Reconhecimento de Atividades em Ambientes de Vivência Assistida: uma Proposta Explorando um Modelo Ontológico

Roger da Silva Machado[1], **Felipe Luzzardi da Rosa**[1], **Eduardo Abreu**[1], **Ana Marilza Pernas**[1], **Adenauer Yamin**[1]

[1]Programa de Pós-Graduação em Computação (PPGC)
Universidade Federal de Pelotas (UFPel) – Pelotas – RS – Brazil

{rdsmachado, fldrosa, eabreu, marilza, adenauer}@inf.ufpel.edu.br

Abstract. *In recent years, there has been an increasing interest in the use of assisted living environments, and these environments should include computational services that can recognize the activities performed by residents and support them when necessary. Thinking about this, this work aims at the design of a model capable of providing the activities recognition performed in a smart environment, being designed based on an ontology. The proposed model, called EXEHDA-AR, was evaluated through a case study using a database of an intelligent house, with an average accuracy of 94.89% in the activities recognition. These results indicate that the proposed model is a viable alternative, thus stimulating the continuity of research efforts.*

Resumo. *Nos últimos anos, pode-se notar um aumento no interesse do uso de ambientes de vivência assistida, sendo que estes ambientes deverão contemplar serviços computacionais que possam reconhecer as atividades realizadas por moradores e auxiliá-los quando necessário. Pensando nisso, o objetivo deste trabalho é a concepção de um modelo capaz de prover o reconhecimento de atividades executadas em um ambiente inteligente, sendo este modelo concebido com base em uma ontologia. O modelo proposto, denominado EXEHDA-AR, foi avaliado por meio de um estudo de caso utilizando uma base de dados de uma casa inteligente, tendo sido obtida uma acurácia média de 94.89% no reconhecimento das atividades. Estes resultados apontam que o modelo proposto é uma alternativa viável, estimulando assim a continuidade dos esforços de pesquisa.*

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 a população do Brasil, que até então é considerada como 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 and Panes 2015].

Esse processo torna necessário novos mecanismos que possam reconhecer as atividades realizadas pelas pessoas em suas residências, visando auxiliá-las quando necessário. 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 et al. 2008], domótica [Singla et al. 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 [Al-Shaqi et al. 2016].

Nesta perspectiva, nota-se um aumento na utilização de casas inteligentes que proporcionem um ambiente assistido de vivência no contexto da saúde. Nestas residências, devem ser contemplados serviços computacionais que possam auxiliar as pessoas nas suas práticas diárias, da forma mais transparente possível [Röcker et al. 2014]. Busca-se assim integrar a tecnologia ao cotidiano do usuário, caracterizando uma infraestrutura computacional de natureza ubíqua que deve exigir o menor envolvimento possível das pessoas no seu gerenciamento. Além disso, torna-se importante que as atividades sejam reconhecidas a partir de dados de contexto coletados por sensores, os quais podem estar presentes em objetos carregados pelas pessoas ou incorporados ao ambiente em que as mesmas interagem [Perera et al. 2015].

Pensando nisso, o objetivo deste trabalho é a concepção de um modelo, chamado aqui de EXEHDA-AR (*Execution Environment for Highly Distributed Applications-Activity Recognition*), para o reconhecimento de atividades realizadas por moradores em um ambiente de vivência assistida. Este trabalho está integrado com o *middleware* EXEHDA [Lopes et al. 2014], contribuindo com o Subsistema de Reconhecimento de Contexto e Adaptação do *middleware* por meio de um sistema para reconhecimento de atividades (Activity Recognition), contemplando assim uma abordagem que permite o reconhecimento de atividades em casas inteligentes.

Para tanto, é concebido um módulo de Pré-processamento para se comunicar com a aquisição dos dados realizada pelo *middleware*, bem como um modelo ontológico capaz de reconhecer atividades realizadas por um morador em uma casa inteligente. Um dos diferenciais do modelo proposto, quando comparado com os trabalhos relacionados, é a exploração do potencial do *middleware* para reconhecimento não intrusivo de atividades em intervalos de tempo próximos ao momento em que as rotinas estão sendo observadas.

O restante deste artigo está organizado da seguinte forma. A Seção 2 descreve as principais características do *middleware* EXEHDA. Na Seção 3 é apresentado o modelo ontológico para reconhecimento de atividade proposto. Já na quarta Seção, é discutida a avaliação da proposta, explorando cenários de uso. A Seção 5 revisa alguns trabalhos relacionados. Finalmente, a Seção 6 apresenta as considerações finais e trabalhos futuros.

