

## Arcabouço de Integração para Residência Assistiva Baseado em Ontologias

L. Ramos\*, R. C. Betini\*\*

\*Departamento Acadêmico de Informática / UTFPR, Curitiba, Brasil

\*\* Departamento Acadêmico de Eletrotécnica / UTFPR, Curitiba, Brasil

leticia\_ramos@hotmail.com

**Resumo:** Segundo a Classificação Internacional de Funcionalidade, Incapacidade e Saúde (CIF) a funcionalidade e a incapacidade dos indivíduos são determinadas também pelo contexto do ambiente. Neste sentido, residências assistivas fornecem suporte para a redução das barreiras nas atividades diárias. Porém, a heterogeneidade de dispositivos e aplicações dificulta a interoperabilidade e integração nestes ambientes. Este artigo descreve um arcabouço para integração em ambientes de domótica assistiva. O arcabouço é baseado em uma ontologia de domínio utilizada como suporte semântico. O modelo permite uma maior flexibilidade da escolha de sensores, dispositivos e aplicações de acordo com as necessidades de cada indivíduo.

**Palavras-chave:** Ontologia, residências assistivas, interoperabilidade semântica.

**Abstract:** *According to the International Classification of Functioning, Disability and Health (ICF) functionality and disability are also determined by the environment. Assistive homes could reduce the barriers in daily activities. However, the heterogeneity of devices and applications makes interoperability and integration a challenge for these environments. This article describes a framework for integration in assistive home automation. The framework is based on domain ontology for semantic support. The model allows flexibility of sensors, devices and applications according to the needs of each individual.*

**Keywords:** *Ontology, Assistive Homes, Semantic interoperability.*

### Introdução

Segundo o Censo 2010 [1], 45 milhões de brasileiros possuem algum tipo de deficiência. Este número corresponde a aproximadamente 24% da população do país. Sendo que 13 milhões apresentam deficiências graves, com grandes dificuldades ou impossibilidade total, necessitando de auxílio para executar tarefas diárias. A Classificação Internacional de Funcionalidade, Incapacidade e Saúde (CIF) [2], desenvolvida pela Organização Mundial da Saúde (OMS) descreve as funcionalidades e incapacidades relacionadas às condições de saúde, identificando o impacto na participação social e no ambiente. Neste sentido, a automação residencial assistiva pode auxiliar o indivíduo na execução de atividades diárias de forma

independente, diminuindo as barreiras relacionadas aos fatores ambientais.

Residências assistivas devem adaptar-se às necessidades do usuário permitindo diferentes sensores e interfaces. Porém essa heterogeneidade dificulta a interoperabilidade entre as aplicações e dispositivos tanto no nível semântico como sintático [3]. Ou seja, os dados compartilhados devem ter o mesmo significado e contexto para todas as aplicações envolvidas.

Uma ontologia, para o contexto computacional, é um tipo de formalismo para a representação do conhecimento. Ela permite uma compreensão comum das informações, possibilita o reuso e faz suposições explícitas em um domínio. [4]. Ontologias são utilizadas de forma efetiva como solução semântica para comunicação [5], porém no contexto da automação de residências não existe um modelo unificado que permita a representação do domínio de forma completa, considerando o ambiente, o indivíduo e as aplicações.

Este artigo propõe uma ontologia como modelo semântico para uma residência assistiva que permita a utilização de diversos sensores e aplicações de acordo com o perfil do usuário, além de auxiliar na descoberta de novos serviços nestes ambientes. Por último é proposto um arcabouço para aplicação da ontologia com objetivo de validar o modelo. A ontologia proposta é denominada AssistiveHome.

### Materiais e métodos

A metodologia *Ontology Development 101* [4] foi utilizada como base para o desenvolvimento da ontologia proposta neste artigo. O primeiro passo é a definição do domínio e escopo. Após, deve-se considerar o reuso de ontologias existentes. O terceiro passo consiste em enumerar os termos importantes para a ontologia. O quarto passo considera a criação das classes e as hierarquias. Então são definidos os atributos das classes, os tipos e valores permitidos. Por último são criadas as instâncias das classes.

