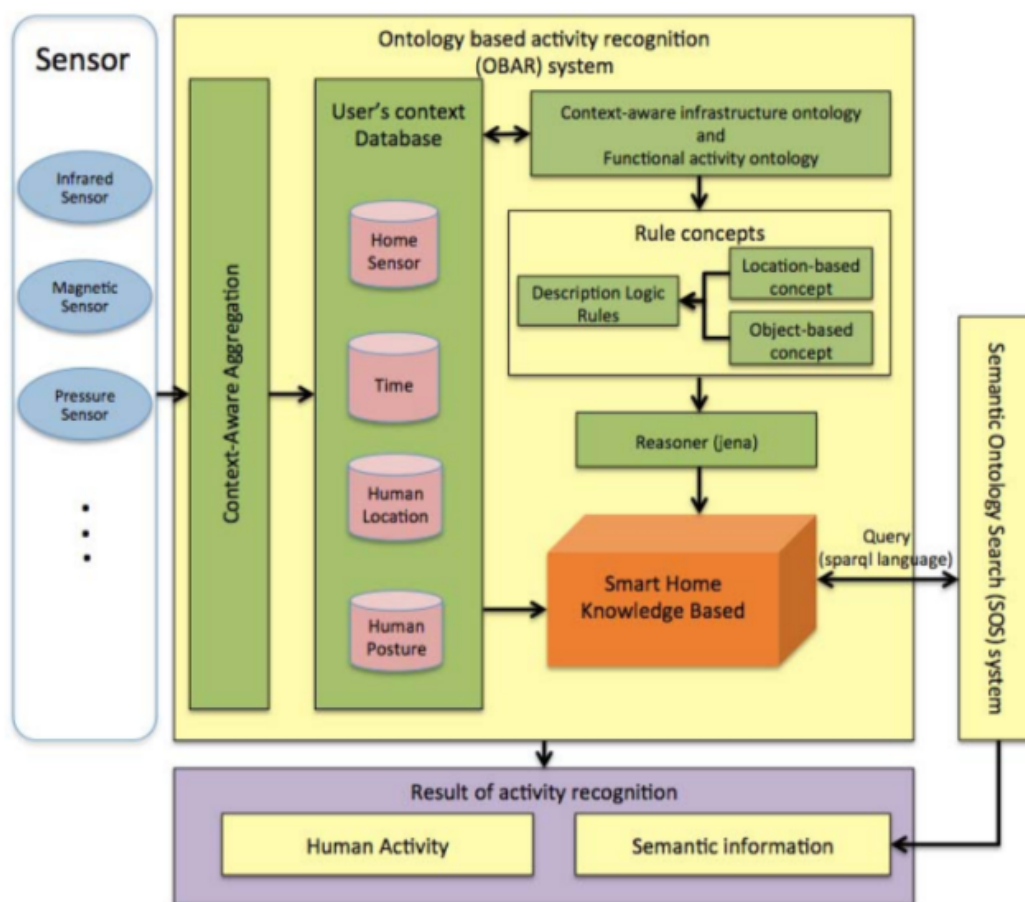


Figura 9: Visão geral da arquitetura do sistema (OBAR). Fonte: (WONGPATIKASEREE et al., 2012).

que é um ambiente para construir/usar ontologias baseadas em uma consideração fundamental de "Função" e "Relacionamento".

O sistema OBAR, emprega as regras de lógica de descrição (DL) para explicar com base no conceito de objetos e localização. O raciocinador da API Jena é utilizado para inferir nas regras DL antes de armazenar uma nova instância de atividade.

Foram concebidas duas ontologias: A primeira é responsável pela infraestrutura do contexto que consiste de seis (6) classes, como as seguintes: *Location*, *Object*, *Human Posture*, *Context*, *InferredActivity* e *Functional Activity*.

Na Figura 10, tem-se uma visão geral da ontologia de infraestrutura de contexto que integra a arquitetura de *software* do OBAR.

A segunda ontologia da atividade funcional é projetada para a definição das atividades. Sua super classe, *FunctionalActivity*, é mapeada para a classe *Context* através da classe *InferredActivity*. Foram utilizados os dados da classe *Context* para reconhecer as atividades com base nos conceitos baseados em objetos e na localização.

As regras DL são projetadas com base nos conceitos de objeto e com a combinação da localização. O esquema de regras concentra-se no contexto dos

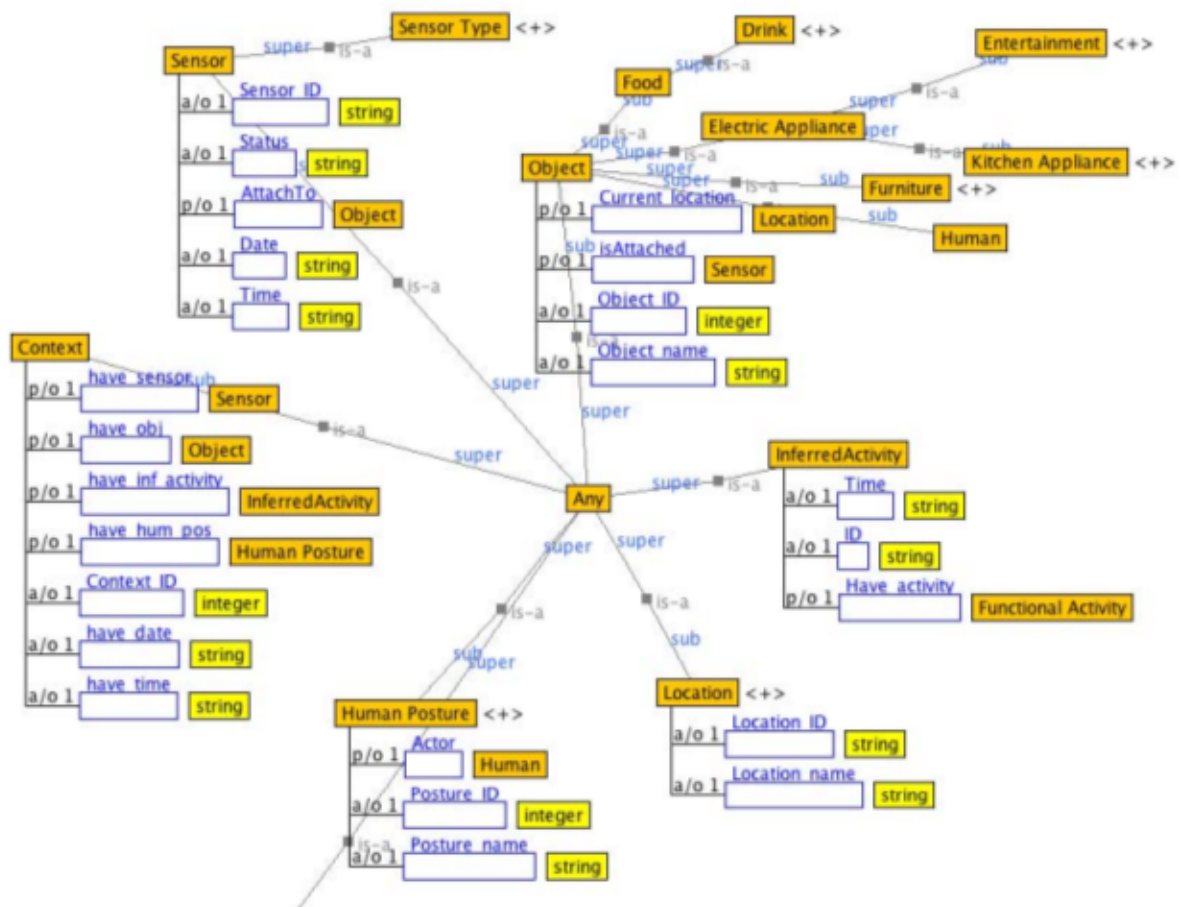


Figura 10: Ontologia de infraestrutura de contexto (OBAR). Fonte: (WONGPATIKASEREE et al., 2012).

usuários. A partir dos dados de observação do usuário, o mesmo só pode utilizar a vassoura para varrer o chão. A regra baseada em objetos para classificar a atividade "Varrer o chão", é definida e mostrada em um termo de linguagem natural para a compreensão, como a seguinte:

$$\begin{aligned}
 \text{Sweep the floor} &\sqsubseteq \text{Functional Activity} \\
 &\sqcap \text{use}(\text{Object.Furniture}(\text{Broom})) \\
 &\sqcap \text{HumanPosture}(\text{Stand})
 \end{aligned}$$

Figura 11: Linguagem natural regra varrer. Fonte: (WONGPATIKASEREE et al., 2012).

Definição da regra na Figura 11, o evento de atividade "Varrer o chão do andar" ocorre quando a vassoura é usada e a postura humana é estar em pé.

No entanto, essa regra representada em linguagem natural que não pode ser entendida pelo sistema, portanto precisa ser reescrita na linguagem Jena, a qual é exibida na Figura 12 a sintaxe da mesma para a atividade "Varrer o chão".

```

Linking_InferredActivity_instance_id_1_to_Context_0: (?x rdf:type ns:
Context) (?x ns:has_have_obj ?y0) (?y0 rdf:type ns:Broom) (?x ns
:has_have_hum_pos ?y1) (?y1 rdf:type ns:Stand) -> (?x ns:
has_have_inf_activity ns:InferredActivity_instance_id_1) (?x ns:
has_rule_name '
Linking_InferredActivity_instance_id_1_to_Context_0')

```

Figura 12: Regra da atividade varrer na linguagem da API JENA. Fonte: (WONGPATI-KASEREE et al., 2012).

O sistema de reconhecimento de atividade baseada em ontologia OBAR, compõe-se de três subcomponentes. Em primeiro lugar, o banco de dados é usado para coletar informações de vários tipos de sensores; segundo, a ontologia é definida como uma semântica do contexto do usuário e da atividade funcional; e por último, as regras DL representam a decisão da atividade baseada no contexto do usuário.

Os dados dos sensores que estão armazenados em uma base de dados relacionais são mapeados utilizando uma ferramenta chamada D2RQ. Esta é usada para transformar em estrutura de descrição de recursos (RDF), modelo padrão para intercâmbio de dados na web semântica.

Para avaliação do OBAR, foram realizados testes utilizando uma base relacional com os dados de entrada de um dia específico, com apenas um usuário no ambiente. Essa entrada é buscada no banco de dados a cada cinco minutos e populada na ontologia com a D2RQ.

A linguagem de consulta "SPARQL", é usada para pesquisar os dados do arquivo RDF, sendo possível recuperar os dados por data, hora, objeto, etc. Ao introduzir-se a data "= 4112012 (11 de abril de 2012)" o sistema SOS exibe a ID de contexto, data, hora, ID do sensor, o objeto de ativação, o nome da postura e a atividade.

3.4 Semantic Smart Homes: Towards Knowledge Rich Assisted Living Environments

Este trabalho apresenta uma arquitetura conceitual de casas inteligentes semânticas e funções elaboradas e explora a interação de componentes que constituem o ambiente, concentrando-se predominantemente na metodologia de modelagem semântica, na geração e gerenciamento de conteúdo (CHEN et al., 2009).

A arquitetura proposta divide-se nas seguintes camadas:

- A camada física consiste em *hardware*, como sensores, atuadores e vários dispositivos, incluindo equipamentos médicos, eletrodomésticos e componentes de rede. Esta camada fornece os meios para monitorar e capturar os eventos e ações de uma Casa Inteligente, e subsequentemente transmiti-los para a camada de dados.

- A camada de dados recebe contexto bruto de fontes heterogêneas que geralmente são díspares em formatos e interfaces de acesso, cada uma delas sendo dedicada a cenários individuais de aplicativos baseados em sensores.
- A camada de aplicação contém vários recursos, ferramentas e subsistemas para a vida assistida. Dentro desta camada, as aplicações processam dados sensoriais que foram passados através da camada de dados e podem ser utilizados para controlar atuadores e/ou instalações multimídia da camada física para oferecer uma vivência assistida, conforme necessário.
- A camada semântica fornece uma visão homogênea sobre dados heterogêneos, permitindo o acesso, o compartilhamento, a integração e a fusão de dados em várias organizações. Isto é possível uma vez que o emprego de ontologias como um *backbone* conceitual unifica a modelagem de representação de dados.
- A camada de serviço inteligente é construída sobre o conteúdo semântico e funcionalidades da camada semântica e a sua finalidade é explorar a semântica e o conhecimento descritivo para fornecer capacidades avançadas de processamento e apresentação de serviços. O primeiro fornece valores adicionados às interfaces de consulta do barramento RDF, pesquisando, analisando e raciocinando mais sobre dados gravados. O último, essencialmente visualiza o conteúdo dos repositórios e as saídas dos serviços de processamento.

A partir da arquitetura proposta o autor descreve a modelagem semântica e como mesma é representada por meio do padrão RDF, que contém um conjunto de triplas, sendo cada tripla constituída pelo sujeito, predicado (propriedade) e objeto. A Figura 13 ilustra o relacionamento entre as diferentes camadas da arquitetura proposta.

Esta estrutura pode ser considerada como uma maneira natural de descrever a grande maioria dos dados processados por máquinas. Uma tripla pode fazer afirmações de que determinadas coisas (como sensores, habitantes) têm propriedades (como "é usado para", "tem um tipo") com certos valores (termostato, demência).

A Figura 14 mostra o gráfico RDF que representa o grupo de instruções: "há um sensor de contato identificado por <http://www.ulster.ac.uk/ssh2008/ssh#Traillabcontactsensor9>, e está localizado no segundo armário *kitchencupboard2* da cozinha, anexado à garrafa de *milkBottle* e ativa o evento *milkmoved*".

O RDF codifica o significado em suas declarações triplas, em que um único item de dados separado não significa nada, portanto o seu significado só pode ser interpretado contra um contexto em que o termo aparece em uma expressão RDF, onde o sujeito, o predicado e o objeto são todos identificados através de *Uniform Resource Identifier* (URI).

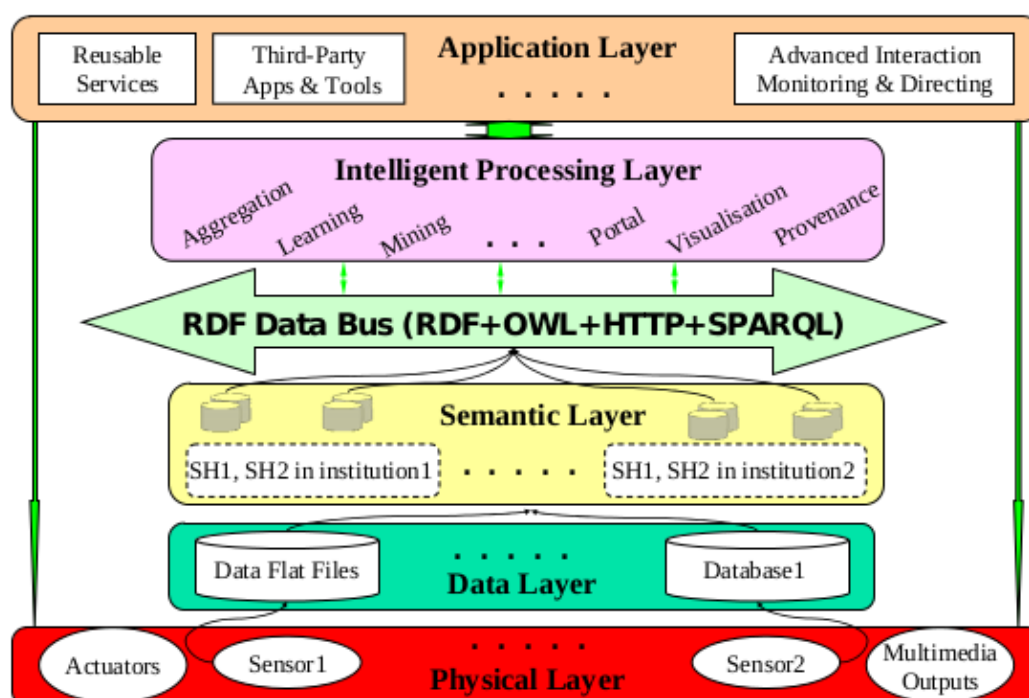


Figura 13: Relacionamento entre as diferentes camadas. Fonte: (CHEN et al., 2009).