2. Middleware EXEHDA

O EXEHDA consiste de um *middleware* adaptativo ao contexto baseado em serviços, o qual visa criar e gerenciar ambientes de computação ubíqua, como os providos pela Internet das Coisas, bem como promover a execução de aplicações sobre ele [Lopes et al. 2014]. A arquitetura do *middleware* contém módulos para reconhecimento e adaptação ao contexto, compreendendo dois tipos principais de servidores: Servidor de Borda e Servidor de Contexto. O Servidor de Borda é responsável pela interação com o meio através de sensores e atuadores, enquanto que o Servidor de Contexto atua no armazenamento e processamento das informações contextuais.

Uma visão geral da arquitetura do Servidor de Contexto é apresentada na Figura 1, 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. Os módulos presentes na Arquitetura do servidor de Contexto são:

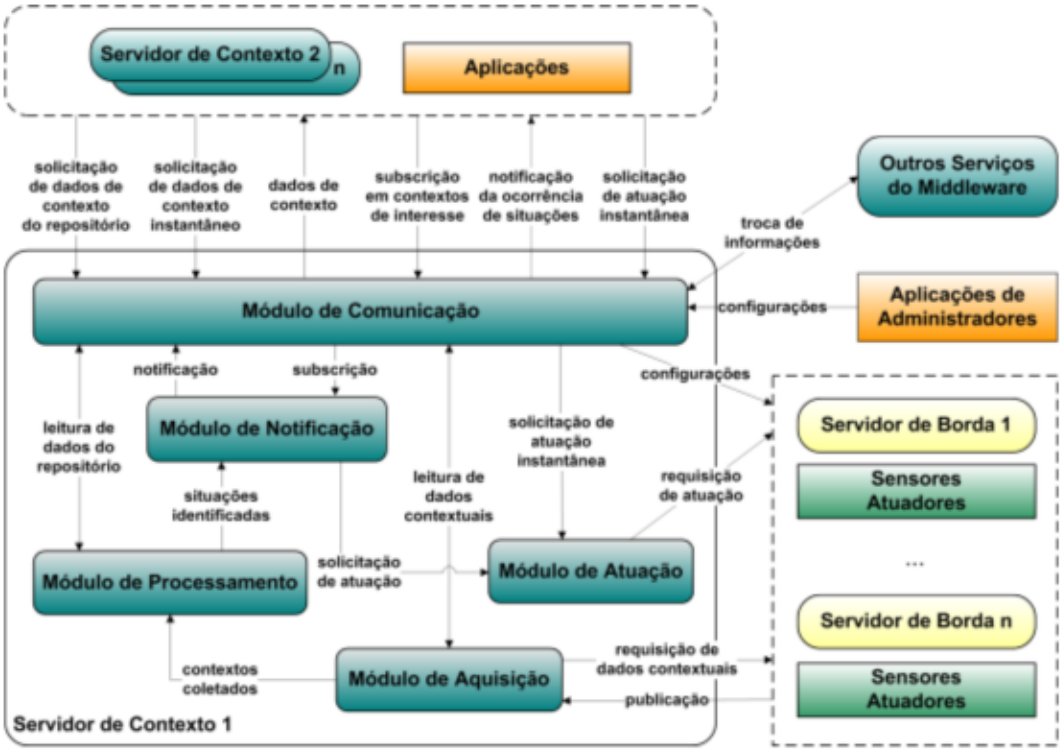

Figura 1. Visão geral da arquitetura do servidor de contexto.

- Aquisição - fornece suporte para a captura de informações contextuais, coletadas por servidores de borda, por meio de interfaces de software e/ou sensores de hardware;
- Atuação - é responsável pelo controle dos atuadores (ativação, desativação e configuração), após ser notificado por outros módulos do servidor de contexto;
- Processamento - processa informações contextuais baseadas em contextos de interesse das aplicações. Este módulo mantém um Repositório de Contexto que emprega um modelo relacional para a representação de contexto;
- Notificação - trata de notificar o resultado do processamento de contexto realizado pelo módulo de Processamento. O notificador recebe todas as decisões de atuação, resultantes da manipulação das regras de contexto;
- Comunicação - usado por servidores de contexto remotos e/ou aplicativos para solicitar dados contextuais e/ou acionar atuadores. Este módulo fornece a disseminação de informações de contexto para outros serviços do *middleware*, assim como pode enviar mensagens para aplicativos do usuário.