### Ontologias para Residências Inteligentes

A modelagem semântica de uma residência está presente no trabalho de diversos autores. Para considerar o reuso, algumas classes da ontologia descrita nesse artigo foram importadas das ontologias *DogOnt* [6] e *Ontology for Smart Spaces* [7], que serão analisadas a seguir.

**DogOnt [6]** – O conceito principal da ontologia é a modelagem de dispositivos domésticos independente da tecnologia. Objetos eletronicamente controláveis como sensores e atuadores são representados através da classe *Controllable*. Esses dispositivos possuem funcionalidades descritas através da classe *Functionality*. Por exemplo, a função *OnOffFunctionality* descreve a função de ligar e desligar um lâmpada. As funcionalidades relacionam-se com comandos ou notificações. A classe *Command* indica os comandos necessários para acionar uma função de controle, no caso, a classe *OnOffFunctionality* relaciona-se com os comandos *OnCommand* e *OffCommand*. Da mesma forma, uma funcionalidade pode representar uma função de notificação relacionando-se com a classe *Notification*, por exemplo, a funcionalidade *OnOffNotification* possui as notificações *OnNotification* e *OffNotification*. As classes *State* e *StateValue* representam os estados de um dispositivo através de valores discretos ou contínuos. O DogOnt ainda possui as classes *BuildingEnvironment* e *UnControllable* para descrever respectivamente os ambientes e objetos não controláveis de uma residência. A ontologia permite a interoperabilidade entre os dispositivos, porém, ela não considera o usuário, suas atividades e as aplicações que processam as informações de acordo com o perfil de um indivíduo.

**Smart Spaces [7]** – Propõe que um ser vive e interage em um ambiente com certa dinâmica e é composta de três classes principais. A classe *Environment* representa o ambiente e divide-se em duas subclasses. A subclasse *Tangible* que inclui objetos como dispositivos, eletrônicos e móveis. E a subclasse *Intangible* que corresponde às entidades computacionais. A propriedade de objeto *hasReferencialLocation* define a posição referencial, como acima e abaixo, entre um objeto da classe *Environment* e outro objeto. A classe também define a localização absoluta através de propriedades de dados que representam as coordenadas no espaço (x, y, z). A classe *Being* classifica o ser como pessoa, animal, etc. Essa classe também descreve o perfil de uma pessoa baseado na CIF [2] através de três subclasses. A classe *PersonalProfile* refere-se às características pessoais tais como idade e gênero. O *HealthProfile* compreende o estado de saúde do indivíduo e o *OrganicProfile* apresenta as funções e estruturas do corpo. Por último a classe *Dynamic* descreve as atividades computacionais e de um ser. Um dos diferenciais da ontologia é o conceito de localização relativa entre dois objetos. Outro ponto positivo é a descrição de uma pessoa em relação à saúde e barreiras funcionais. Porém a ontologia não favorece a interoperabilidade, já que possui apenas a classe principal referente aos dispositivos.

## Ontologia AssistiveHome

A ontologia AssistiveHome foi desenvolvida para representar o domínio de uma residência considerando o ambiente, o indivíduo e as aplicações. As classes principais da ontologia e suas propriedades estão

disponíveis na Figura 1 e serão detalhadas a seguir. As classes amarelas foram importadas de DogOnt [6], os conceitos verdes são derivados de SmartSpaces [7] e as classes de cor azul representam as novas classes.

**Classe *BuildingEnvironment*** – Descreve a residência e seus cômodos, ela mantém o nome da classe descrita por Bonino, porém foi adicionado o conceito de localização absoluta no ambiente, conforme proposto por Abdulrazak [7], através das propriedades de dados x, y e z. Cada cômodo como, por exemplo, quarto e cozinha representa uma subclasse de *BuildingEnvironment*.

**Classes de Controle** – Importa da ontologia DogOnt [6] os conceitos de dispositivos controláveis, funcionalidades, notificações e estados que relacionam-se conforme exemplificado no item anterior. Porém, a classe *Controllable* não oferecia suporte à localização absoluta e relativa. Assim, as propriedades de dados que representam as dimensões e a propriedade de objetos *hasReferencialLocation* descritos por Abdulrazak [7] foram incluídas.