O contexto é, em essência, as ontologias que formalmente definem todos os conceitos fundamentais e as relações entre eles. Uma ontologia típica, geralmente contém uma estrutura hierárquica de conceitos e sub-conceitos. As relações entre conceitos são estabelecidas atribuindo propriedades a conceitos e permitindo que os sub conceitos herdem tais propriedades de seus conceitos "pai".

O desenvolvimento da ontologia é um processo formal de aquisição de conhecimento e modelagem. Quando nos referimos a uma Casa Inteligente, deve-se considerá-la como um micro ecossistema complexo, composta pelos aspectos a seguir:

- Um ambiente físico com vários móveis, eletros-eletrônicos e salas que oferecem um espaço de vivência;
- Habitantes que realizam diversas atividades dentro do ambiente;
- Sensores, atuadores e dispositivos médicos para detectar e agir sobre as alterações ambientais e comportamentais.

A partir dessas definições conceituais, os autores utilizaram uma ontologia de ADL própria para validar sua arquitetura conceitual, o qual pôde testar o reconhecimento de atividade com fragmento da ontologia (Fig. 15).

(CHEN et al., 2009), através de suas conclusões, acreditam que com o desenvolvimento de sua pesquisa, foi possível introduzir os conceitos de uma Casa Inteligente semântica, visando derrubar barreiras da heterogeneidade e isolamentos entre

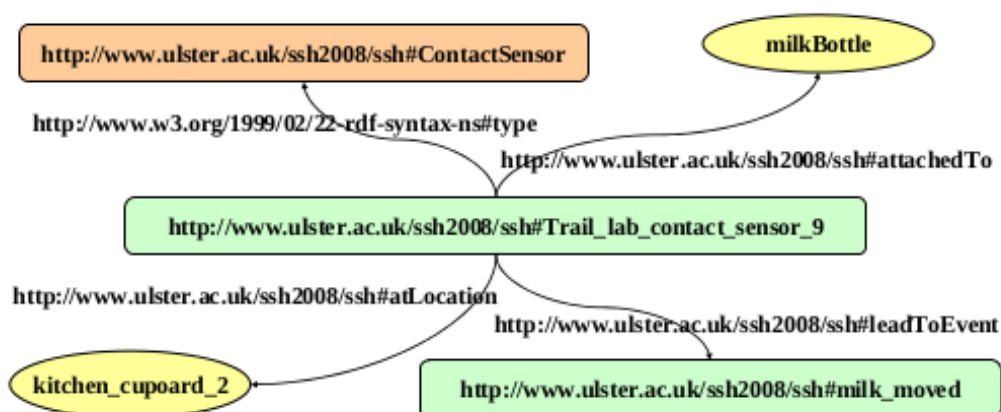


Figura 14: Um gráfico RDF descrevendo um sensor de contato de concreto. Fonte: (CHEN et al., 2009).

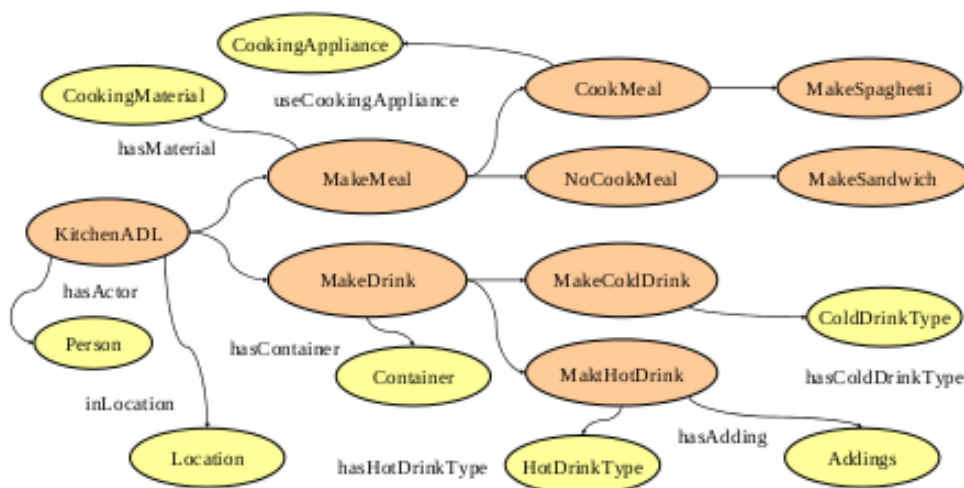


Figura 15: Fragmento da ontologia. Fonte: (CHEN et al., 2009).

dispositivos, dados, capacidades e aplicações, além de desencadear o potencial da abordagem através de semântica, regras e representação expressiva.

3.5 Considerações sobre o Capítulo

Este capítulo apresentou trabalhos que propuseram modelos ontológicos para Reconhecimento de Atividades em uma Casa Inteligente e que se mostraram relacionados à proposta desta dissertação. Foram descritos os trabalhos, buscando caracterizar a sua arquitetura de *software*, como também as abordagens utilizadas para coletar e interpretar o contexto do usuário. Uma análise dos trabalhos relacionados será realizada ao final no capítulo 4, tendo em vista a satisfação dos requisitos para a concepção de abordagens para reconhecimento dos padrões.

4 EXEHDA-AR: ARQUITETURA E FUNCIONALIDADES

Este capítulo apresenta a concepção dos componentes do EXEHDA-AR para tratamento de informações contextuais, visando prover suporte para Reconhecimento de Atividades em Casas Inteligentes empregando processamento semântico. O esforço de concepção destes componentes considerou sua integração com o EXEHDA, levando em conta, portanto, os princípios operacionais do *middleware*. Deste modo, a contribuição central desta dissertação para com o EXEHDA foi ampliar as funcionalidades do seu Subsistema de Reconhecimento de Contexto e Adaptação, de forma a atender demandas inerentes ao Reconhecimento de Atividades.

4.1 EXEHDA-AR: Visão Geral da Proposta

A proposta do EXEHDA-AR contempla os seguintes aspectos na sua concepção: (i) ser distribuída, permitindo a coleta e processamento das informações contextuais em diferentes localizações simbólicas; (ii) ser dirigida por eventos, viabilizando a possibilidade de associação de regras aos contextos de interesse das aplicações, as quais possuem as condições e ações associadas a um evento, sendo estas regras disparadas automaticamente em função de mudanças no estado dos contextos de interesse; e (iii) e viabilizar a utilização de técnicas baseadas em especificação para identificação de atividades. A decisão de integrar esses aspectos na concepção do EXEHDA-AR entende-se que pode prover o necessário suporte para o Reconhecimento de Atividades.

O ambiente ubíquo gerenciado pelo EXEHDA é constituído por células nas quais se distribuem os dispositivos computacionais. Os tipos básicos de dispositivos computacionais são: (i) EXEHDAbase: elemento central da célula responsável por todos serviços básicos e referência para os demais elementos; (ii) EXEHDA nodo: corresponde aos dispositivos computacionais responsáveis pela execução das aplicações; (iii) EXEHDA nodo móvel: tipo específico de EXEHDA nodo que corresponde aos dispositivos móveis que podem se deslocar entre as células do ambiente ubíquo, como *notebooks*, *tablets* ou *smartphones*; e (iv) EXEHDA borda: elemento de borda do am-

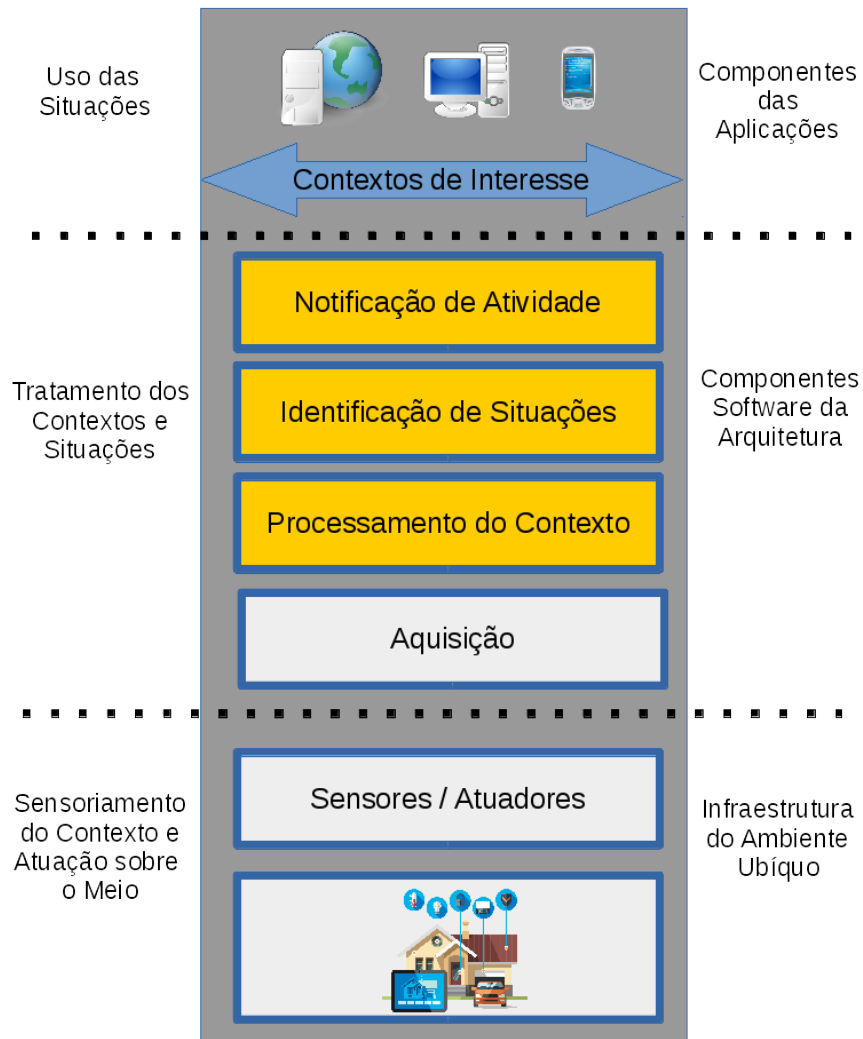


Figura 17: Visão geral da proposta do EXEHDA-AR. Fonte: Dados do Autor.

Assim, a abordagem para Reconhecimento de Atividades no EXEHDA-AR tem por base o Servidor de Contexto, o qual disponibiliza funcionalidades que proveem armazenamento e processamento das informações de contexto.

4.2 Servidor de Contexto do EXEHDA na visão EXEHDA-AR

A descrição dos componentes previstos para o EXEHDA-AR apresentada nas próximas seções, está organizada a partir dos servidores do EXEHDA envolvidos e suas respectivas funcionalidades, sendo realizadas as necessárias associações entre os mesmos.

Os componentes do EXEHDA-AR que foram incorporados aos módulos do Servidor de Contexto interoperam no provimento das funcionalidades para dar suporte para o Reconhecimento de Atividades. Os módulos do Servidor de Contexto modificados estão descritos a seguir, sendo responsáveis pelas etapas que abrangem desde a aquisição do contexto, a partir dos dados coletados através dos Servidores de Borda, até o momento em que este é armazenado e processado com o intuito de identificar situações, cuja ocorrência é notificada às aplicações e/ou aos demais serviços do *middleware* EXEHDA.

Uma visão geral da arquitetura do Servidor de Contexto é apresentada na Figura 18, na qual é caracterizada a relação com (i) Servidores de Borda; (ii) outros serviços do *middleware* EXEHDA; (iii) outros Servidores de Contexto remotos; e (iv) aplicações de usuários e administradores.

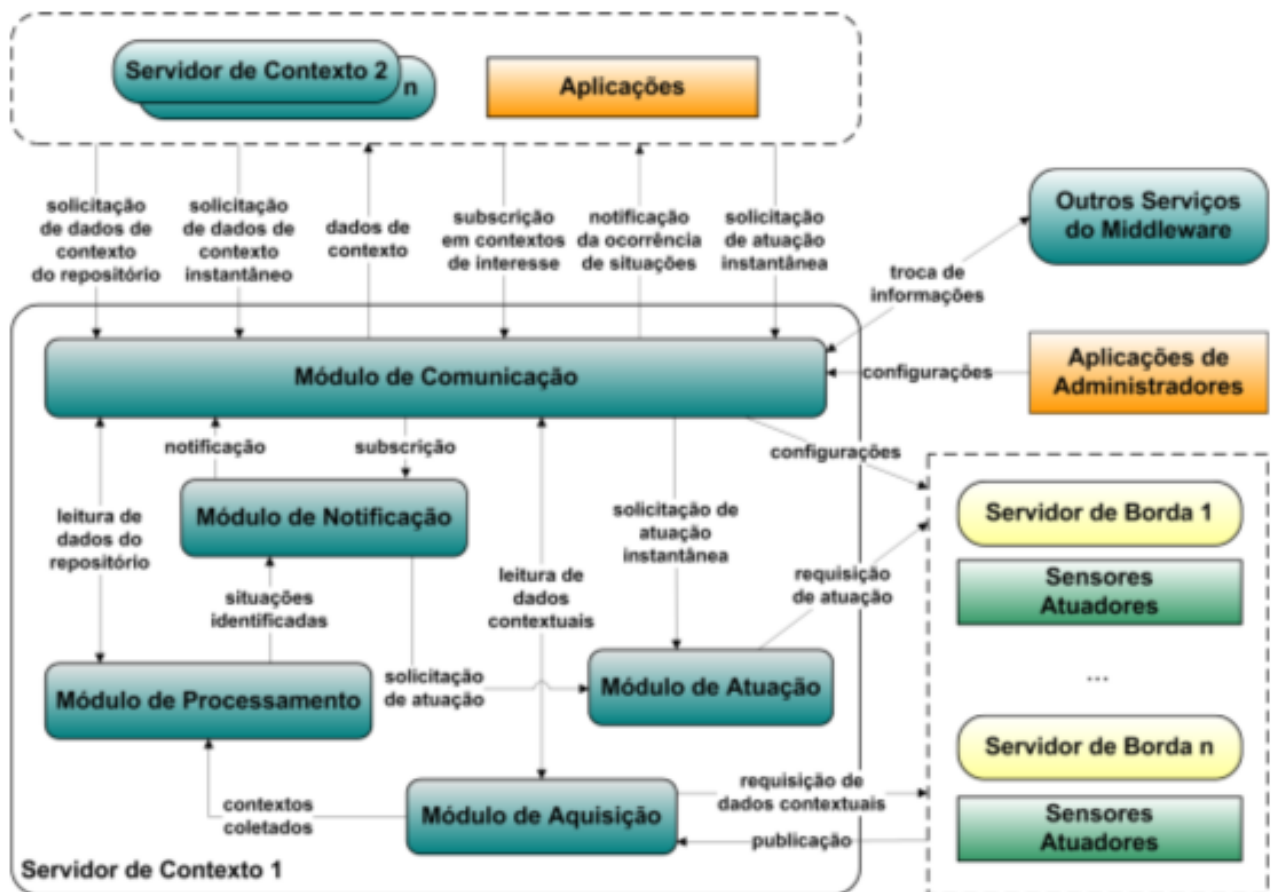


Figura 18: Visão geral da arquitetura do EXEHDA Servidor de Contexto expandido. Fonte: (LOPES et al., 2014).

As contribuições desta dissertação foram introduzidas nos módulos de Aquisição, Processamento, Interpretação e Coleta do *middleware* EXEHDA. Estes módulos modificados serão discutidos nas próximas secções.

4.2.1 Módulo de Aquisição no EXEHDA-AR

O Módulo de Aquisição é o responsável pelo provimento de suporte à captura das informações contextuais coletadas pelos Servidores de Borda considerando dois tipos de sensores: sensores lógicos, implementados através de interfaces de software, e/ou sensores físicos providos por interfaces de hardware. Esse módulo provê suporte tanto para o procedimento de coleta agendada dos dados contextuais, como para a obtenção instantânea dos mesmos por solicitações de consumidores interessados.

A obtenção de dados de contexto pelo Módulo de Aquisição acontece através de publicações do Servidor de Borda, o qual considera os seguintes parâmetros: (i) o intervalo de tempo entre medições; (ii) a flutuação mínima do valor coletado para que aconteça a publicação; e (iii) a inserção na faixa de valores que deverão ser publicados.