3. Modelo Ontológico Proposto

Com a inclusão do modelo proposto ao *middleware* EXEHDA, tornou-se necessária a concepção de um módulo de Pré-processamento. Este modulo é responsável por comunicar-se com o componente de aquisição do *middleware* e receber os dados coletados

do ambiente de vivência assistida monitorado, realizando a interpretação e a agregação das informações contextuais.

O módulo de Pré-processamento realiza a agregação das informações relacionadas aos sensores, onde são combinados os dados de horário de início e fim e a identificação da janela em que foi coletado o evento. Outra abstração provida por este componente é a normalização no formato da data coletado, cujas informações são convertidas para milissegundos, com intuito de serem utilizadas na construção das regras.

Além disso, este módulo emprega o conceito de janela de tempo finita para realizar a coleta dos dados, permitindo assim agrupar as informações coletadas, sendo possível aumentar ou diminuir a janela de tempo de acordo com perfil do morador do ambiente de vivência assistida. O tamanho da janela de tempo empregada varia de 60 segundos até 1 hora.

Após o janelamento de tempo, os eventos coletados são repassados para o modelo ontológico para realizar o reconhecimento de atividade. A ontologia concebida é apresentada na Figura 2, a qual utilizou como base a ontologia do projeto PalS-POT [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*, com seus atributos e relacionamentos. Com base nessas classes, foram criadas subclasses e a classe *Windows*, a qual possui a finalidade de receber as informações do módulo de Pré-processamento.

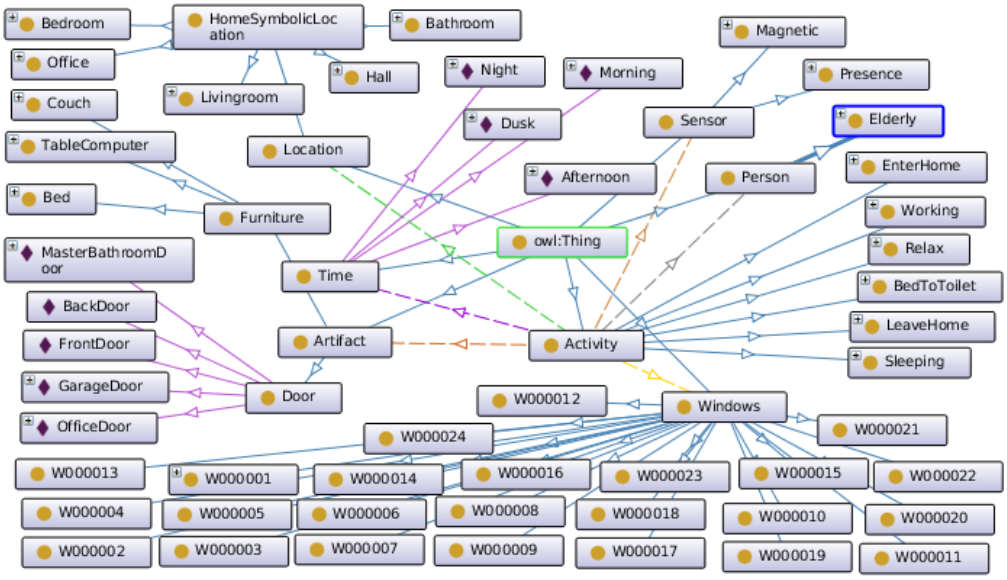

Figura 2. Visão geral da ontologia concebida.

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 3. No modelo proposto, 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.