**Classe *BuildingFunctionality*** – Descreve as funções que podem estar presentes em uma residência assistiva. Ela compreende funcionalidades de automação, como controle de iluminação e funções de controle de saúde como monitoramento de pressão arterial.

**Classe *Agent*** – Representa um agente de software instanciado no ambiente. Um agente pode suportar funcionalidades da classe *BuildingFunctionality* ou pode representar dispositivos da classe *Controllable*. Um agente de controle envia e recebe informações de acordo com as funcionalidades e comandos descritos na ontologia. Enquanto agentes de funcionalidade processam diversas informações recebidas por agentes de controle para efetuar uma mudança no ambiente. Os agentes podem buscar outros agentes que controlam um determinado tipo de dispositivo para certo contexto.

**Classe *Person e FunctionalProfile*** – As classes referentes a pessoa representam um indivíduo na residência e o seu perfil funcional. Baseada parcialmente nos conceito proposto por Abdulrazak [7], a classe *Person* possui atributos que caracterizam o indivíduo, como altura, peso e idade criados como propriedades de objetos. Enquanto o perfil de funcionalidade originou uma nova classe denominada *FunctionalProfile* composta de subclasses de acordo com a barreira funcional do indivíduo.

## Definição do arcabouço

O arcabouço proposto tem por objetivo a integração entre aplicações responsáveis pela automação de uma residência através de uma arquitetura de agentes. Ele utiliza a ontologia como linguagem de comunicação e para descoberta de serviços. A ontologia oferece suporte semântico, porém para que a comunicação seja efetiva a arquitetura segue os protocolos descritos pela FIPA [8] para comunicação em seus três níveis: protocolos de interação, atos comunicativos e linguagens de conteúdo.

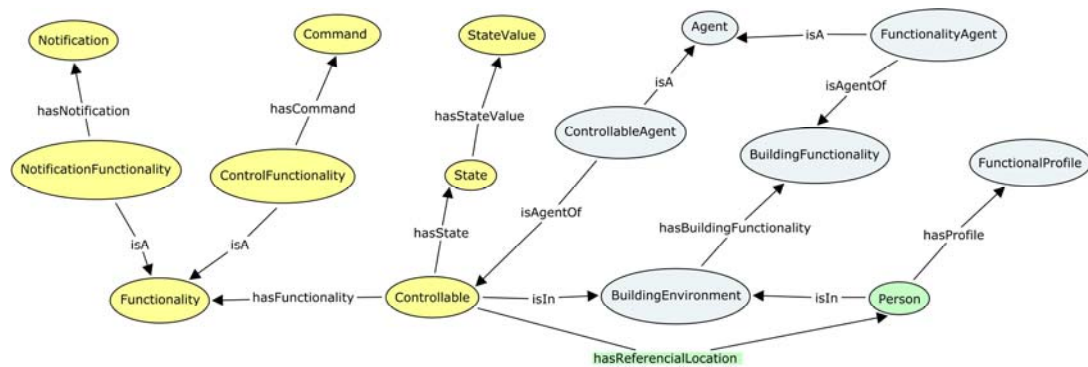


Figura 1: Classes principais da ontologia AssistiveHome.

Cada aplicação do arcabouço é representada por uma instância da classe *Agent*. Um agente pode ser responsável por uma funcionalidade, *FunctionalityAgent*, ou pelo controle de um dispositivo, *ControllableAgent*. Os agentes interagem através do protocolo FIPA-Request [8] quando a intenção é solicitar alguma ação do outro agente. Neste caso, o protocolo é necessário, pois o agente receptor da mensagem pode aceitar ou recusar a solicitação além de informar o resultado de sucesso ou falha. Existe também a interação apenas para informar um agente interessado sobre o estado de um dispositivo, neste caso o agente apenas envia uma mensagem de *Inform* [8] para os agentes interessados.

A ontologia atua na camada de software da aplicação. Desta forma, para cada novo dispositivo é necessário que um driver seja desenvolvido com a finalidade de traduzir os sinais recebidos em dados computacionais ou vice versa. Da mesma forma, interfaces de comunicação com o usuário podem ser criadas de acordo com a necessidade, porém para o arcabouço uma interface também é um agente de funcionalidade.