Uma abordagem que se considerou necessária para o reconhecimento de atividades, foi analisar o tempo da sequência de eventos, assim descrevendo-os como discretos para identificar padrões de interesse.

O modelo de Janelas Deslizantes utilizado no EXEHDA-AR considerou o trabalho de (SALFNER; LENK; MALEK, 2010) que possui dois tipos janela: (i) baseada em uma quantidade específica de número de eventos que corresponde a um padrão de interesse e; (ii) janela baseada em tempo, que tem um tempo finito onde eventos de interesse são monitorados.

Como as atividades coletadas em uma Casa Inteligente se apresentam de forma dinâmica, foi concebido um algoritmo de janelamento de tempo finito. O algoritmo de janelamento foi implementado em *Python* como objetivo de agrupar informações coletadas, sendo que esse algoritmo oferece opções de aumentar ou diminuir a janela de tempo de acordo com perfil do morador da casa.

A escolha do tamanho da janela para testes no EXEHDA-AR, baseou-se em tamanho como de (KRISHNAN; COOK, 2014) e (RIBONI et al., 2011), que utilizam janelas variando de 60 segundos até 1 hora. A representação gráfica do conceito de janela deslizante de tempo está ilustrado na Figura 19 na qual a janela deslizante aglutina a informação de 5 eventos em um determinado período de tempo determinada Δt .

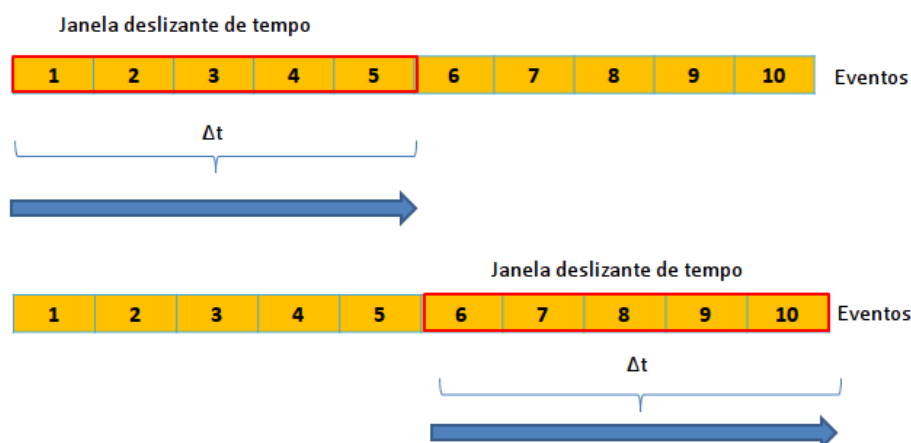


Figura 19: Representação de janelas de tempo deslizante. Fonte: Dados do Autor.

A quantidade de eventos capturados pelo janelamento varia de acordo com os eventos registrados naquele intervalo de tempo, como por exemplo, se a pessoa está se deslocando pela casa, o número do evento será maior do que se pessoa estivesse relaxando no sofá. Feito o janelamento as informações passam para o próximo módulo, o de Processamento.

4.2.2 Módulo de Processamento no EXEHDA-AR

O Módulo de Processamento tem como principal função realizar as tarefas pertinentes ao tratamento das informações contextuais para identificar atividades relacionadas aos contextos de interesse das aplicações. Na Figura 20 é apresentada uma visão geral do Módulo de Processamento do EXEHDA-AR.

Para execução destas tarefas, esse módulo é constituído por quatro componentes cujas suas funcionalidades serão descritas nos próximos itens.

Componente para Gerenciamento de Contexto

Entre as finalidades do componente de Gerenciamento de Contexto, estão as etapas de interpretação e agregação das informações contextuais. Estas etapas tem como objetivo aumentar o grau de abstração para melhorar a disponibilidade e usabilidade na identificação de atividades.

Uma das formas de abstração providas por esse componente é a idempotência, a qual independentemente do número de vezes que sensor detecta eventos (ON/OFF), é considerado que mesmo identificou uma ocorrência em uma janela de tempo válida.

Outra abstração provida por este componente é a normalização do tipo de dado *date time*, cuja as informações são convertidos para milissegundos, com intuito de serem utilizados na construção das regras.

Neste componente são agregadas informações, como identificação do sensor, o horário de início e fim e a identificação da janela em que foi coletado o evento.

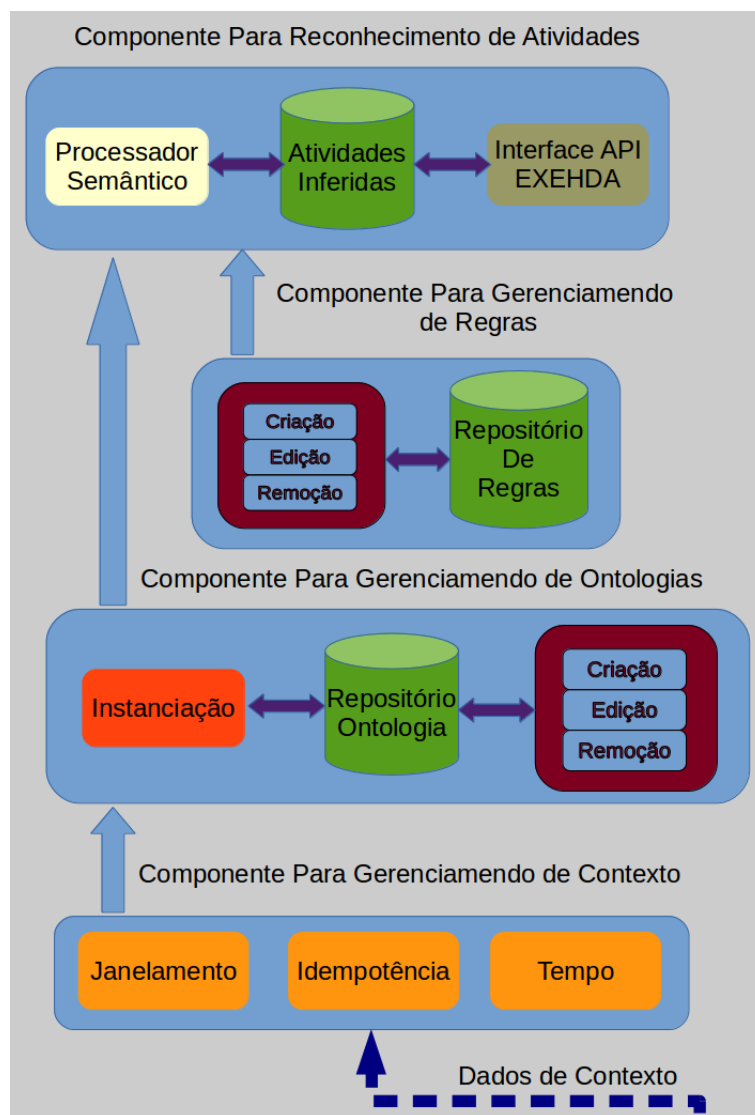


Figura 20: Visão geral do Módulo de Processamento do EXEHDA-AR. Fonte: Dados do Autor.

Componente de Gerenciamento de Ontologias

O componente de Gerenciamento de Ontologias tem por finalidade gerenciar a camada semântica do módulo de processamento do EXEHDA-AR possibilitando instânciação de ontologias. Nesta dissertação, foi desenvolvida uma ontologia para reconhecimento de atividades em uma casa inteligente, a mesma é apresentada na Figura 21.

A ontologia concebida para o reconhecimento de atividades para o EXEHDA-AR, utilizou como base a ontologia do projeto PalSPOT (RIBONI et al., 2011) devido a mesma ser também instanciada em um *middleware* e focada no reconhecimento de atividades.

Foram herdadas da ontologia original as classes *Activity*, *Location*, *Artifact*, *Time* e *Person*, das quais foram criadas sub classes, seus atributos e relacionamentos.

Também foi criada a classe *Windows* com a finalidade de receber as informações do Componente para Gerenciamento de Contexto.

No EXEHDA-AR quando uma atividade é reconhecida, é criada uma nova instância na classe *Activity* relacionando-a com as demais classes para agregar informações relevantes como localização, objeto, segmento do dia e a janela em que a atividade foi reconhecida.

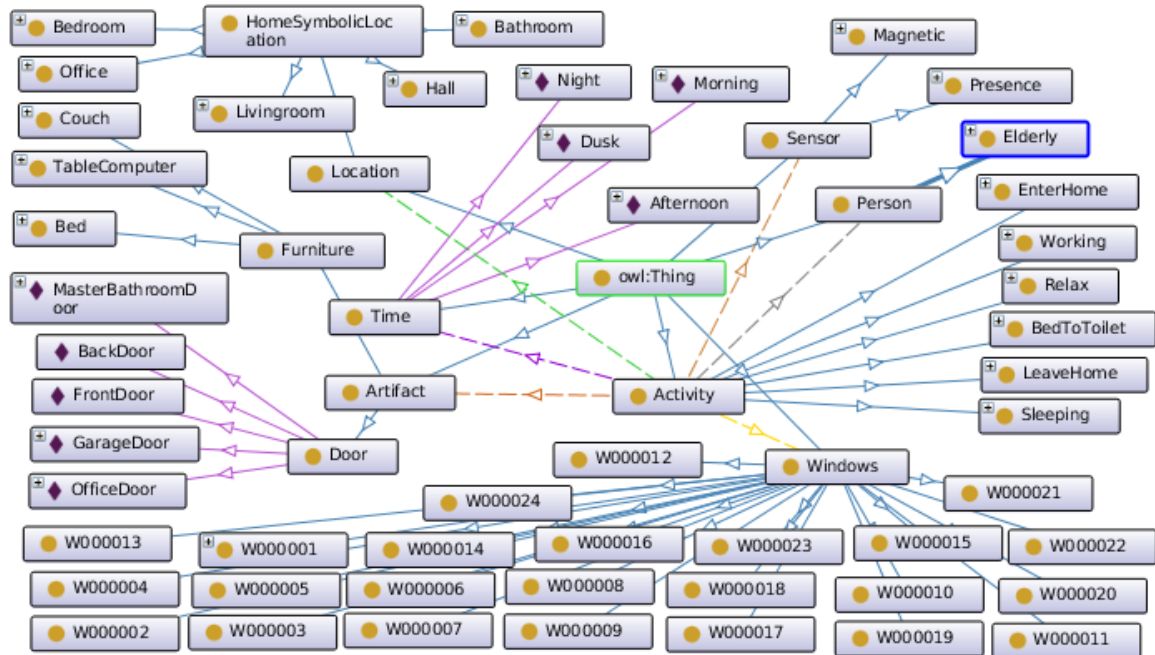


Figura 21: Visão geral da ontologia concebida. Fonte: Dados do Autor.

A ontologia proposta contém 258 axiomas, 51 classes, 6 objetos de propriedades e 11 propriedades de dados. Uma visão parcial da ontologia pode ser visto na Figura 22.

A classe *Activity*, tem subclasses com as atividades que se deseja reconhecer, como *Sleeping*, *Working*, *EnterHome*, *LeaveHome* e *BedToToilet*. Cada vez que uma atividade é reconhecida, o EXEHDA-AR cria uma instância na classe correspondente com as informações do tempo inicial e final da janela, objetos, localização, indivíduo e sensores.

As propriedades de dados recebem os valores oriundos dos sensores. Como exemplo, *hasStatusSensor* recebe o valor "dt" quando há eventos detectados. Devido a utilização da idempotência e do janelamento dos eventos de sensores apenas, recebe o valor de tempo do primeiro e do último evento na janela corrente.

As ontologias empregadas pelo EXEHDA-AR são armazenadas em disco no repositório de ontologias e gerenciadas por meio da *Application Programming Interface* (API) JENA, um *framework* Java, de código aberto para desenvolvimento de aplicações com suporte à Web Semântica e as linguagens RDF, OWL e *SPARQL Protocol and RDF Query Language* (SPARQL), incluindo uma máquina de inferência

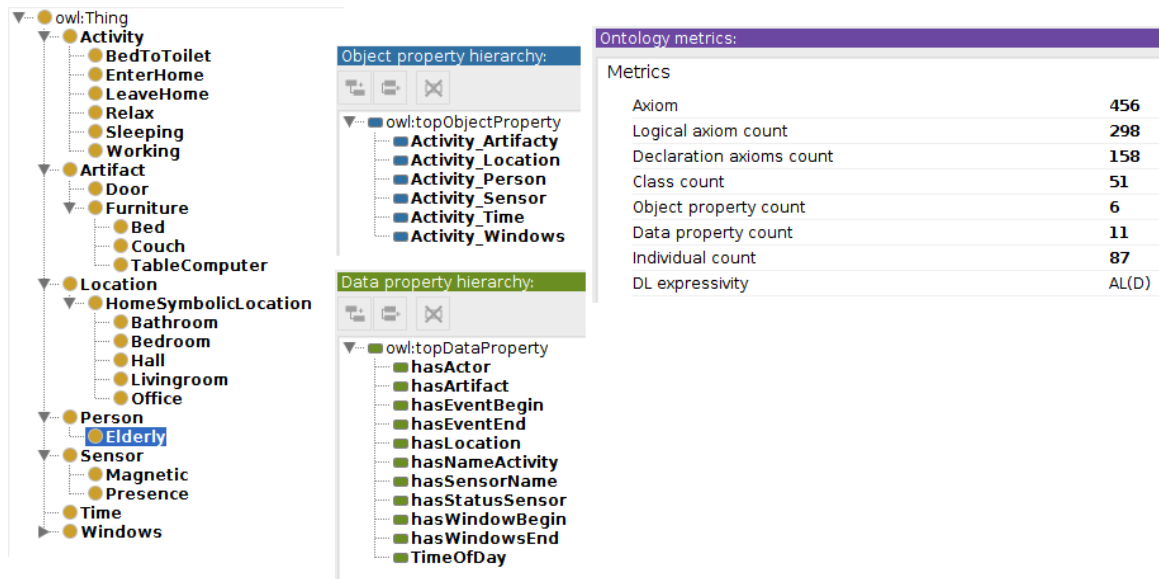


Figura 22: Classes e atributos da ontologia proposta. Fonte: Dados do Autor.

baseada em regras (RIBONI; BETTINI, 2011).

API JENA foi empregada neste trabalho devido a sua simplicidade na implementação, por conter uma vasta documentação além de ser facilmente integrada em diferentes arquiteturas.

Componente para Gerenciamento de Regras

As regras foram construídas em *Semantic Web Rule Language* (SWRL) (OKEYO et al., 2014) utilizando os conceitos propostos por (CHEN; NUGENT, 2009), através da relação entre a localização simbólica e objeto para reconhecer qual atividade o indivíduo está realizando. Essas regras são processadas em nível de instâncias e, desta forma, motores de inferência conseguem raciocinar sobre as informações contextuais.

Essas regras são armazenadas no componente de gerenciamento de regras o qual forma um repositório, que possibilita ao desenvolvedor a criação, edição e remoção das mesmas. Na Figura 23, é apresentada a aplicação dos conceitos de (CHEN; NUGENT, 2009) para a concepção de regras.

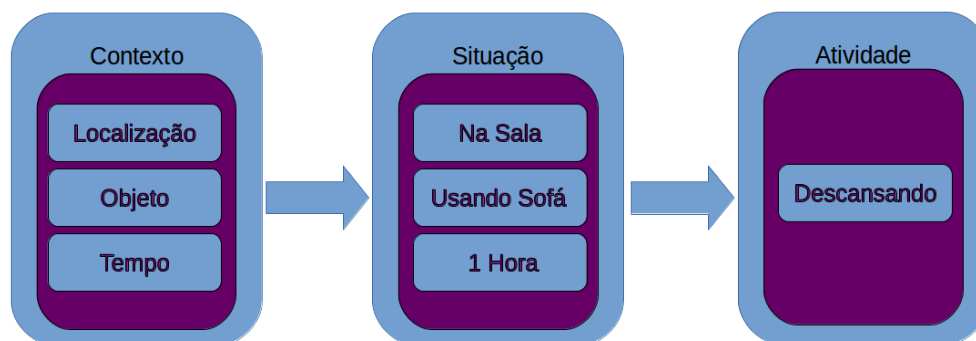


Figura 23: Concepção de regras para o EXEHDA-AR. Fonte: Dados do Autor.