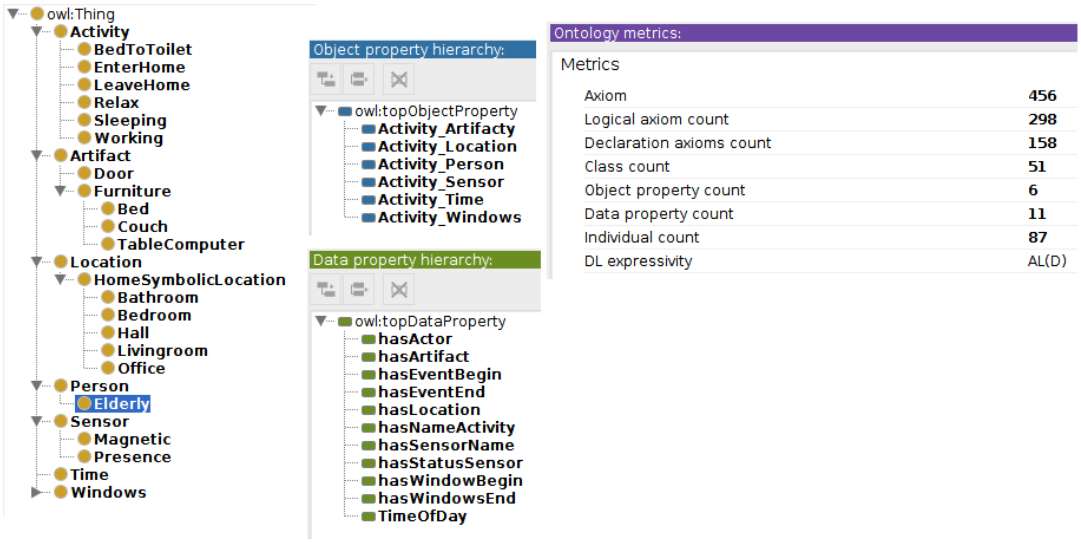

Figura 3. Classes e atributos da ontologia proposta.

A classe *Activity* tem subclasses com as atividades que deseja-se 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, são inseridos somente o tempo do primeiro e do último evento na janela corrente.

Para realizar o processamento dos dados e o reconhecimento de atividades, foram construídas regras usando a linguagem SWRL (*Semantic Web Rule Language*) utilizando os conceitos propostos por [Chen and Nugent 2009], por meio da relação entre a localização simbólica e o 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. 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 indivíduo.

Após o processamento da ontologia e o possível reconhecimento de atividades, os dados são enviados para o Repositório de Contexto do EXEHDA, no qual foi inserido o modelo de triplas. Com o armazenamento no modelo de triplas, é possível oferecer um modelo mais eficiente, por exemplo, que um modelo relacional [Can et al. 2017]. Desta forma, é possível realizar consultas SPARQL tanto nos dados atuais que estão sendo processados na ontologia como, também, em dados históricos que estão no repositório.

4. Avaliação do Modelo Ontológico

Para realizar a avaliação do modelo proposto, foi selecionada uma base de dados pertencente ao grupo de pesquisa CASAS da Universidade de Washington [Cook 2012]. Esta base de dados é proveniente de uma casa denominada Aruba, na qual uma senhora idosa reside sozinha. Na base de dados estão presentes nove atividades monitoradas, sendo que

no estudo de caso realizado optou-se por analisar cinco destas atividades: (i) dormir; (ii) deslocamento da cama para banheiro; (iii) entrar em casa; (iv) sair de casa; e (v) trabalhar no escritório.

Com o intuito de realizar o reconhecimento das atividades presentes na base de dados, foram concebidas regras utilizando a linguagem SWRL, sendo estas concebidas utilizando a relação entre localidade, objeto e tempo. Na Figura 4 é apresentada a distribuição dos sensores na casa Aruba, os quais terão seus valores coletados e analisados pelo modelo proposto. A moradora da casa não tem seu nome revelado por questões de privacidade, atribuiu-se, portanto, o pseudo nome de Maria a ser utilizado na descrição dos cenários.