O arcabouço também possui um agente denominado gerente responsável por manter e criar as instâncias das classes da ontologia. Quando um novo agente é incluído no arcabouço ele deve enviar uma mensagem para o agente de gerência informando o tipo de agente, a funcionalidade ou controle que exerce e o ambiente em que está localizado. O gerente então cria as instâncias pertinentes e assim é construída a representação da residência no domínio. Da mesma forma, quando um agente necessita descobrir, por exemplo, qual agente exerce determinada função em um ambiente ou a localização de uma pessoa ele envia uma query SPARQL para o agente gerente que retorna o resultado ou falha. O agente gerente permite que novos sistemas sejam adicionados ao longo do tempo e possibilita a análise do contexto atual da residência.

### Aplicação prática

Para validar a ontologia e arcabouço foi construído um protótipo para automação de cinco dispositivos. A aplicação foi desenvolvida em um PC configurado com um processador Intel Pentium 4 de 3.00 GHz e 512 MB de memória RAM. Uma placa para automação desenvolvida pela Universidade Tecnológica Federal do Paraná foi utilizada para interface entre o computador e

os dispositivos. A ontologia proposta foi desenvolvida através da ferramenta Protege<sup>1</sup> que tem por função a representação da ontologia e criação de instâncias. Para a manipulação da ontologia empregou-se a API (*Application Program Interface*) denominada Jena<sup>2</sup>. Através dela é possível realizar consultas SPARQL<sup>3</sup> e manipular triplas RDF. Os agentes foram desenvolvidos no framework Jade<sup>4</sup> que possui suporte para o protocolo de comunicação FIPA-ACL.

Na inicialização do sistema criaram-se as instâncias de dois cômodos da classe *BuildingEnvironment*: um quarto e uma sala. O quarto possui uma lâmpada, um ar condicionado, um sensor de temperatura e um sensor de presença. A sala possui uma lâmpada, um atuador para cortina, um ar condicionado, um sensor de presença, um sensor de iluminação e um sensor de temperatura. Cada dispositivo é controlado por um agente. Quando inicializados estes agentes solicitaram ao agente de gerência a criação das instâncias na ontologia. Por exemplo, para a lâmpada do quarto, foi criada a instância do agente da classe *ControllableAgent* e de uma lâmpada da classe *Lamp*, subclasse da classe *Controllable*. O agente relaciona-se com a lâmpada através da propriedade *isAgentOf* enquanto a lâmpada informa que está localizada no quarto através da propriedade *isIn*. Os comandos e notificações utilizados como linguagem de comunicação referem-se às classes *Notification* e *Command* da ontologia. Os agentes de sensores simulam os dispositivos reais através de uma interface computacional conforme Figura 2. Enquanto os atuadores são representados por lâmpadas que quando acesas simulam a alteração de estado, Figura 3.

Também foram adicionados dois agentes de funcionalidade no ambiente: um para controle de temperatura e um para controle de iluminação. O primeiro criou uma instância da classe *TemperatureRegulationFunctionality* e o segundo uma instância da classe *LightRegulationFunctionality*, ambos subclasses de *BuildingFunctionality*.

Quando, por exemplo, o agente que controla o sensor de iluminação da sala detecta uma mudança ele envia uma mensagem para o agente da funcionalidade

<sup>1</sup> <http://protege.stanford.edu/doc/users.html>

<sup>2</sup> <http://jena.apache.org/documentation/inference/>

<sup>3</sup> <http://www.w3.org/TR/rdf-sparql-query/>

<sup>4</sup> <http://jade.tilab.com>

de iluminação da sala informando o novo estado. Nesse caso o botão noite foi pressionado. Como não existe nenhuma pessoa no ambiente o agente não efetua nenhuma alteração. Em seguida, o agente do sensor de presença é ativado, enviando uma mensagem ao agente de iluminação. O agente de iluminação interpreta que existe uma pessoa na sala e solicita ao agente de gerência a referência de agentes que controlem lâmpadas na sala através de uma sentença SPARQL. De acordo com o endereço recebido o agente de iluminação envia a mensagem de mudança de estado (*OnCommand*) para o agente da lâmpada. Após algum tempo, o sensor de iluminação informa que a luminosidade externa é alta e o agente de iluminação solicita aos respectivos agentes a abertura da cortina e o desligamento da lâmpada.