Considerando que as regras estão associadas aos objetos do ambiente e a localização na casa a partir dessas relações, é possível identificar a atividade que está sendo realizado pelo usuário.

No capítulo 5 as regras elaboradas para EXEHDA-AR são apresentas, quanto suas características e contextos de interesse.

Componente para Reconhecimento de Atividade

O componente para o Reconhecimento de Atividade disponibiliza as funcionalidades necessárias para o processo de raciocínio sobre os dados contextuais, visando à identificação das atividades.

No presente componente as regras realizam a inferência nas instâncias da classe *Windows*, assim criando novas instâncias nas classes referentes as atividades através da função `swrlx:makeOWLThing(?y, ?x)`.

Na Figura 24 é apresentado uma instância criada quando a regra para reconhecer a atividade *working* tem suas condições atendidas, em que as informações relevantes são relacionadas a instância criada, como localização, objeto, sensores envolvidos na atividade.

```
<owl:NamedIndividual rdf:about="http://www.semanticweb.org/activity.owl#AWK00001">
  <rdf:type rdf:resource="http://www.semanticweb.org/activity.owl#Working"/>
  <Activity_Artifact rdf:resource="http://www.semanticweb.org/activity.owl#OfficeDoor"/>
  <Activity_Artifact rdf:resource="http://www.semanticweb.org/activity.owl#TablePC"/>
  <Activity_Location rdf:resource="http://www.semanticweb.org/activity.owl#Office"/>
  <Activity_Person rdf:resource="http://www.semanticweb.org/activity.owl#Maria"/>
  <Activity_Sensor rdf:resource="http://www.semanticweb.org/activity.owl#M026"/>
  <Activity_Sensor rdf:resource="http://www.semanticweb.org/activity.owl#M028"/>
  <Activity_Time rdf:resource="http://www.semanticweb.org/activity.owl#Morning"/>
  <hasNameActivity>working</hasNameActivity>
  <hasWindowBegin rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2010-11-06T10:00:00Z</hasWindowBegin>
  <hasWindowsEnd rdf:datatype="http://www.w3.org/2001/XMLSchema#dateTime">2010-11-06T10:59:59Z</hasWindowsEnd>
</owl:NamedIndividual>
```

Figura 24: Instância criada quando uma atividade é reconhecida. Fonte: Dados do Autor.

4.2.3 EXEHDA-AR: Servidor de Borda

Utilizou-se a arquitetura já concebida do *middleware*, que é formada por módulos destinados a tratar a rede de sensores e/ou atuadores, bem como efetuar as publicações dos dados contextuais coletados no Servidor de Contexto. Dentre os módulos do Servidor de Borda, o EXEHDA-AR realizou algumas contribuições que serão descritas nas próximas seções. Na Figura 25 é apresentada a arquitetura do Servidor de Borda.

Módulo Interoperação

O Módulo de Interoperação é destinado a gerenciar a comunicação com o Servidor de Contexto, tendo como referência o estilo arquitetural REST. Com isso, todas as funcionalidades associadas aos sensores e atuadores são mapeadas na forma de recursos e acessadas via URIs.

Através desse módulo é disponibilizada uma interface padronizada para manipulação dos dispositivos conectados ao Servidor de Borda, abrangendo aspectos como: coleta, configuração, acionamento, inserção e remoção de sensores ou atuadores.

A manipulação destes dispositivos pode ocorrer a partir de diferentes serviços do *middleware*. Essa funcionalidade do Módulo de Interoperação é importante para o desenvolvimento de aplicações em ambientes ubíquos, altamente distribuídos e heterogêneos, pois restringe ao Servidor de Borda o tratamento dos aspectos tecnológicos envolvidos na interação com o meio, evitando que isso se reflita no restante da arquitetura.

Módulo de Coleta

Esse módulo provê o tratamento de uma rede de sensores permitindo que cada sensor possa ser lido por parâmetros individualmente configurados. Este tratamento é responsável por aspectos que vão desde a gerência física (interfaces, frequência de leitura) até a normalização computacional (validação, tradução) dos valores coletados. Também, este módulo provê funcionalidades para publicação das informações coletadas pela rede de sensores no Servidor de Contexto.

O componente **Parâmetros Operacionais dos Sensores** especifica o *driver* a ser utilizado para leitura dos diferentes sensores, bem como a agenda de aquisição dos dados a ser praticada. Esta agenda trata individualmente os sensores, permitindo a especificação de leituras periódicas e/ou atreladas a datas e/ou horários específicos. O componente **Agendador**, a partir dos Parâmetros Operacionais dos Sensores, dispara leituras considerando o interesse do usuário. O Agendador emprega o relógio do Servidor de Borda para o disparo dos seus procedimentos, sendo possível especificar a frequência com que estes disparos ocorrem.

O *driver* do sensor é responsável pela aquisição do valor referente à grandeza física medida pelo sensor. Sendo usual cada tipo de sensor possuir uma forma específica de acesso para obtenção dos seus dados, a estratégia de encapsular a operação do *driver* do sensor tem por objetivo evitar que as diferenças operacionais de cada *driver* se projetem nas outras camadas de *software* da arquitetura.

Dentre outros aspectos, este encapsulamento contribui para a manutenção (troca de sensores, atualização de *drivers* por parte do fabricante).

O componente **Validador** avalia se uma determinada coleta de dados sensorizados

deve ser publicada ou não, empregando para isso critérios especificados pelo usuário.

Esses critérios são baseados em regras que estão identificadas no componente Parâmetros Operacionais dos Sensores, a partir do ID do sensor (IDs). Por exemplo, com o intuito de minimizar custos de comunicação na rede, um valor sensoriado, cuja diferença entre o último valor lido e o antepenúltimo for menor que a precisão operacional do sensor, a critério do usuário, poderá não ser publicado.

O componente **Tradutor** objetiva adequar o dado coletado à natureza da aplicação do usuário, por exemplo, faixas de temperatura podem ser convertidas em descrições do tipo ?Muito Alta?, ?Alta?, ?Média?, ?Baixa?, através de uma regra que dispara alertas caso a temperatura esteja muito alta (acima de 55°C) oferecendo riscos. O componente **Leitura Instantânea** tem o objetivo de permitir a leitura de um determinado sensor sob demanda das aplicações, a qualquer momento. As requisições de leitura podem ser provenientes de aplicações de usuários ou de Servidores de Contexto como consequência da execução de regras.

Este componente recebe de forma assíncrona as solicitações e, a partir do ID do sensor, dispara o *driver* correspondente.

O componente Tratador de Regras Locais é responsável por tratar regras de contingência, com a intenção de evitar que os dispositivos envolvidos atinjam estados críticos. Estas regras atuam sobre o mecanismo de atuação, ativando ou desativando atuadores. Esse componente é ativado sempre que ocorrer a leitura de um determinado sensor, adquirindo elevada importância quando a comunicação com o Servidor de Contexto for interrompida por problemas de infraestrutura de rede.

4.3 Discussão dos Trabalhos Relacionados

Nesta seção são discutidos os trabalhos relacionados. Na discussão foram empregados os seguintes critérios: (i) proposição de um modelo semântico específico, (ii) forma de avaliação do modelo, (iii) verificação da acurácia, (iv) exploração de conceitos de objeto e localização e (v) qual tipo de regras é empregado para o reconhecimento de atividades.

A partir dos critérios acima, foi elaborada uma tabela comparativa entre os trabalhos, considerando a legenda abaixo:

- I *A Context-Aware System Infrastructure for Monitoring;*
- II *AAL Domain Ontology for Event-based Human Activity Recognition;*
- III *Activity Recognition using Context-Aware Infrastructure Ontology in Smart Home Domain;*
- IV *Semantic Smart Homes: Towards Knowledge Rich Assisted Living Environments.*

V EXEHDA-AR.

Tabela 2: Comparação dos Trabalhos Relacionados

Tranlhos Re-lacionados	Propõe Modelo Ontológico para ADL	Utiliza dataset	Verifica a Acurácia	Explora Objeto e Localização	Tipo de Regras
I	Baseado em Modelo Existente	Não	Não	Parcialmente	JENA
II	Baseado em Modelo Existente	Não	Não	Parcialmente	JENA
III	Propõe Modelo Próprio	Não	Não	Utiliza	JENA
IV	Propõe Modelo Próprio	Não	Não	Utiliza	AXIOMA
V	Baseado em Modelo Existente	Sim	Sim	Utiliza	JENA e SWRL

Os trabalhos I, II e V basearam-se em ontologias já existentes, sendo que o V (EXEHDA-AR), proposta desenvolvida nesta dissertação de mestrado, tem o seu modelo ontológico baseado na ontologia PalSPOT.

Por sua vez, os trabalhos I e II, são baseados nas ontologia DOLCE e DnS combinadas, nos quais as ontologias são empregadas tanto para o Reconhecimento de Atividades como no gerenciamento da própria infra estrutura do ambiente inteligente. No caso do EXEHDA-AR, o gerenciamento do ambiente é feito por outros serviços do *middleware*.

Em contrapartida os trabalhos III e IV desenvolveram seus modelos apenas utilizando os conceitos de localização e objeto, sendo que trabalho IV emprega somente uma ontologia assim como EXEHDA-AR; já o trabalho III utiliza duas ontologias uma para gerir os contexto da Casa Inteligente e outra para reconhecer atividades.

Entre os trabalhos, o V (EXEHDA-AR), utiliza um *dataset* público para avaliar sua proposta simulando os contextos coletados, enquanto os trabalhos II, III e IV simulam atividades sem que as mesmas tenham sido coletadas de um ambiente real.

O trabalho III faz uso de um *framework* chamado DB2R para criar suas ontologias a partir de uma base de dados relacional, realizando o reconhecimento analisando os *logs* de uma Casa Inteligente por meio de regras de descrição, o que deixa próximo a abordagem do EXEHDA-AR.

Por sua vez o V (EXEHDA-AR) simula a coleta dos dados pelo servidor de borda do *middleware* EXEHDA, de acordo com o dado coletado e a natureza da aplicação do usuário, por exemplo, faixas de temperatura podem ser convertidas em descrições do tipo ?Alta?, ?Média?, ?Baixa?, através de uma regra.

Os demais trabalhos teriam que criar esses conceitos na ontologia ou ontologias como é caso do II e III, para realizar essa conversão e abstrair o contexto para nível mais alto.

Quanto as ontologias dos trabalhos I, II e IV, não são fornecidas informações relevantes quanto às suas classes e de como se relacionam, como são suas propriedades de dados, além de não descreverem suas regras de descrição lógicas; o trabalho III tem boa parte da sua ontologia descrita mas não a disponibiliza assim como ou-

tros, para que sejam realizados testes como a mesma, assim impossibilitando que se estabeleça uma métrica de comparação entre os modelos ontológicos.

Nenhum dos trabalhos verificou a acurácia com relação à inferência em atividades diárias, uma vez que esta medida representa a qualidade do modelo ontológico frente aos desafios no Reconhecimento de Atividades humanas por meio de sensores não intrusivos.

4.4 Considerações sobre o Capítulo

As premissas de concepção da EXEHDA-AR consideram os aspectos operacionais do *middleware* EXEHDA, bem como buscam contemplar uma abordagem distribuída, dirigida por eventos e regras no que tange a operação da arquitetura. Foi empregado um modelo baseado em especificação, com uma ontologia de domínio para o processamento dos contextos de interesse visando à identificação das atividades relacionadas ao ambiente.

Este capítulo descreveu a abordagem para tratamento do contexto definida para a Arquitetura EXEHDA-AR, a qual prevê dois tipos de servidores: (i) o Servidor de Contexto que provê armazenamento e processamento das informações contextuais, viabilizando o processo de raciocínio para identificação de atividades; e (ii) o Servidor de Borda que é responsável por coletar e pré-processar os dados contextuais, publicando-os no **Servidor de Contexto**.

Os módulos previstos para a Arquitetura EXEHDA-AR, foram descritos neste capítulo, tendo sido organizados em função dos servidores de **Contexto** e de **Borda**, bem como suas respectivas funcionalidades. Ainda, foi apresentada a estrutura de processamento dos dados contextuais concebida para o Servidor de Contexto, a qual contempla o emprego de ontologias, API Jena e do *Reasoner* Pellet para processar os dados coletados pelos **Servidores de Borda**.

No próximo capítulo, é apresentada a avaliação das funcionalidades do EXEHDA-AR, tendo por base cenário de uma Casa Inteligente, a ontologia de domínio elaborada para avaliar a arquitetura proposta.

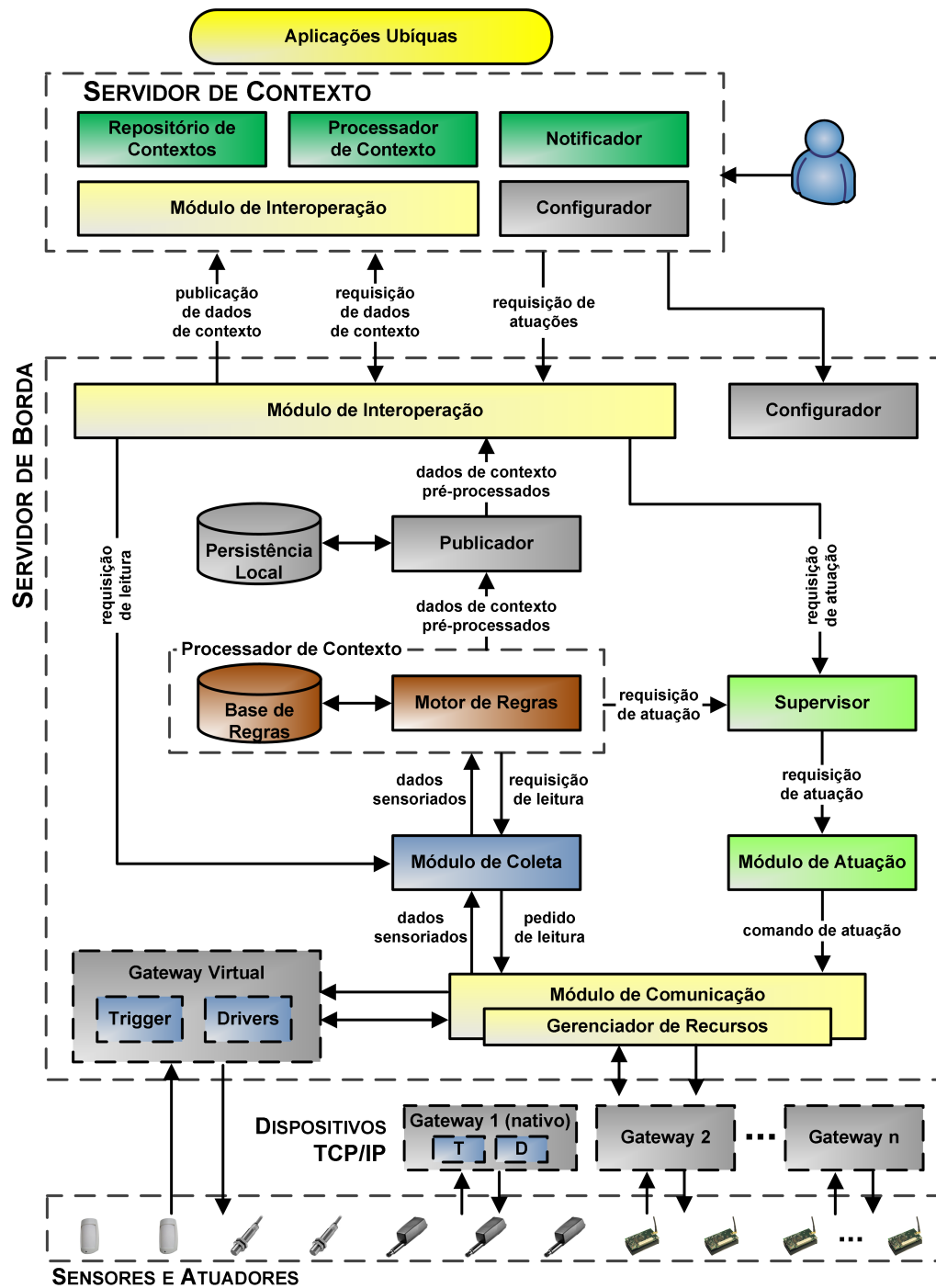


Figura 25: Servidor de Borda. Fonte: (DAVET et al., 2015).