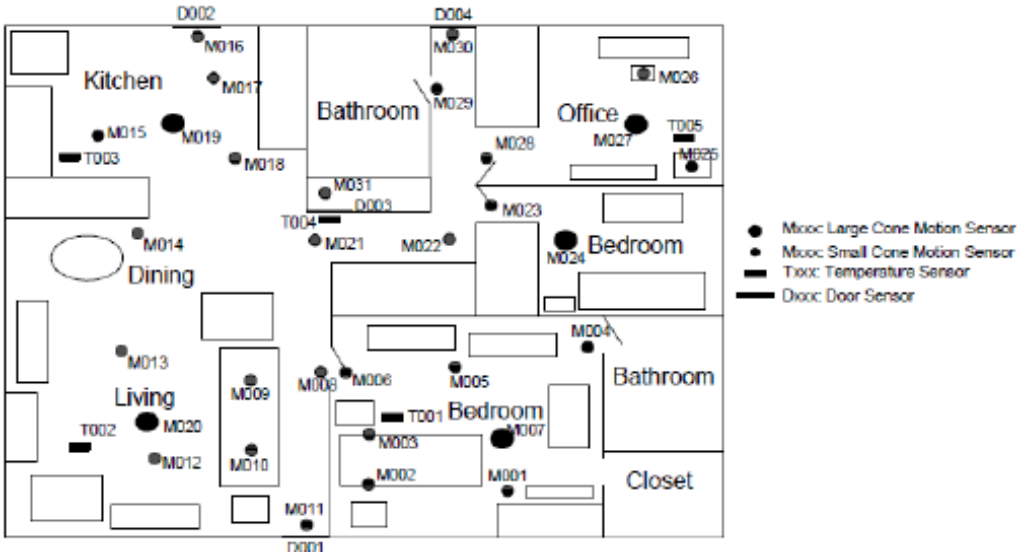

Figura 4. Distribuição dos sensores casa Aruba.
Fonte: [Yala et al. 2015].

4.1. Cenário 1: Atividade dormir

Nesta atividade a moradora Maria dirige-se para o quarto de dormir, passa pela porta do quarto, e deita-se na cama. Após observar a distribuição dos sensores pelo quarto, foi definido que o contexto de interesse está associado a detecção de presença pelo sensor M003 (cama), e a interação da pessoa com objeto deve ser maior que 120000 milissegundos. Por sua vez, os sensores M004 (banheiro), M005 (corredor), M006 (porta do quarto) não devem detectar presença, assim indicando que a pessoa permanece na cama. Na Figura 5 é apresentada a regra SWRL elaborada para reconhecer a atividade “dormir”.

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).

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. A regra SWRL desenvolvida para realizar o reconhecimento da atividade “deslocamento da cama para o banheiro” é apresentada na Figura 6.

```
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") ^ swrlx:makeOWLThing(?a, ?an) -> Activity(?an) ^
Activity_Windows(?an, ?w) ^ Activity_Sensor(?an, ?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifacty(?an, ?at) ^
hasWindowBegin(?an, ?w) ^ hasWindowsEnd(?an, ?w)
```

Figura 5. Regra SWRL para a atividade dormir.

```
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) ^ swrlx:makeOWLThing(?a, ?an) -> Activity(?an) ^ Activity_Windows(?an, ?w) ^
Activity_Sensor(?an, ?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifacty(?an, ?atn) ^ hasWindowBegin(?an, ?w) ^
hasWindowsEnd(?an, ?w)
```

Figura 6. Regra SWRL para a atividade deslocamento da cama para banheiro

Cenário 3: Atividade Entrar em Casa

Neste cenário Maria está chegando em casa. Ela abre a porta da garagem e passa pelo corredor em direção à sala de estar. Nessa atividade, considera-se como contexto de interesse os sensores D004 (porta da casa), M030 (em cima da porta), M022 e M021 (presentes no hall de acesso às outras dependências da casa).

A regra SWRL desenvolvida para realizar o reconhecimento da atividade “entrar em casa” é apresentada na Figura 7.

```
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")^swrlx:makeOWLThing(?a, ?an) -> Activity(?an) ^ Activity_Windows(?an, ?w) ^ Activity_Sensor(?an,
?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifacty(?an, ?at) ^ hasWindowBegin(?an, ?w) ^ hasWindowsEnd(?an, ?w)
```

Figura 7. Regra SWRL para a atividade entrar em casa.

Cenário 4: Atividade Sair 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 uma regra empregando como contexto de interesse o sensor M022 (hall de acesso á outras dependências da casa) como evento inicial. Na sequência de eventos, os sensores M021

(hall de acesso), M030 (em cima da porta) e D004 (porta da casa) são empregados para indicar a atividade.