Em um segundo momento uma nova lâmpada é incluída no ambiente da sala. Quando o agente de funcionalidade de iluminação pesquisa por agentes de lâmpadas na sala, ele receberá a referência de dois agentes. Para essa situação o agente de iluminação busca a lâmpada que se relaciona com o usuário através da posição relativa entre dois objetos. Por exemplo, uma lâmpada pode estar acima de um indivíduo, através da propriedade de objeto *upTo*.



Figura 2: Simulação de sensores para testes

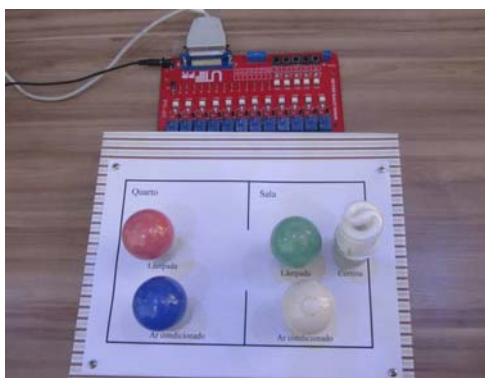


Figura 3: Placa de automação e protótipo

## Discussão e Conclusão

O desenvolvimento de um sistema de automação residencial inteligente é bastante abrangente, ele envolve conceitos relacionados ao usuário, ao ambiente e aos dispositivos. A ontologia AssitiveHome foi

desenvolvida para representar de forma unificada o contexto de uma residência. Existem diversas outras ontologias para o domínio de automação, porém cada modelo pretende solucionar um problema específico. Por exemplo, DogOnt [5], DomoMI [9] descrevem os dispositivos domóticos de forma detalhada, porém não incluem o conceito de posição absoluta. Já CONON [10] e Ontology for Smart Spaces [6] modelam ambientes pervasivos, porém não são específicas para automação residencial e não descrevem todos os dispositivos de automação, seus estados e funções.

O arcabouço desenvolvido permite a realização de inferências em um ambiente, assim dispositivos e aplicações podem ser adicionados ao longo do tempo e de forma flexível. Este tipo de adaptabilidade é importante para residências assistivas, pois diferentes indivíduos possuem diferentes barreiras em relação aos fatores ambientais.

Apesar da classe de dispositivo importar o modelo de Bonino [5] o arcabouço proposto utiliza a ontologia como referência para comunicação entre os agentes, o que não havia sido explorado no modelo anterior. A especificação detalhada do modelo de conteúdo das mensagens e a validação em relação ao protocolo FIPA serão abordadas em um artigo futuro.

## Referências

- [1] IBGE. Censo Demográfico. 2010. Disponível em: <http://www.censo2010.ibge.gov.br>
- [2] World Health Organization. International Classification of Functioning, Disability, and Health: ICF. World Health Organization. 2001.
- [3] Edwards, W. K., & Grinter, R. E. At home with ubiquitous computing: seven challenges. In Ubicomp 2001: Ubiquitous Computing. Springer Berlin Heidelberg; 2010; p. 256-272.
- [4] Noy, N. F., McGuinness, D. L. Ontology Development 101: A Guide To Creating Your First Ontology, 2001.
- [5] Lewis, G. A., Morris, E., Simanta, S., & Wraga, L. Why standards are not enough to guarantee end-to-end interoperability. In Proceedings of the Seventh International Conference on Composition-Based Software Systems. IEEE; 2008; p. 164-173.
- [6] Bonino, D.; Corno, F. Dogont-Ontology Modeling For Intelligent Domotic Environments. The Semantic Web-Iswc 2008; p 790-803.
- [7] Abdulrazak, B., Chikhaoui, B., Guoin-Vallerand, C., & Fraikin, B. A standard ontology for smart spaces. International Journal of Web and Grid Services, 6(3); 2010; p 244-268.
- [8] Foundation for Intelligent Physical Agents FIPA. FIPA Communicative Act Library Specification, 2002.
- [9] Sommaruga, L.; Perri, A.; Furfari, F. Domoml-Env: An Ontology For Human Home Interaction. Swap; 2005.
- [10] 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; p. 18-22.