5 ESTUDO DE CASO

Neste capítulo será apresentado o estudo de caso desenvolvido com o objetivo de explorar funcionalidades propostas para o EXEHDA-AR.

As funcionalidades exploradas neste estudo de caso contemplam tarefas relacionadas a aquisição e processamento de dados contextuais, e o decorrente reconhecimento de atividades.

De modo mais específico o estudo de caso trata as contribuições possíveis com a utilização do EXEHDA-AR para a área de **Reconhecimento de Atividades**, uma área que possui um grande potencial econômico e social, que vem motivando um aporte significativo de pesquisas e investimentos em tecnologia direcionados a Casa (Ambientes Assistido de Vivência) para pessoas idosas, para que estas tenham vidas mais confortáveis e independentes (AL-SHAQI; MOURSHED; REZGUI, 2016).

Foram sistematizadas algumas características das estratégias utilizadas pelos trabalhos relacionados com intuito de identificar tendências no tema da pesquisa, que viessem a contribuir na concepção do EXEHDA-AR.

Para realização da avaliação do EXEHDA-AR foram pesquisados na literatura *datasets* que tenha sido empregados em projetos reconhecidos internacional e nacionalmente, elaborou-se regras de descrição lógica em SWRL para o reconhecer atividades realizadas por um habitante de uma Casa Inteligente.

5.1 *Datasets* Considerados

Esta dissertação teve como critério de seleção do *datasets* empregados para avaliação do EXEHDA-AR, o mesmo ter sido produzido por uma casa inteligente habitada por uma única pessoa.

As várias casas inteligentes prototipadas no cenário internacional, produziram diversos *datasets* a partir do seu emprego em diferentes condições.

Os esforços de pesquisa em casas inteligentes concentram-se principalmente em dois grupos diferentes.

No primeiro grupo, existem estudos em que centenas de sensores e aparelhos

domésticos equipados com sensores são implantados em um laboratório de casas inteligentes (KIENTZ et al., 2008), (HELAL et al., 2005), (INTILLE et al., 2005). Esses estudos geralmente se concentram em interações humanas com os futuros ambientes inteligentes.

O segundo grupo de estudos, focaliza a detecção da atividade para o monitoramento do estado de saúde, sendo esta linha seguida pelos pesquisadores (COOK, 2012), (VAN KASTEREN et al., 2008), (GALLISSOT et al., 2011), (TAPIA; INTILLE; LARSON, 2004). Também se concentra em tornar as casas de hoje, inteligentes o suficiente para fornecer acompanhamento de saúde a longo prazo para as pessoas que vivem sozinhas em suas moradias.

Para este último grupo, foi feita uma revisão de literatura para identificar o perfil dos *datasets* produzidos no que diz respeito o número de casas que são monitoradas, suas características, o perfil do morador, a duração da coleta de dados, o número de sensores, as atividades que foram monitoradas e o número de ocorrências das atividades. A Figura 26 resume estas informações obtidas em artigos científicos e/ou documentações técnicas dos *datasets*.

	Nº de Casas	Nº de Residentes	Duração	Sensores	Atividades	Nº de Ocorrências
ARAS	3	Mais de 1	2 meses	20	27	1021 - 2020
CASAS	7	1 ou mais	2-8 meses	20 - 86	11	6 - 1513
UvA	3	1	3 meses	14 - 21	11 - 16	200-300
DOMUS	1	1	10 dias	78	5	NA
MIT	3	1	14 dias	77 -84	9-13	176 - 278

Figura 26: Comparação entre datasets produzidos por casas inteligente. Fonte: Dados do Autor.

Dentre os *dataset* pesquisados, foi selecionado um pertencente ao grupo de Pesquisas CASAS, Universidade de Washington. Este *dataset* é proveniente de uma casa denominada Aruba no qual há uma senhora idosa residindo sozinha, sendo que este apresenta características alinhadas com as motivações desta dissertação. Essa casa tem 9 atividades alvo monitoradas, que são: dormir, locomover-se da cama para banheiro, relaxar na sala de estar, trabalhar no escritório, lavar os pratos, preparar alimentos, comer, entrar em casa, sair de casa. Figura 28 mostra a distribuição dos sensores na casa Aruba.

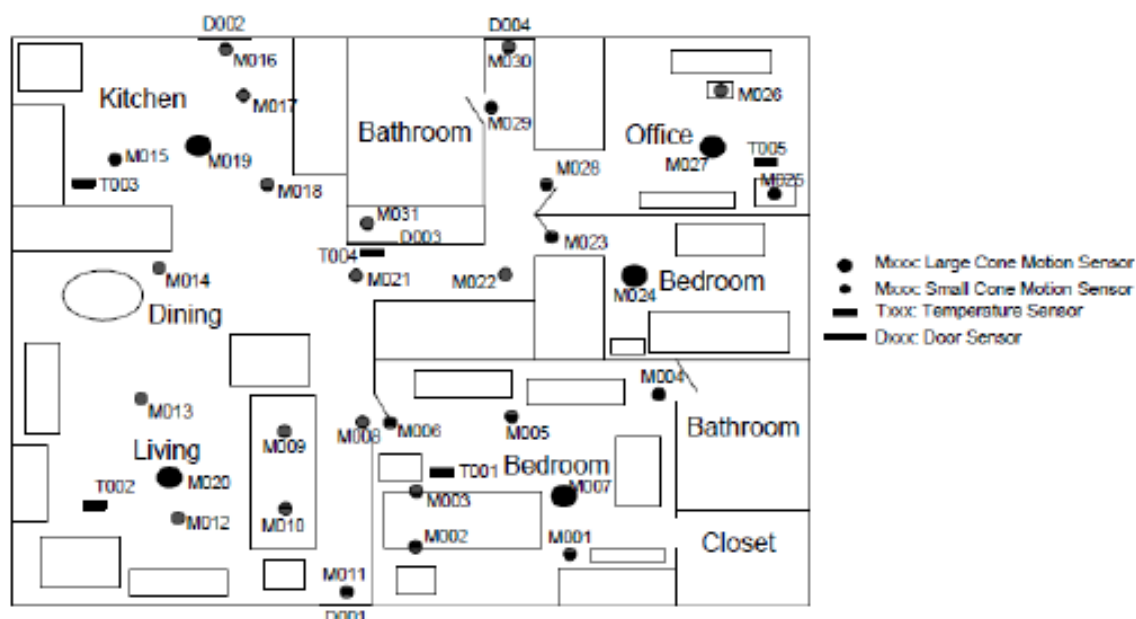


Figura 27: Distribuição dos sensores casa Aruba. Fonte: (YALA; FERGANI; FLEURY, 2015).

Na Figura ?? pode-se visualizar como os sensores são instalados e distribuídos em uma casa instrumentada pelo grupo CASAS.



Figura 28: Distribuição dos sensores casa Aruba. Fonte: (YALA; FERGANI; FLEURY, 2015).

Destas 9 atividades alvos, foram escolhidas 5 delas para modelar as regras de descrição lógica, sendo as seguintes: entrar em casa, sair de casa, dormir, ir da cama para banheiro e trabalhar no escritório. Para a elaboração das regras, foram empregados métodos descritos no trabalho (CHEN; NUGENT, 2009), que utiliza a relação entre localidade, objeto e tempo.

Representação em 3D da casa inteligente Aruba concebido por (YALA; FERGANI; FLEURY, 2015) vide na Figura 29:



Figura 29: Visão da casa Aruba em 3D. Fonte: (YALA; FERGANI; FLEURY, 2015).

O estudo de caso foi segmentado em 5 cenários, usuais no dia-a-dia de um morador típico, portanto com um maior número de ocorrências na casa inteligente Aruba, sendo que cada uma corresponde a uma das atividades alvo que se busca reconhecer.

Cenário 1: Atividade dormir

Neste primeiro cenário, empregando as atividades coletadas do *dataset* Aruba, é reconhecida a atividade dormir. A moradora da casa não tem seu nome revelado por questões de privacidade, portanto, atribuiu-se o pseudônimo de Maria.

Maria dirige-se para quarto de dormir, passando pela porta do quarto e deita-se na cama. Para inferir essa atividade considerou-se a localização neste cenário é o quarto de dormir, o objeto de interação é a cama (representada pelo sensor M003) e a interação da pessoa com objeto deve ser maior que 120000 milissegundos, equivalente a 20 minutos. Assim sendo, foi detectado, que a mesma está dormindo.

Em destaque a distribuição dos sensores do quarto na Figura 34.

Observou-se a distribuição dos sensores pelo quarto e foi definido que o contexto de interesse é o sensor detectando presença M003, e enquanto os sensores M005, M006, M004 não devem detectar presença assim indicando que pessoa permanece na cama.

Na figura 31 é possível observar a regra que foi elaborada para reconhecer a atividade dormir na casa Aruba.

Utilizando essa regra, o EXEHDA-AR consegue reconhecer 398 das 401

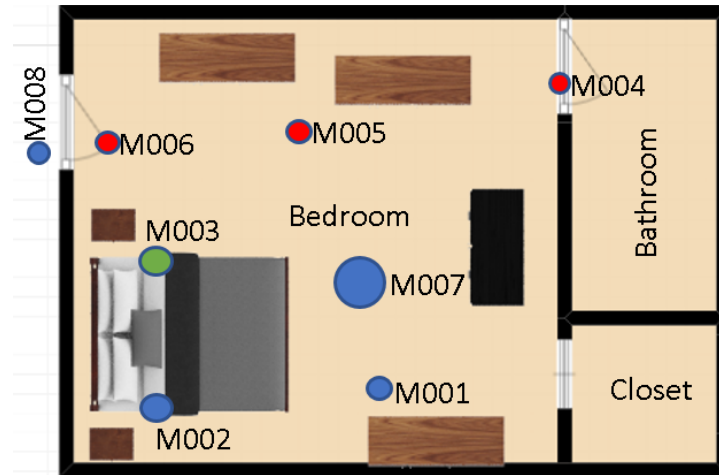


Figura 30: Destacando o sensores do quarto de Dormir. Fonte: Dados do Autor.

Name
Sleeping
Comment
Status
Ok
<code>Windows(?w) ^ hasSensorName(?w, ?wsn) ^ swrlb:equal(?wsn, "M003") ^ hasStatusSensor(?w, ?wss) ^ swrlb:equal(?wss, "dt") ^ hasEventBegin(?w, ?eb) ^ hasEventEnd(?w, ?ee) ^ hasSensorName(?w, ?wsn1) ^ swrlb:equal(?wsn1, "M004") ^ hasStatusSensor(?w, ?wss1) ^ swrlb:equal(?wss1, "ndt") ^ hasSensorName(?w, ?wsn2) ^ swrlb:equal(?wsn2, "M005") ^ hasStatusSensor(?w, ?wss2) ^ swrlb:equal(?wss2, "ndt") ^ hasSensorName(?w, ?wsn3) ^ swrlb:equal(?wsn3, "M006") ^ hasStatusSensor(?w, ?wss3) ^ swrlb:equal(?wss3, "ndt") ^ swrlb:subtract(?d, ?ee, ?eb) ^ swrlb:greaterThan(?d, 1200000) ^ Sensor(?s) ^ hasSensorName(?s, ?sn) ^ swrlb:equal(?sn, ?wsn) ^ hasSensorName(?s, ?sn1) ^ swrlb:equal(?sn1, ?wsn1) ^ hasSensorName(?s, ?sn2) ^ swrlb:equal(?sn, ?wsn2) ^ hasSensorName(?s, ?sn3) ^ swrlb:equal(?sn3, ?wsn3) ^ Location(?l) ^ hasLocation(?l, ?lt) ^ swrlb:equal(?lt, "Bedroom") ^ Artifact(?ar) ^ hasArtifact(?ar, ?at) ^ swrlb:equal(?at, "Bed") ^ Person(?p) ^ hasWindowBegin(?w, ?wb) ^ hasWindowsEnd(?w, ?we) ^ Activity(?a) ^ hasNameActivity(?a, ?an) ^ swrlb:equal(?an, "sleeping") ^ Person(?p) ^ hasActor(?p, ?at) ^ swrlb:equal(?at, "Maria") ^ swrlb:makeOWLThing(?a, ?an) -> Activity(?an) ^ Activity_Windows(?an, ?w) ^ Activity_Sensor(?an, ?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifact(?an, ?at) ^ hasWindowBegin(?an, ?w) ^ hasWindowsEnd(?an, ?w)</code>

Figura 31: Regra Dormir SWRL. Fonte: Dados do Autor.

ocorrências registradas na documentação do *dataset* obtendo nessa atividade, uma acurácia de 99,25%.

Após a realização das inferências é possível realizar uma busca semântica nos dados através da linguagem SPARQL, para obter mais informações sobre a atividade reconhecida. O código da consulta SPARQL pode ser visto na Figura 33.

É apresentado na figura 33 o resultado da consulta SPARQL, na qual são exibidas diferentes informações referentes a atividade *Dormir* como localização, objeto, o ator da atividade, sensores envolvidos, nome da atividade e a instância da janela na qual a atividade foi reconhecida.

A coluna *act* representa a instância criada na classe *Sleeping* da ontologia proposta quando satisfaz as condições determinada na regra, em que as instancias

```

SELECT *
WHERE {
    ?act rdf:type ont:Activity .
    ?act ont:hasNameActivity ?name .
    ?act ont:hasNameActivity 'sleeping' .
    ?act ont:Activity_Location ?location .
    ?act ont:Activity_Person ?person .
    ?act ont:Activity_Artifact ?artifact .
    ?act ont:Activity_Sensor ?sensor .
}

```

Figura 32: Consulta SPARQL para atividade Dormir. Fonte: Dados do Autor.

ASL00001 e ASL00002 correspondem a atividade *Dormindo* em janelas de tempo diferentes.

act	name	location	person	artifact	sensor
ont:ASL00001	"sleeping"	ont:MasterBedroom	ont:Maria	ont:Doublebed	ont:M004
ont:ASL00001	"sleeping"	ont:MasterBedroom	ont:Maria	ont:Doublebed	ont:M003
ont:ASL00001	"sleeping"	ont:MasterBedroom	ont:Maria	ont:Doublebed	ont:M005
ont:ASL00001	"sleeping"	ont:MasterBedroom	ont:Maria	ont:Doublebed	ont:M006
ont:ASL00002	"sleeping"	ont:MasterBedroom	ont:Maria	ont:Doublebed	ont:M004
ont:ASL00002	"sleeping"	ont:MasterBedroom	ont:Maria	ont:Doublebed	ont:M003
ont:ASL00002	"sleeping"	ont:MasterBedroom	ont:Maria	ont:Doublebed	ont:M005
ont:ASL00002	"sleeping"	ont:MasterBedroom	ont:Maria	ont:Doublebed	ont:M006

Figura 33: Resultado consulta atividade Dormir SPARQL. Fonte: Dados do Autor.

Cenário 2: Atividade Deslocamento da Cama para o Banheiro

Neste cenário Maria acorda no meio da noite e sai da cama para banheiro. Para detecção dessa atividade foram selecionados os sensores M003 (cama) e M004 (banheiro), vide Figura 34.

Foi necessário fazer uma sequência temporal dos eventos dos sensores. Para isso foram feitas comparações entre os tempos de cada sensor envolvido no contexto de interesse. O tempo inicial da ativação do sensor do banheiro precisa ser maior que o tempo final do sensor da cama, assim pode-se constatar que Maria saiu da cama para o banheiro.

Durante os testes foi avaliado que a não utilização dos sensores M005 e M007 na regra para detectar a atividade da *Cama para o Banheiro* obteve-se um ganho de

5% de acurácia devido que estes sensores nem sempre serem ativados na ocorrência dessa atividade.