A regra SWRL concebida para realizar o reconhecimento da atividade “sair de casa” é apresentada na Figura 8.

```
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") ^ swrlx:makeOWLThing(?a, ?an) -> Activity(?an) ^
Activity_Windows(?an, ?w) ^ Activity_Sensor(?an, ?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifacty(?an, ?at) ^
hasWindowBegin(?an, ?w) ^ hasWindowsEnd(?an, ?w)
```

Figura 8. Regra SWRL para a atividade sair de casa

Cenário 5: Atividade Trabalhar no Escritório

Maria decide trabalhar no computador, localizado no escritório. Nesse cenário, utilizou-se a seguinte sequência de eventos de sensores. O sensor M028 (porta do escritório) que representa o evento inicial, o qual irá indicar se Maria está entrando na localização de interesse da casa que é escritório. O sensor M026 que representa o objeto de interesse (mesa do computador). Para realizar a detecção dessa atividade optou-se por especificar a necessidade de uma interação superior a 600000 milissegundos com o objeto de interesse.

A regra SWRL elaborada para realizar o reconhecimento da atividade “trabalhar no escritório” é apresentada na Figura 9.

```
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") ^ swrlx:makeOWLThing(?a, ?an) -> Activity(?an) ^
Activity_Windows(?an, ?w) ^ Activity_Sensor(?an, ?s) ^ Activity_Location(?an, ?l) ^ Activity_Artifacty(?an, ?at) ^
hasWindowBegin(?an, ?w) ^ hasWindowsEnd(?an, ?w)
```

Figura 9. Regra SWRL para a atividade trabalhar no escritório.

4.2. Discussão da Avaliação

A verificação da acurácia do modelo ontológico proposto foi baseada na métrica de precisão, que é a proporção de instâncias classificadas corretamente. O uso dessa métrica é justificado pelo fato de a mesma ser amplamente empregada para avaliar a qualidade dos resultados em diversas pesquisas [Japkowicz and Shah 2011].

Aplicou-se como método para avaliação da acurácia o desenvolvido por [van Kasteren et al. 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. Este método foi escolhido por ser amplamente empregado na literatura

Tabela 1. Matriz de confusão dos cenários realizados.

	Dormir	Deslocamento	Entrar	Sair	Trabalhar	Acurácia
Dormir	398	-	-	-	-	99,25%
Deslocamento	-	154	-	-	-	98,08%
Entrar	-	-	400	-	-	92,80%
Sair	-	-	-	405	-	93,96%
Trabalhar	-	-	-	-	150	87,71%

[Al Machot and Mayr 2016]. A tabela 1 apresenta a matriz de confusão com a acurácia obtida pelo modelo proposto nos cenários realizados.

Conforme pode ser visualizado na tabela 1, a acurácia obtida variou entre os diferentes cenários, mas de forma geral se manteve com uma taxa de acertos considerada satisfatória. O número de ocorrências analisadas para cada atividade e a respectiva acurácia foram:

- dormir: acertou 398 atividades de 401 presentes na base de dados, obtendo uma acurácia de 99,25%;
- deslocamento da cama para o banheiro: acertou 154 atividades de 157 presentes na base de dados, obtendo uma acurácia de 98,08%;
- entrar em casa: acertou 400 atividades de 431 presentes na base de dados, obtendo uma acurácia de 92,80%;
- sair de casa: acertou 405 atividades de 431 presentes na base de dados, obtendo uma acurácia de 93,96%;
- trabalhar no escritório: acertou 150 atividades de 171 presentes na base de dados, obtendo uma acurácia de 87,71%.

O modelo proposto acertou ao total 1507 de 1588 atividades, alcançando uma acurácia média de 94.89%. Dentre as atividades, “dormir”teve o melhor resultado, enquanto que a atividade “trabalhar no escritório”o pior. Os resultados obtidos mostraram que o modelo proposto alcançou resultados promissores para ser empregado no reconhecimento de atividades em ambientes de vivência assistida.

5. Trabalhos Relacionados

Em [Ni et al. 2016] é proposto uma arquitetura 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. Para representar o domínio e realizar o reconhecimento de atividades, é empregada uma ontologia, a qual foi modelada com base na ontologia DOLCE + Dns Ultralite (DUL).

O trabalho de [Culmone et al. 2014] apresenta um *framework* baseado em ontologias que visa efetuar consultas semânticas em um repositório de dados contextuais, com o objetivo de prover o reconhecimento de atividades. A ontologia empregada chama-se OnoAALISABETH, sendo esta uma ontologia de domínio explorada em diferentes níveis de abstração, e sendo utilizada no processamento dos dados com uso de regras Jena.