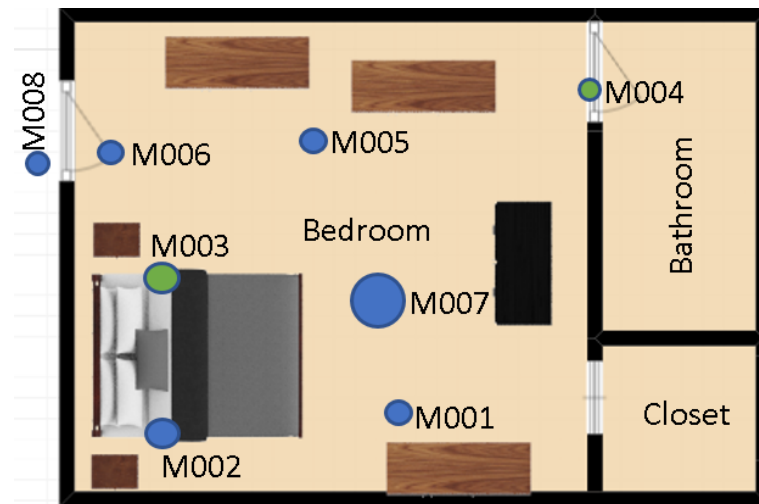


Figura 34: Destacando os sensores da atividade Cama para Banheiro. Fonte: Dados do Autor.

Usando a regra apresentada na Figura 35 obteve-se uma de acurácia de 98,08% no reconhecimento da atividade Cama para Banheiro.

Name
BedToToilet
Comment
Status
Ok
<pre>Windows(?w) ^ hasSensorName(?w, ?wsn) ^ swrlb:equal(?wsn, "M003") ^ hasStatusSensor(?w, ?wss) ^ swrlb:equal(?wss, "dt") ^ hasSensorName(?w, ?wsn1) ^ swrlb:equal(?wsn1, "M004") ^ hasStatusSensor(?w, ?wss1) ^ swrlb:equal(?wss1, "dt") ^ hasTimeEventStart(?h, ?tb) ^ hasTimeEventStart(?h, ?tb1) ^ swrlb:greaterThan(?tb1, ?tb) ^ Activity(?a) ^ hasNameActivity(?a, ?an) ^ swrlb:equal(?an, "bedtotoilet") ^ hasTimeEventEnd(?w, ?be) ^ hasTimeEventEnd(?w, ?be1) ^ swrlb:greaterThan(?be1, ?be) ^ Sensor(?s) ^ hasSensorName(?s, ?sn) ^ swrlb:equal(?sn, ?wsn) ^ hasSensorName(?s, ?sn1) ^ swrlb:equal(?sn, ?wsn1) ^ Person(?p) ^ hasActor(?p, ?at) ^ swrlb:equal(?at, "Maria") ^ Location(?l) ^ hasLocation(?l, ?lt) ^ swrlb:equal(?lt, "MasterBathroom") ^ Artifact(?at) ^ hasArtifact(?at, ?atn) ^ swrlb:equal(?atn, "DoorMasterToilet") ^ hasWindowBegin(?w, ?wb) ^ hasWindowsEnd(?w, ?we) ^ swrlb:makeOWLThing(?a, ?an) -> Activity(?an) ^ Activity_Windows(?an, ?w) ^ Activity_Sensor(?an, ?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifact(?an, ?atn) ^ hasWindowBegin(?an, ?w) ^ hasWindowsEnd(?an, ?w)</pre>

Figura 35: Regra Cama para Banheiro SWRL. Fonte: Dados do Autor.

Utilizou-se a seguinte consulta SPARQL exibida na Figura 36 para verificar informações relevantes da atividade reconhecida da *Cama para Banheiro*.


```

SELECT *
WHERE {
    ?act rdf:type ont:Activity .
    ?act ont:hasNameActivity ?name .
    ?act ont:hasNameActivity 'bedtotoilet' .
    ?act ont:Activity_Location ?location .
    ?act ont:Activity_Person ?person .
    ?act ont:Activity_Artifact ?artifact .
    ?act ont:Activity_Sensor ?sensor .
}

```

Figura 36: Consulta SPARQL da atividade Cama para Banheiro. Fonte: Dados do Autor.

As informações resultantes da busca em SPARQL da atividade reconhecida Cama para Banheiro pode ser observada na Figura 37.

act	name	location	person	artifact	sensor
ont:ABT00001	"bedtotoilet"	ont:Masterbathroom	ont:Maria	ont:MasterBathroom Door	ont:M004
ont:ABT00001	"bedtotoilet"	ont:Masterbathroom	ont:Maria	ont:MasterBathroom Door	ont:M003

Figura 37: Resultado da consulta SPARQL da atividade Cama para Banheiro. Fonte: Dados do Autor.

A consulta exibiu a localização (banheiro), o objeto de interação (a porta do banheiro) e os sensores do contexto de interesse. Novamente pode-se ver na coluna *act* a instância *cria ABT00001*.

Cenário 3: Atividade Entrando em Casa

Neste cenário Maria está chegando em casa. Ela abre a porta da garagem, passa pelo corredor em direção à sala de estar.

Nessa atividade considera-se como contexto de interesse os sensores D004 (correspondente à porta da casa), o M030 (sensor de presença em cima da porta), o M022 e o M021 (presentes no hall de acesso às outras dependências da casa). Na Figura 38, é mostrado a distribuição dos sensores envolvidos.

Aplicando a regra apresentada na Figura 41, obteve-se uma acurácia de 92,80% quando a pessoa entra em casa. Neste caso pode-se calcular a acurácia somente quando a pessoa entra em casa pela porta da garagem, que foi a única entrada que teve seus eventos descritos na documentação da casa inteligente Aruba com relação à atividade entrando em casa.

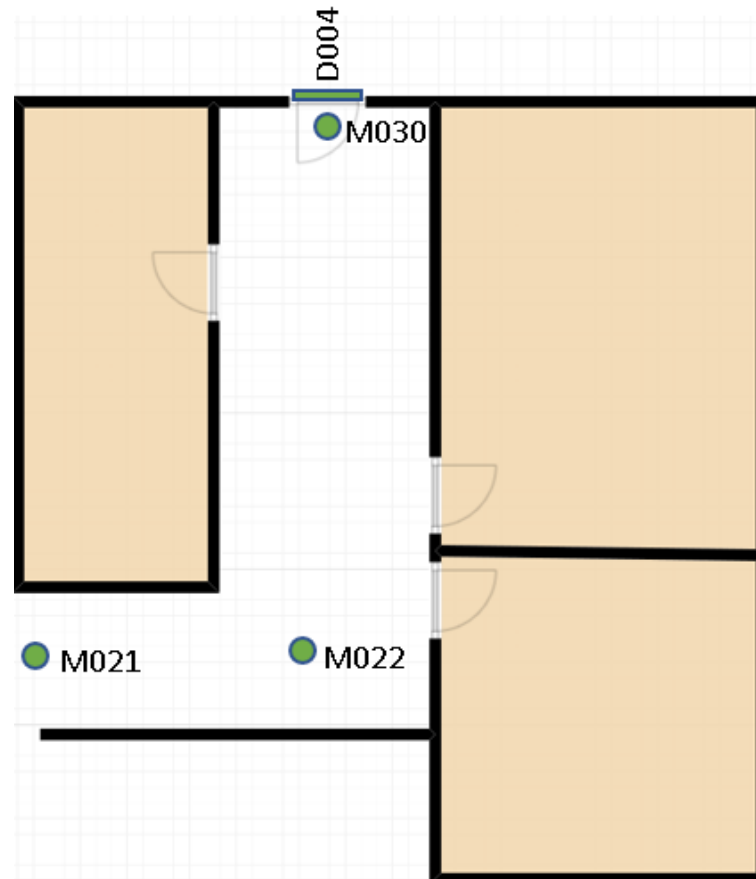


Figura 38: Sensores de interesse da tividade Entrando em Casa. Fonte: Dados do Autor.

Name
EnterHome
Comment
Status
Ok
<pre>Windows(?w) ^ hasSensorName(?w, ?wsn) ^ swrlb:equal(?wsn, "M022") ^ hasStatusSensor(?w, ?wss) ^ swrlb:equal(?wss, "dt") ^ hasEventBegin(?w, ?tb) ^ hasSensorName(?w, ?wsn1) ^ swrlb:equal(?wsn1, "M021") ^ hasStatusSensor(?w, ?wss1) ^ swrlb:equal(?wss1, "dt") ^ hasStatusSensor(?w, ?tb1) ^ hasSensorName(?w, ?wsn2) ^ swrlb:equal(?wsn2, "D004") ^ hasStatusSensor(?w, ?wss2) ^ swrlb:equal(?wss2, "dt") ^ hasEventBegin(?w, ?tb2) ^ Activity(?a) ^ hasNameActivity(?a, ?an) ^ swrlb:equal(?an, "enterhome") ^ swrlb:greaterThan(?tb1, ?tb) ^ swrlb:greaterThan(?tb1, ?tb2) ^ Sensor(?s) ^ hasSensorName(?s, ?sn) ^ swrlb:equal(?sn, ?wsn) ^ hasSensorName(?s, ?sn1) ^ swrlb:equal(?sn1, ?wsn1) ^ hasSensorName(?s, ?sn2) ^ swrlb:equal(?sn, ?wsn2) ^ Location(?l) ^ hasLocation(?l, ?lt) ^ swrlb:equal(?lt, "HallGarage") ^ Artifact(?ar) ^ hasArtifact(?ar, ?at) ^ swrlb:equal(?at, "DoorGarage") ^ Person(?p) ^ hasWindowBegin(?w, ?wb) ^ hasWindowsEnd(?w, ?we) ^ hasActor(?p, ?at) ^ swrlb:equal(?at, "Maria") ^ swrlb:makeOWLThing(?a, ?an) -> Activity(?an) ^ Activity_Windows(?an, ?w) ^ Activity_Sensor(?an, ?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifact(?an, ?at) ^ hasWindowBegin(?an, ?w) ^ hasWindowsEnd(?an, ?w)</pre>

Figura 39: Regra Entrando em Casa. Fonte: Dados do Autor.

Através da seguinte consulta SPARQL evidenciada na Figura 40 pode ser visto

informações relevantes na atividade *Entrando em Casa*.

```
SELECT *
WHERE {
    ?act rdf:type ont:Activity .
    ?act ont:hasNameActivity ?name .
    ?act ont:hasNameActivity 'enter' .
    ?act ont:Activity_Location ?location .
    ?act ont:Activity_Person ?person .
    ?act ont:Activity_Artifact ?artifact .
    ?act ont:Activity_Sensor ?sensor .
}
```

Figura 40: Consulta SPARQL da atividade Entrando em Casa. Fonte: Dados do Autor.

Informações resultantes da busca em SPARQL da atividade Entrando em Casa podem ser observadas na Figura 37.

act	name	location	artifact	sensor
ont:AEH00001	"enter"	ont:HallGarage	ont:GarageDoor	ont:D004
ont:AEH00001	"enter"	ont:HallGarage	ont:GarageDoor	ont:M021
ont:AEH00001	"enter"	ont:HallGarage	ont:GarageDoor	ont:M030
ont:AEH00001	"enter"	ont:HallGarage	ont:GarageDoor	ont:M022

Figura 41: Resultado da consulta SPARQL da atividade Entrando em Casa. Fonte: Dados do Autor.

Cenário 4: Atividade Saindo de Casa

Maria resolve sair de casa para fazer compras. Para isso, ela dirige-se ao corredor da casa que dá acesso à porta da garagem. Na atividade *Sair de Casa*, foi elaborada a regra empregando como contexto de interesse, os sensores M022 (hall de acesso á outras dependências da casa) como evento inicial da regra. Na sequencia de eventos, os sensores M021, M030 e D004 (no *hall* de acesso, em cima da porta e na porta da casa, respectivamente) são empregados para indicar a atividade. Foi empregada a mesma estratégia de sequencia temporal de eventos utilizada na regra anterior, alterando somente o evento de início e fim.

Obteve-se com a regra construída para reconhecer a atividade *Sair de Casa* exibida na Figura 42 com uma acurácia 93,96%.

Empregando a seguinte consulta SPARQL para atividade *Sair de Casa* exibida na Figura 43.

O resultado da consulta SPARQL da Figura 43 para atividade Sair de Casa é exibido na Figura 44.

Name
LeaveHome
Comment
Status
Ok
<pre>Windows(?w) ^ hasSensorName(?w, ?wsn) ^ swrlb:equal(?wsn, "M022") ^ hasStatusSensor(?w, ?wss) ^ swrlb:equal(?wss, "dt") ^ hasEventBegin(?w, ?tb) ^ hasSensorName(?w, ?wsn1) ^ swrlb:equal(?wsn1, "M021") ^ hasStatusSensor(?w, ?wss1) ^ swrlb:equal(?wss1, "dt") ^ hasEventBegin(?w, ?tb1) ^ hasSensorName(?w, ?wsn2) ^ swrlb:equal(?wsn2, "D004") ^ hasStatusSensor(?w, ?wss2) ^ swrlb:equal(?wss2, "dt") ^ hasEventBegin(?w, ?tb2) ^ Activity(?a) ^ hasNameActivity(?a, ?an) ^ swrlb:equal(?an, "leavehome") ^ swrlb:greaterThan(?tb, ?tb1) ^ swrlb:greaterThan(?tb2, ?tb1) ^ Sensor(?s) ^ hasSensorName(?s, ?sn) ^ swrlb:equal(?sn, ?wsn) ^ hasSensorName(?s, ?sn1) ^ swrlb:equal(?sn1, ?wsn1) ^ hasSensorName(?s, ?sn2) ^ swrlb:equal(?sn, ?wsn2) ^ Location(?l) ^ hasLocation(?l, ?lt) ^ swrlb:equal(?lt, "HallGarage") ^ Artifact(?ar) ^ hasArtifact(?ar, ?at) ^ swrlb:equal(?at, "DoorGarage") ^ Person(?p) ^ hasWindowBegin(?w, ?wb) ^ hasWindowsEnd(?w, ?we) ^ hasActor(?p, ?at) ^ swrlb:equal(?at, "Maria") ^ swrlb:makeOWLThing(?a, ?an) -> Activity(?an) ^ Activity_Windows(?an, ?w) ^ Activity_Sensor(?an, ?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifact(?an, ?at) ^ hasWindowBegin(?an, ?w) ^ hasWindowsEnd(?an, ?w)</pre>

Figura 42: Regra Saindo de Casa SWRL. Fonte: Dados do Autor.

```
SELECT *
WHERE {
    ?act rdf:type ont:Activity .
    ?act ont:hasNameActivity ?name .
    ?act ont:hasNameActivity 'leave' .
    ?act ont:Activity_Location ?location .
    ?act ont:Activity_Artifact ?artifact .
    ?act ont:Activity_Sensor ?sensor .
}
```

Figura 43: Consulta SPARQL da atividade Sair de Casa. Fonte: Dados do Autor

act	name	location	artifact	sensor
ont:ALH00001	"leave"	ont:HallGarage	ont:GarageDoor	ont:D004
ont:ALH00001	"leave"	ont:HallGarage	ont:GarageDoor	ont:M021
ont:ALH00001	"leave"	ont:HallGarage	ont:GarageDoor	ont:M030
ont:ALH00001	"leave"	ont:HallGarage	ont:GarageDoor	ont:M022

Figura 44: Resultado da consulta SPARQL da atividade *Sair de Casa*. Fonte: Dados do Autor.

Cenário 5: Atividade Trabalhando no Escritório

Maria decide trabalhar no computador, localizado no escritório.

No atual cenário, utilizou-se a seguinte sequência de eventos de sensores M028

(porta do escritório) representa o evento inicial, o qual irá indicar se Maria está entrando na localização de interesse da casa que é escritório. O sensor M026, representa o objeto de interesse (mesa do computador), sendo necessária, para detecção dessa atividade, uma interação superior a 600000 milissegundos.

A Figura 45 exibe a distribuição dos sensores pelo escritório.