Em [Wongpatikaseree et al. 2012] é proposta uma infraestrutura baseada em ontologias para reconhecimento de atividades de uma casa inteligente. A ontologia é utilizada para distinguir as atividades por meio de conceitos baseados em objetos, localização e postura do corpo humano. Para realizar o processamento dos dados são usados axiomas, sendo executado o raciocínio da ontologia a cada nova inserção na base de conhecimento.

O trabalho de [Chen et al. 2009] apresenta uma arquitetura conceitual de casas inteligentes, explorando a interação de componentes que constituem o ambiente. O foco do trabalho é a metodologia de modelagem semântica, a geração e o gerenciamento de conteúdo. A modelagem e o raciocínio das informações são realizados com o uso de uma ontologia, empregando axiomas para geração de novas informações.

A Tabela 2 mostra uma comparação entre os trabalhos relacionados e o modelo ontológico proposto EXEHDA-AR. Para esta comparação, foram considerados os seguintes critérios: (i) utilização de base de dados na avaliação do modelo; (ii) verificação da acurácia; (iii) exploração de conceitos de objeto e localização; e (iv) tipo de regra empregado para o reconhecimento de atividades.

Tabela 2. Comparação com os trabalhos relacionados.

Trabalhos Relacionados	Utiliza Base de Dados	Verifica a Acurácia	Explora Objeto e Localização	Linguagem das Regras
[Ni et al. 2016]	Não	Não	Parcialmente	SWRL
[Culmone et al. 2014]	Não	Não	Parcialmente	JENA
[Wongpatikaseree et al. 2012]	Não	Não	Utiliza	Axioma
[Chen et al. 2009]	Não	Não	Utiliza	Axioma
EXEHDA-AR	**Sim**	**Sim**	**Utiliza**	**SWRL**

O modelo proposto foi avaliado utilizando uma base de dados pública para avaliar sua proposta, com dados de contextos coletados em um ambiente real, enquanto os outros trabalhos simulam as atividades a serem reconhecidas. O trabalho [Wongpatikaseree et al. 2012] faz uso de uma base de dados relacional com informações de logs de uma casa inteligente, onde após a extração desses dados os mesmos são instanciados em uma ontologia para realizar o reconhecimento de atividades.

Outro diferencial do modelo proposto em relação aos trabalhos relacionados é que neste trabalho é realizada a verificação da acurácia com relação ao reconhecimento de atividades diárias. A análise desta medida representa a qualidade do modelo ontológico frente aos desafios no reconhecimento de atividades humanas por meio de sensores não intrusivos.

6. Considerações Finais

A principal contribuição deste trabalho é a concepção de um modelo ontológico responsável pelo reconhecimento de atividades realizadas por um morador em uma casa inteligente. A ontologia proposta utiliza regras SWRL para realizar o processamento das informações coletadas em um ambiente. Além disso, a proposta possui um módulo de Pré-processamento que emprega o conceito de janela deslizante, permitindo combinar dados coletados em um determinado intervalo de tempo.

As contribuições do modelo proposto foram introduzidas nos módulos de Aquisição e Processamento do *middleware* EXEHDA, introduzindo um módulo de Pré-processamento que se comunica com o módulo de aquisição adicionando novas funcionalidades para realizar a coleta dos dados. No módulo de Processamento é incluída a possibilidade de processar os dados por meio de um modelo ontológico, permitindo um processamento semântico dos dados de contexto coletados. Essa característica mostrou-se oportuna para a detecção de atividades realizadas em uma casa inteligente, permitindo ao usuário não se envolver de forma direta com o procedimento de monitoramento das suas atividades, empregando assim uma abordagem não intrusiva.

O modelo ontológico foi avaliado por meio de um estudo de caso utilizando a base de dados CASA, a qual provê dados reais de uma casa inteligente. Nos testes realizados, foi possível reconhecer diferentes atividades executadas por um morador, as quais são: (i) dormir; (ii) deslocamento da cama para o banheiro; (iii) entrando em casa; (iv) saindo de casa; (v) trabalhar no escritório. A ontologia proposta alcançou uma acurácia média de 94.89% no reconhecimento das atividades presentes na base de dados. Os resultados obtidos foram promissores, estimulando a continuidade da pesquisa.

Dentre os aspectos levantados para continuidade do trabalho, destacam-se: (i) melhorar as regras criadas para trabalhar com outros dados, tornando-as mais genéricas; (ii) desenvolver um raciocínio híbrido para reconhecimento de atividades, empregando além do modelo ontológico técnicas baseadas em aprendizagem de máquina; e (iii) realizar novos testes, utilizando para isso outras bases de dados.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e da FAPERGS (Programa Pesquisador Gaúcho - PqG). Roger S. Machado é Bolsista FAPERGS/CAPES - BRASIL, nível doutorado.

Referências

Al Machot, F. and Mayr, H. C. (2016). Improving human activity recognition by smart windowing and spatio-temporal feature analysis. In *9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, page 56.

Al-Shaqi, R., Mourshed, M., and Rezgui, Y. (2016). Progress in ambient assisted systems for independent living by the elderly. *SpringerPlus*, 5(1):624.

Can, O., Sezer, E., Bursa, O., and Unalir, M. O. (2017). Comparing relational and ontological triple stores in healthcare domain. *Entropy*, 19(1):30.

Chen, L. and Nugent, C. (2009). Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems*, 5(4):410–430.

Chen, L., Nugent, C., Mulvenna, M., Finlay, D., and Hong, X. (2009). Semantic smart homes: towards knowledge rich assisted living environments. In *Intelligent Patient Management*, pages 279–296. Springer.

Cook, D. J. (2012). Learning setting-generalized activity models for smart spaces. *IEEE intelligent systems*, 27(1):32–38.

Culmone, R., Falcioni, M., Giuliodori, P., Merelli, E., Orru, A., Quadrini, M., Ciampolini, P., Grossi, F., and Matrella, G. (2014). Aal domain ontology for event-based human activity recognition. In *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on*, pages 1–6. IEEE.

Japkowicz, N. and Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, New York, NY, USA.

Lai, C.-F., Lai, Y.-X., Yang, L. T., and Chao, H.-C. (2012). Integration of iot energy management system with appliance and activity recognition. In *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, pages 66–71. IEEE.

Lopes, J. L., de Souza, R. S., Geyer, C. F. R., da Costa, C. A., Barbosa, J. L., Pernas, A. M., and Yamin, A. C. (2014). A middleware architecture for dynamic adaptation in ubiquitous computing. *J. UCS*, 20(9):1327–1351.

Marin, M. J. S. and Panes, V. C. B. (2015). Envelhecimento da população e as políticas públicas de saúde. *Revista do Instituto de Políticas Públicas de Marília*, 1(1).

Ni, Q., García Hernando, A. B., and Pau de la Cruz, I. (2016). A context-aware system infrastructure for monitoring activities of daily living in smart home. *Journal of Sensors*, 2016.

Osmani, V., Balasubramaniam, S., and Botvich, D. (2008). Human activity recognition in pervasive health-care: Supporting efficient remote collaboration. *Journal of network and computer applications*, 31(4):628–655.

Perera, C., Member, C. H. L., Jayawardena, S., and Chen, M. (2015). Context-aware computing in the internet of things: A survey on internet of things from industrial market perspective.

Riboni, D., Pareschi, L., Radaelli, L., and Bettini, C. (2011). Is ontology-based activity recognition really effective? In *Pervasive Computing and Communications Workshops (PERCOM)*, pages 427–431. IEEE.

Röcker, C., Ziefle, M., and Holzinger, A. (2014). From computer innovation to human integration: current trends and challenges for pervasive healthtechnologies. In *Pervasive Health*, pages 1–17. Springer.

Singla, G., Cook, D. J., and Schmitter-Edgecombe, M. (2010). Recognizing independent and joint activities among multiple residents in smart environments. *Journal of ambient intelligence and humanized computing*, 1(1):57–63.

van Kasteren, T. L., Alemdar, H., and Ersoy, C. (2011). Effective performance metrics for evaluating activity recognition methods. *ARCS 2011*.

Wongpatikaseree, K., Ikeda, M., Buranarach, M., Supnithi, T., Lim, A. O., and Tan, Y. (2012). Activity recognition using context-aware infrastructure ontology in smart home domain. In *Knowledge, Information and Creativity Support Systems (KICSS), 2012 Seventh International Conference on*, pages 50–57. IEEE.

Yala, N., Fergani, B., and Fleury, A. (2015). Feature extractionand incremental learning to improve activity recognition on streaming data. In *Evolving and Adaptive Intelligent Systems (EAIS), 2015 IEEE International Conference on*, pages 1–8. IEEE.