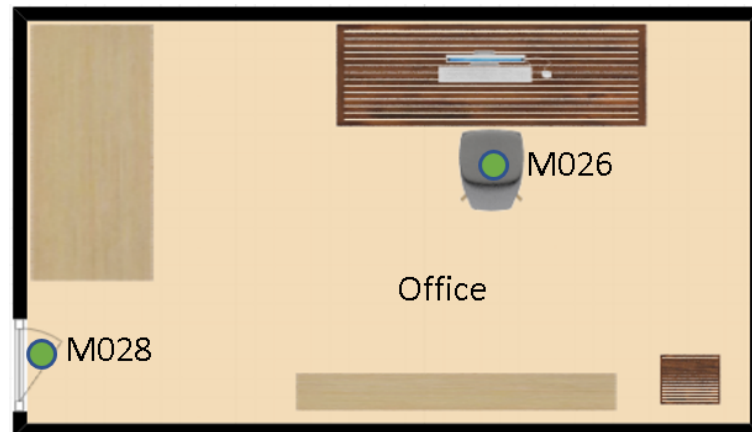


Figura 45: Sensores do Escritório. Fonte: Dados do Autor.

Esta regra é exibida na Figura 46. Aferiu-se a acurácia de 87,71% no reconhecimento da atividade Trabalhando no Escritório.

Name
Working
Comment
Status
Ok
<pre>Windows(?w) ^ hasSensorName(?w, ?wsn) ^ swrlb:equal(?wsn, "M020") ^ hasStatusSensor(?w, ?wss) ^ swrlb:equal(?wss, "dt") ^ hasEventBegin(?w, ?bt) ^ hasSensorName(?w, ?wsn1) ^ swrlb:equal(?wsn1, "M009") ^ hasStatusSensor(?w, ?wss1) ^ hasEventBegin(?w, ?bt1) ^ swrlb:greaterThan(?bt1, ?bt) ^ Activity(?a) ^ hasNameActivity(?a, ?an) ^ swrlb:equal(?an, "working") Sensor(?s) ^ hasSensorName(?s, ?sn) ^ swrlb:equal(?sn, ?wsn) ^ hasSensorName(?s, ?sn1) ^ swrlb:equal(?sn1, ?wsn1) ^ hasSensorName(?s, ?sn2) ^ swrlb:equal(?sn, ?wsn2) ^ Location(?l) ^ hasLocation(?l, ?lt) ^ swrlb:equal(?lt, "Livingroom") ^ Artifact(?ar) ^ hasArtifact(?ar, ?at) ^ swrlb:equal(?at, "Couch") ^ Person(?p) ^ hasWindowBegin(?w, ?wb) ^ hasWindowsEnd(?w, ?we) ^ hasActor(?p, ?at) ^ swrlb:equal(?at, "Maria") ^ swrlb:makeOWLThing(?a, ?an) -> Activity(?an) ^ Activity_Windows(?an, ?w) ^ Activity_Sensor(?an, ?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifact(?an, ?at) ^ hasWindowBegin(?an, ?w) ^ hasWindowsEnd(?an, ?w)</pre>

Figura 46: Regra Trabalhando no Escritório em SWRL. Fonte: Dados do Autor.

Empregando a consulta SPARQL evidenciada na Figura 47 pode ser visto os contextos inferidos.

```

SELECT *
WHERE {
    ?act rdf:type ont:Activity .
    ?act ont:hasNameActivity ?name .
    ?act ont:hasNameActivity 'working' .
    ?act ont:Activity_Location ?location .
    ?act ont:Activity_Person ?person .
    ?act ont:Activity_Artifact ?artifact .
    ?act ont:Activity_Sensor ?sensor .
}

```

Figura 47: Consulta SPARQL atividade Trabalhando no Escritório. Fonte: Dados do Autor.

A consulta exibe na figura 48 a atividade após sua detecção e instanciação na ontologia que possibilita a visualização de informações relevantes como a localização que é o escritório e os objetos de interação que são a porta do escritório e a mesa do computador e os sensores do contexto de interesse. Essa consulta mostra 4 atividades detectadas em um dia.

act	name	location	person	artifact	sensor
ont:AWK00001	"working"	ont:Office	ont:Maria	ont:OfficeDoor	ont:M028
ont:AWK00001	"working"	ont:Office	ont:Maria	ont:OfficeDoor	ont:M026
ont:AWK00001	"working"	ont:Office	ont:Maria	ont:TablePC	ont:M028
ont:AWK00001	"working"	ont:Office	ont:Maria	ont:TablePC	ont:M026
ont:AWK00004	"working"	ont:Office	ont:Maria	ont:OfficeDoor	ont:M028
ont:AWK00004	"working"	ont:Office	ont:Maria	ont:OfficeDoor	ont:M026
ont:AWK00004	"working"	ont:Office	ont:Maria	ont:TablePC	ont:M028
ont:AWK00004	"working"	ont:Office	ont:Maria	ont:TablePC	ont:M026
ont:AWK00002	"working"	ont:Office	ont:Maria	ont:OfficeDoor	ont:M028
ont:AWK00002	"working"	ont:Office	ont:Maria	ont:OfficeDoor	ont:M026
ont:AWK00002	"working"	ont:Office	ont:Maria	ont:TablePC	ont:M028
ont:AWK00002	"working"	ont:Office	ont:Maria	ont:TablePC	ont:M026
ont:AWK00003	"working"	ont:Office	ont:Maria	ont:OfficeDoor	ont:M028
ont:AWK00003	"working"	ont:Office	ont:Maria	ont:OfficeDoor	ont:M026
ont:AWK00003	"working"	ont:Office	ont:Maria	ont:TablePC	ont:M028
ont:AWK00003	"working"	ont:Office	ont:Maria	ont:TablePC	ont:M026

Figura 48: Resultado da consulta SPARQL atividade Trabalhando no Escritório. Fonte: Dados do Autor.

5.2 Avaliação de Acurácia no EXEHDA-AR

Aplicou-se como método para avaliação da acurácia, o desenvolvido por (KASTEREN; ALEMDAR; ERSOY, 2011), que consiste em criar um matriz de confusão na qual as linhas representam atividades e as colunas, a frequência dos verdadeiros positivos das atividades reconhecidas. A partir da matriz de confusão é possível mensurar a acurácia do reconhecimento das atividades, conforme ilustrado na Figura 49, tem-se um exemplo da construção da matriz.

		Class			
		1	2	3	
Class	1	TP_1	ϵ_{12}	ϵ_{13}	NG_1
	2	ϵ_{21}	TP_2	ϵ_{23}	NG_2
	3	ϵ_{31}	ϵ_{32}	TP_3	NG_3
		NI_1	NI_2	NI_3	$Total$

Figura 49: Matriz de confusão. Fonte: (KASTEREN; ALEMDAR; ERSOY, 2011).

A Figura 50 apresenta uma visão geral do método (KASTEREN; ALEMDAR; ERSOY, 2011):

$$Accuracy = \frac{\sum_{i=1}^Q TP_i}{Total}$$

Figura 50: Visão geral do método. Fonte: (KASTEREN; ALEMDAR; ERSOY, 2011).

De acordo com a Figura 50, **TP** representa o número de verdadeiros positivos das atividades reconhecidas, **Q** o número de atividades a serem reconhecidas (Dormir, Entrar em Casa, entre outras) e **Total** representa o número total de janelas de tempo deslizantes no conjunto de dados.

Esse método para a avaliação da acurácia, vem sendo empregado por diversos autores como, (COOK, 2012), (KABIR et al., 2016), (AL MACHOT; MAYR, 2016), entre outros.

Foi construída uma matriz de confusão para avaliar a acurácia do EXEHDA-AR. Apresenta uma estrutura de 5x5 colunas, onde cada célula com fundo amarelo representa o número de atividades reconhecidas na Figura 51.

A matriz está organizada da seguinte maneira: as atividades estão representadas por números de 1 a 5. A última coluna Ac/Atv representa acurácia por atividade e a última linha, a acurácia média do EXEHDA-AR. A representação das atividades estão descritas a seguir:

- atividade **Dormindo** é representada pelo número 1 e tem 401 verdadeiros positivos;
- atividade **Cama para Banheiro** é representada pelo número 2 e tem 157 verdadeiros positivos;
- atividade **Entrando em Casa** é representada pelo número 4 e tem 431 verdadeiros positivos;
- atividade **Saindo de Casa** é representada pelo número 5 e tem 431 verdadeiros positivos;
- atividade **Trabalhando no Escritório** é representada pelo número 6 e tem 171 verdadeiros positivos.

A Figura 51 representa a matriz de confusão do EXEHDA-AR, que atingiu uma acurácia média de 94,36%. Dentre as atividades, **Dormindo** teve o melhor resultado, enquanto que a atividade **Trabalhando no Escritório** apresentou um índice menor de acurácia, comparado aos demais.

Atividade	1	2	3	4	5	Ac/Atv
1	398	-	-	-	-	99,25%
2	-	154	-	-	-	98,08%
3	-	-	400	-	-	92,80%
4	-	-	-	405	-	93,96%
5	-	-	-	-	150	87,71%
Acurácia Média			94,36%			

Figura 51: Matriz de confusão do EXEHDA-AR. Fonte: Dados do Autor.

Acurácia obtida pelo EXEHDA-AR mostrou-se promissora frente ao estado da arte, aonde encontrou-se nos trabalhos como (COOK, 2012), (RIBONI et al., 2011), (VAN KASTEREN et al., 2008) e (TAPIA; INTILLE; LARSON, 2004) nos quais a acurácia média varia de 25% á 97% dependendo da situação avaliada.

5.3 Considerações sobre o Capítulo

Este capítulo apresentou o estudo de caso desenvolvido para testar o EXEHDA-AR. Para a execução dos testes, é importante destacar os esforços de pesquisas necessários para identificar *datasets* produzidos por casa inteligentes.

Como resultado da revisão de literatura foram identificados cinco(5) como os mais oportunos considerando a perspectiva do EXEHDA-AR, consideram a premissa do

EXEHDA-AR com pessoas que moram sozinhas foi selecionando o *dataset* pertinente a casa inteligente Aruba.

No presente estudo de caso, foram descritas as regras desenvolvidas para reconhecer atividades, consulta semântica das atividades inferidas bem como a matriz de confusão resultante.

6 CONSIDERAÇÕES FINAIS

Neste capítulo encontram-se resumidas as principais conclusões decorrentes da pesquisa relacionada ao desenvolvimento do EXEHDA-AR, as publicações realizadas, bem como os trabalhos futuros identificados.

6.1 Principais conclusões

O estudo de **Reconhecimento de Atividades** vem ganhando destaque nos últimos anos devido ao aumento da população de pessoas idosas, onde estas em sua maioria moram sozinhas e com passar do tempo necessitam de acompanhamento nas suas atividades diárias.

Como principais contribuições deste trabalho, destaca-se a elaboração dos componentes que foram integrados ao *middleware* EXEHDA os quais foram responsáveis pelo reconhecimento atividades utilizando processamento semântico.

A proposta do EXEHDA-AR, empregando sensores associados ao ambiente, e não diretamente no usuário, mostrou-se oportuna para a detecção de atividades, não necessitando que usuário se envolva de forma intencional com o procedimento de monitoramento das suas atividades, caracterizando assim uma abordagem não intrusiva.

A revisão da literatura realizada sobre as principais frentes de pesquisas e desafios envolvidos no reconhecimento de atividades, apontou para a importância da realização do reconhecimento de atividades diárias de forma o menos intrusiva possível.

O esforço de empregar no EXEHDA-AR somente tecnologias livres em sua concepção, permite uma abordagem promissora para o emprego do *middleware* em diferentes cenários de uso.

Com a avaliação das funcionalidades do EXEHDA-AR, os resultados do realizado estudo de caso realizado no domínio de uma Casa Inteligente utilizando dados reais, mostraram-se promissores com uma acurácia média de 94,36% de reconhecimento de atividade, o qual estimula a continuidade das pesquisas na área para trabalhos.

6.2 Publicações Realizadas

No decorrer desta dissertação de mestrado alguns artigos foram publicados, contemplando resultados parciais do trabalho desenvolvido, bem como a projeção destes nos esforços de pesquisa em andamento no LUPS.

- ABREU, E. S.; GUEDES, A. ; FLEISCHMANN, A. M. P. ; YAMIN, A. C. . EXEHDA-AR: Explorando Ciência de Situação para o Reconhecimento de Atividades na UbiComp. In: XVIII Encontro de Pós-Graduação UFPEL, 2016, Pelotas. XVIII Encontro de Pós-Graduação UFPEL.
- XAVIER, L. M. S. ; KAISER FILHO, H. ; ABREU, E. S. ; DAVET, P. T. ; PARREIRA, W. D. ; YAMIN, A. C. . Instrumentação na IoT: uma Abordagem Autônoma e Distribuída.. In: XXXIV Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT 2016), 2016, Santarém. Instrumentação na IoT: uma Abordagem Autônoma e Distribuída.. Rio de Janeiro: Sociedade Brasileira de Telecomunicações,, 2016.
- LOPES, J. L. B. ; GEYER, C. F. R. ; ABREU, E. S. ; REISER, R. H. S. ; SCHEUNEMANN, D. A. ; YAMIN, A. C. . Middleware Architecture For Supporting A Hybrid Processing Of Context Data Targeted To Detection Of Situations In UbiComp. In: Uncertainty Modelling in Knowledge Engineering and Decision Making: Proceedings of the 12th International FLINS Conference (FLINS 2016), 2016, Roubaix. MIDDLEWARE ARCHITECTURE FOR SUPPORTING A HYBRID PROCESSING OF CONTEXT DATA TARGETED TO DETECTION OF SITUATIONS IN UBICOMP, 2016. v. 10. p. 252-256.

6.3 Trabalhos Futuros

Os resultados obtidos com a realização desta dissertação de mestrado se mostraram promissores, estimulando a continuidade dos esforços de pesquisa relacionados a reconhecimento de atividades empregando o *middleware* EXEHDA. Dentre os aspectos levantados para continuidade do trabalho de pesquisa nesta frente junto ao grupo destacam-se:

- propor um novo modelo ontológico ainda mais genérico para o domínio de Casas Inteligentes;
- desenvolver um modelo híbrido para Reconhecimento de Atividades empregando técnicas baseadas em aprendizagem de máquina e em processamento semântico;

- viabilizar o Reconhecimento de Atividades oportunístico direcionado a atender a dinamicidade usual dos ambientes de IoT, quanto a entrada e saída de recursos de sensoriamento;
- explorar o emprego dos mecanismos concebidos para o EXEHDA-AR em outros cenários de uso.

Além desses trabalhos futuros, está prevista a escrita de artigos direcionados a periódicos científicos e congressos na área, divulgando assim o trabalho desenvolvido.

REFERÊNCIAS

AGOSTINI, A.; BETTINI, C.; RIBONI, D. Hybrid reasoning in the CARE middleware for context awareness. **International journal of Web engineering and technology**, [S.l.], v.5, n.1, p.3–23, 2009.

AL MACHOT, F.; MAYR, H. C. Improving Human Activity Recognition by Smart Windowing and Spatio-Temporal Feature Analysis. In: ACM INTERNATIONAL CONFERENCE ON PERVASIVE TECHNOLOGIES RELATED TO ASSISTIVE ENVIRONMENTS, 9., 2016. **Proceedings...** [S.l.: s.n.], 2016. p.56.

AL-SHAQI, R.; MOURSHED, M.; REZGUI, Y. Progress in ambient assisted systems for independent living by the elderly. **SpringerPlus**, [S.l.], v.5, n.1, p.624, 2016.

AMIRIBESHELI, M.; BENMANSOUR, A.; BOUCHACHIA, A. A review of smart homes in healthcare. **Journal of Ambient Intelligence and Humanized Computing**, [S.l.], p.1–23, 2015.

ATALLAH, L.; YANG, G.-Z. The use of pervasive sensing for behaviour profiling? a survey. **Pervasive and Mobile Computing**, [S.l.], v.5, n.5, p.447–464, 2009.

BAADER, F. **The description logic handbook**: Theory, implementation and applications. [S.l.]: Cambridge university press, 2003.

BECKER, C.; DÜRR, F. On location models for ubiquitous computing. **Personal and Ubiquitous Computing**, [S.l.], v.9, n.1, p.20–31, 2005.

BETTINI, C.; BRDICZKA, O.; HENRICKSEN, K.; INDULSKA, J.; NICKLAS, D.; RANGANATHAN, A.; RIBONI, D. A survey of context modelling and reasoning techniques. **Pervasive and Mobile Computing**, [S.l.], v.6, n.2, p.161–180, 2010.

CASSENS, J.; KOFOD-PETERSEN, A. Using Activity Theory to Model Context Awareness: A Qualitative Case Study. In: FLAIRS CONFERENCE, 2006. **Anais...** [S.l.: s.n.], 2006. p.619–624.

CHEN, L.; NUGENT, C. Ontology-based activity recognition in intelligent pervasive environments. **International Journal of Web Information Systems**, [S.l.], v.5, n.4, p.410–430, 2009.

CHEN, L.; NUGENT, C.; MULVENNA, M.; FINLAY, D.; HONG, X. Semantic smart homes: towards knowledge rich assisted living environments. In: **Intelligent Patient Management**. [S.l.]: Springer, 2009. p.279–296.

COOK, D. J. Learning setting-generalized activity models for smart spaces. **IEEE intelligent systems**, [S.l.], v.27, n.1, p.32–38, 2012.

CULMONE, R.; FALCIONI, M.; GIULIODORI, P.; MERELLI, E.; ORRU, A.; QUADRINI, M.; CIAMPOLINI, P.; GROSSI, F.; MATRELLA, G. AAL domain ontology for event-based human activity recognition. In: MECHATRONIC AND EMBEDDED SYSTEMS AND APPLICATIONS (MESA), 2014 IEEE/ASME 10TH INTERNATIONAL CONFERENCE ON, 2014. **Anais...** [S.l.: s.n.], 2014. p.1–6.

DAVET, P.; KAISER FILHO, H.; JOÃO, L.; XAVIER, L.; CARVALHO, T.; FLEISCHMANN, A.; YAMIN, A.; LOPES, J.; SOUZA, R. Consciência de Contexto na IoT: uma Arquitetura Distribuída e Escalável. In: V SBESC, 2015. **Proceedings...** [S.l.: s.n.], 2015.

DAWADI, P. N.; COOK, D. J.; SCHMITTER-EDGECOMBE, M. Modeling patterns of activities using activity curves. **Pervasive and Mobile Computing**, [S.l.], 2015.

DEY, A. K. Understanding and using context. **Personal and ubiquitous computing**, [S.l.], v.5, n.1, p.4–7, 2001.

DIXON, C.; MAHAJAN, R.; AGARWAL, S.; BRUSH, A.; LEE, B.; SAROIU, S.; BAHL, P. An operating system for the home. In: USENIX CONFERENCE ON NETWORKED SYSTEMS DESIGN AND IMPLEMENTATION, 9., 2012. **Proceedings...** [S.l.: s.n.], 2012. p.25–25.

GALLISSOT, M.; CAELEN, J.; BONNEFOND, N.; MEILLON, B.; PONS, S. Using the multicom domus dataset. **LIG, Grenoble, France, Research Report RR-LIG-020**, [S.l.], 2011.

GU, T.; WU, Z.; TAO, X.; PUNG, H. K.; LU, J. epsicar: An emerging patterns based approach to sequential, interleaved and concurrent activity recognition. In: PERVASIVE COMPUTING AND COMMUNICATIONS, 2009. PERCOM 2009. IEEE INTERNATIONAL CONFERENCE ON, 2009. **Anais...** [S.l.: s.n.], 2009. p.1–9.

HALPIN, T. **Information Modeling and Relational Databases**: From Conceptual Analysis to Logical Design, ISBN-13: 978-1-55860-672-2, ISBN-10: 1-55860-672-6. [S.l.]: San Francisco, California, Morgan Kaufman Publishers, 2001.

HELAL, S.; MANN, W.; EL-ZABADANI, H.; KING, J.; KADDOURA, Y.; JANSEN, E. The gator tech smart house: A programmable pervasive space. **Computer**, [S.l.], v.38, n.3, p.50–60, 2005.

HENRICKSEN, K.; INDULSKA, J. Modelling and using imperfect context information. In: PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS, 2004. PROCEEDINGS OF THE SECOND IEEE ANNUAL CONFERENCE ON, 2004. **Anais...** [S.l.: s.n.], 2004. p.33–37.

HENRICKSEN, K.; INDULSKA, J. Developing context-aware pervasive computing applications: Models and approach. **Pervasive and mobile computing**, [S.l.], v.2, n.1, p.37–64, 2006.

HENRICKSEN, K.; INDULSKA, J.; RAKOTONIRAINY, A. Modeling context information in pervasive computing systems. In: **Pervasive Computing**. [S.l.]: Springer, 2002. p.167–180.

HERVÁS, R.; BRAVO, J.; FONTECHA, J. A Context Model based on Ontological Languages: a Proposal for Information Visualization. **J. UCS**, [S.l.], v.16, n.12, p.1539–1555, 2010.

HORROCKS, I.; PATEL-SCHNEIDER, P. F.; VAN HARMELEN, F. From SHIQ and RDF to OWL: The making of a web ontology language. **Web semantics: science, services and agents on the World Wide Web**, [S.l.], v.1, n.1, p.7–26, 2003.

INTILLE, S. S.; LARSON, K.; BEAUDIN, J.; NAWYN, J.; TAPIA, E. M.; KAUSHIK, P. A living laboratory for the design and evaluation of ubiquitous computing technologies. In: CHI'05 EXTENDED ABSTRACTS ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2005. **Anais...** [S.l.: s.n.], 2005. p.1941–1944.

KABIR, M. H.; HOQUE, M. R.; THAPA, K.; YANG, S.-H. Two-layer hidden markov model for human activity recognition in home environments. **International Journal of Distributed Sensor Networks**, [S.l.], 2016.

KASTEREN, T. L. van; ALEMDAR, H.; ERSOY, C. Effective performance metrics for evaluating activity recognition methods. **ARCS 2011**, [S.l.], 2011.

KIENTZ, J. A.; PATEL, S. N.; JONES, B.; PRICE, E.; MYNATT, E. D.; ABOWD, G. D. The georgia tech aware home. In: CHI'08 EXTENDED ABSTRACTS ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2008. **Anais...** [S.l.: s.n.], 2008. p.3675–3680.

KIM, E.; HELAL, S.; COOK, D. Human activity recognition and pattern discovery. **Pervasive Computing, IEEE**, [S.l.], v.9, n.1, p.48–53, 2010.

KLYNE, G. Composite capability/preference profiles (cc/pp): Structure and vocabularies. <http://www.w3.org/TR/2001/WD-CCPP-struct-vocab-20010315>, [S.l.], 2001.

KRISHNAN, N. C.; COOK, D. J. Activity recognition on streaming sensor data. **Pervasive and mobile computing**, [S.l.], v.10, p.138–154, 2014.

KRUMM, J. **Ubiquitous computing fundamentals**. [S.l.]: CRC Press, 2016.

LAI, C.-F.; LAI, Y.-X.; YANG, L. T.; CHAO, H.-C. Integration of IoT Energy Management System with Appliance and Activity Recognition. In: GREEN COMPUTING AND COMMUNICATIONS (GREENCOM), 2012 IEEE INTERNATIONAL CONFERENCE ON, 2012. **Anais...** [S.l.: s.n.], 2012. p.66–71.

LOPES, J. L.; SOUZA, R. S. de; GEYER, C. F. R.; COSTA, C. A. da; BARBOSA, J. L.; PERNAS, A. M.; YAMIN, A. C. A Middleware Architecture for Dynamic Adaptation in Ubiquitous Computing. **J. UCS**, [S.l.], v.20, n.9, p.1327–1351, 2014.

MARIN, M. J. S.; PANES, V. C. B. ENVELHECIMENTO DA POPULAÇÃO E AS POLÍTICAS PÚBLICAS DE SAÚDE. **Revista do Instituto de Políticas Públicas de Marília**, [S.l.], v.1, n.1, 2015.

MCMICHAEL, H. An activity based perspective for information systems research. In: AUSTRALIAN CONFERENCE ON INFORMATION SYSTEMS, 10., 1999. **Anais...** [S.l.: s.n.], 1999.

MUNGUIA TAPIA, E. **Activity recognition in the home setting using simple and ubiquitous sensors**. 2003. Tese (Doutorado em Ciência da Computação) — Massachusetts Institute of Technology.

NI, Q.; GARCÍA HERNANDO, A. B.; CRUZ, I. Pau de la. A Context-Aware System Infrastructure for Monitoring Activities of Daily Living in Smart Home. **Journal of Sensors**, [S.l.], v.2016, 2016.

NICKLAS, D.; GROSSMANN, M.; MÍNGUEZ, J.; WIELAND, M. Adding high-level reasoning to efficient low-level context management: A hybrid approach. In: PERVASIVE COMPUTING AND COMMUNICATIONS, 2008. PERCOM 2008. SIXTH ANNUAL IEEE INTERNATIONAL CONFERENCE ON, 2008. **Anais...** [S.l.: s.n.], 2008. p.447–452.

OKEYO, G.; CHEN, L.; WANG, H.; STERRITT, R. Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. **Pervasive and Mobile Computing**, [S.l.], v.10, p.155–172, 2014.

OSMANI, V.; BALASUBRAMANIAM, S.; BOTVICH, D. Human activity recognition in pervasive health-care: Supporting efficient remote collaboration. **Journal of network and computer applications**, [S.l.], v.31, n.4, p.628–655, 2008.

PEDDEMORS, A.; EERTINK, H.; NIEMEGEREERS, I. Predicting mobility events on personal devices. **Pervasive and Mobile Computing**, [S.l.], v.6, n.4, p.401–423, 2010.

PERERA, C.; MEMBER, C. H. L.; JAYAWARDENA, S.; CHEN, M. Context-aware Computing in the Internet of Things: A Survey on Internet of Things From Industrial Market Perspective. **arXiv preprint arXiv:1502.00164**, [S.l.], 2015.

PERERA, C.; ZASLAVSKY, A.; CHRISTEN, P.; GEORGAKOPOULOS, D. Context aware computing for the internet of things: A survey. **Communications Surveys & Tutorials, IEEE**, [S.l.], v.16, n.1, p.414–454, 2014.

RATHNAYAKA, A. D.; POTDAR, V. M.; KURUPPU, S. J. Evaluation of wireless home automation technologies. In: DIGITAL ECOSYSTEMS AND TECHNOLOGIES CONFERENCE (DEST), 2011 PROCEEDINGS OF THE 5TH IEEE INTERNATIONAL CONFERENCE ON, 2011. **Anais...** [S.l.: s.n.], 2011. p.76–81.

RIBONI, D.; BETTINI, C. COSAR: hybrid reasoning for context-aware activity recognition. **Personal and Ubiquitous Computing**, [S.l.], v.15, n.3, p.271–289, 2011.

RIBONI, D.; PARESCHI, L.; RADAELLI, L.; BETTINI, C. Is ontology-based activity recognition really effective? In: PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS (PERCOM WORKSHOPS), 2011 IEEE INTERNATIONAL CONFERENCE ON, 2011. **Anais...** [S.l.: s.n.], 2011. p.427–431.

RÖCKER, C.; ZIEFLE, M.; HOLZINGER, A. From computer innovation to human integration: current trends and challenges for pervasive HealthTechnologies. In: **Pervasive Health**. [S.l.]: Springer, 2014. p.1–17.

ROGGEN, D.; TROESTER, G.; LUKOWICZ, P.; FERSCHA, A.; MILLÁN, J. d. R.; CHAVARRIAGA, R. Opportunistic human activity and context recognition. **Computer**, [S.l.], v.46, n.2, p.36–45, 2013.

SALFNER, F.; LENK, M.; MALEK, M. A survey of online failure prediction methods. **ACM Computing Surveys (CSUR)**, [S.l.], v.42, n.3, p.10, 2010.

SILVA MACHADO, R. da; ALMEIDA, R. B.; YAMIN, A. C.; PERNAS, A. M. LogA-DM: An Approach of Dynamic Log Analysis. **IEEE Latin America Transactions**, [S.l.], v.13, n.9, p.3096–3102, 2015.

SINGLA, G.; COOK, D. J.; SCHMITTER-EDGEcombe, M. Recognizing independent and joint activities among multiple residents in smart environments. **Journal of ambient intelligence and humanized computing**, [S.l.], v.1, n.1, p.57–63, 2010.

TAPIA, E. M.; INTILLE, S. S.; LARSON, K. Activity recognition in the home using simple and ubiquitous sensors. In: INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING, 2004. **Anais...** [S.l.: s.n.], 2004. p.158–175.

UDEN, L. Activity theory for designing mobile learning. **International Journal of Mobile Learning and Organisation**, [S.l.], v.1, n.1, p.81–102, 2006.

VAN KASTEREN, T.; NOULAS, A.; ENGLEBIENNE, G.; KRÖSE, B. Accurate activity recognition in a home setting. In: UBIQUITOUS COMPUTING, 10., 2008. **Proceedings...** [S.l.: s.n.], 2008. p.1–9.

VENECIAN, L. R. et al. Um Mecanismo de Sensibilidade ao Contexto com Suporte Semântico para Computação Ubíqua. , [S.l.], 2010.

VIEIRA, V.; TEDESCO, P.; SALGADO, A. C. Modelos e Processos para o desenvolvimento de Sistemas Sensíveis ao Contexto. **André Ponce de Leon F. de Carvalho, Tomasz Kowaltowski.(Org.). Jornadas de Atualização em Informática**, [S.l.], p.381–431, 2009.

WANG, X. H.; ZHANG, D. Q.; GU, T.; PUNG, H. K. Ontology based context modeling and reasoning using OWL. In: PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS, 2004. PROCEEDINGS OF THE SECOND IEEE ANNUAL CONFERENCE ON, 2004. **Anais...** [S.l.: s.n.], 2004. p.18–22.

WEISER, M. The computer for the 21st century. **Scientific american**, [S.l.], v.265, n.3, p.94–104, 1991.

WEISSENBERG, N.; GARTMANN, R.; VOISARD, A. An ontology-based approach to personalized situation-aware mobile service supply. **Geoinformatica**, [S.l.], v.10, n.1, p.55–90, 2006.

WONGPATIKASEREE, K.; IKEDA, M.; BURANARACH, M.; SUPNITHI, T.; LIM, A. O.; TAN, Y. Activity recognition using context-aware infrastructure ontology in smart home domain. In: KNOWLEDGE, INFORMATION AND CREATIVITY SUPPORT SYSTEMS (KICSS), 2012 SEVENTH INTERNATIONAL CONFERENCE ON, 2012. **Anais...** [S.l.: s.n.], 2012. p.50–57.

XU, C.; CHEUNG, S.; MA, X.; CAO, C.; LU, J. Adam: Identifying defects in context-aware adaptation. **Journal of Systems and Software**, [S.l.], v.85, n.12, p.2812–2828, 2012.

YALA, N.; FERGANI, B.; FLEURY, A. Feature extraction and incremental learning to improve activity recognition on streaming data. In: EVOLVING AND ADAPTIVE INTELLIGENT SYSTEMS (EAIS), 2015 IEEE INTERNATIONAL CONFERENCE ON, 2015. **Anais...** [S.l.: s.n.], 2015. p.1–8.

YAU, S. S.; LIU, J. Hierarchical situation modeling and reasoning for pervasive computing. In: SOFTWARE TECHNOLOGIES FOR FUTURE EMBEDDED AND UBIQUITOUS SYSTEMS, 2006 AND THE 2006 SECOND INTERNATIONAL WORKSHOP ON COLLABORATIVE COMPUTING, INTEGRATION, AND ASSURANCE. SEUS 2006/WCCIA 2006. THE FOURTH IEEE WORKSHOP ON, 2006. **Anais...** [S.l.: s.n.], 2006. p.6–pp.

YE, J.; COYLE, L.; DOBSON, S.; NIXON, P. Using situation lattices to model and reason about context. **Proceedings of MRC 2007 (coexist with CONTEXT'07)**, [S.l.], p.1–12, 2007.

YE, J.; STEVENSON, G.; DOBSON, S. KCAR: A knowledge-driven approach for concurrent activity recognition. **Pervasive and Mobile Computing**, [S.l.], v.19, p.47–70, 2015.

YURUR, O.; LIU, C.; MORENO, W. A survey of context-aware middleware designs for human activity recognition. **IEEE Communications Magazine**, [S.l.], v.52, n.6, p.24–31, 2014.

ZIMMERMANN, A.; LORENZ, A.; OPPERMANN, R. An operational definition of context. In: **Modeling and using context**. [S.l.]: Springer, 2007. p.558–571.