

(Meta)Modelagem de Espaços Inteligentes Pessoais e Espaços Inteligentes Fixos para Aplicações Ubíquas

Marcos Alves Vieira¹, Sergio T. Carvalho²

¹Instituto Federal de Educação, Ciência e Tecnologia Goiano (IF Goiano)
Iporá – GO – Brasil

²Instituto de Informática – Universidade Federal de Goiás (UFG)
Goiânia – GO – Brasil

marcos.vieira@ifgoiano.edu.br, sergio@inf.ufg.br

Abstract. *User mobility and the increasing number of smart spaces, fostered also by the Internet of Things (IoT) and the Web of Things (WoT), can lead to smart spaces overlap, where a certain smart object is configured in different smart spaces, whether fixed or personal. In addition, smart spaces are complex and difficult to model and maintain, as, among other factors, they have to deal with different smart objects. This paper proposes the use of Model-Driven Engineering techniques to enable modeling of ubiquitous computing scenarios, considering the coexistence between fixed and personal smart spaces. The validation of the proposal was carried out by modeling a scenario with the aid of a graphical modeling tool, constructed to produce models conforming to the proposed metamodel.*

Resumo. *A mobilidade dos usuários e o crescente número de espaços inteligentes tanto pessoais quanto fixos, fomentados também pela Internet das Coisas e Web das Coisas, podem levar à sobreposição de espaços inteligentes, onde um determinado objeto inteligente pode ser utilizado em diferentes espaços inteligentes. Além disso, espaços inteligentes são complexos e difíceis de modelar e manter, pois, entre outros fatores, precisam lidar com diferentes objetos inteligentes. Este trabalho apresenta o uso de técnicas de Engenharia Dirigida por Modelos para possibilitar a modelagem de cenários de computação ubíqua, considerando a coexistência entre espaços inteligentes fixos e pessoais. A validação da proposta se deu pela modelagem de um cenário com o auxílio de uma ferramenta gráfica de modelagem construída para produzir modelos em conformidade com o metamodelo proposto.*

1. Introdução

O conceito de ubiquidade vislumbrado por Mark Weiser [Weiser 1991] está sendo concretizado pela recente convergência, disseminação e popularização das tecnologias de rádio, dos microprocessadores e dos dispositivos eletrônicos digitais pessoais, aliadas aos novos paradigmas computacionais, como a Internet das Coisas (*Internet of Things* - IoT) e a Web das Coisas (*Web of Things* - WoT). Esses dispositivos podem ser integrados aos espaços físicos cotidianos, transformando-os, assim, em espaços inteligentes.

Em complementação aos espaços inteligentes tradicionais, que em geral são fixos e confinados a uma determinada área, surgiu recentemente o conceito de espaços inteligentes pessoais [Dolinar et al. 2008]. Um espaço inteligente pessoal (do inglês, *Personal Smart Space* - PSS) é composto pelo conjunto de objetos inteligentes que um usuário carrega consigo e o acompanha durante sua mobilidade. O PSS está sempre disponível e atua como uma interface entre o usuário e os serviços disponíveis em seu próprio espaço inteligente pessoal e entre este e os demais espaços inteligentes, sejam fixos ou pessoais [Dolinar et al. 2008].

Uma abordagem que vem sendo considerada no desenvolvimento de sistemas ubíquos é a adoção de conceitos da Engenharia Dirigida por Modelos (do inglês, *Model-Driven Engineering* - MDE). A MDE considera os modelos como os principais artefatos no desenvolvimento de um sistema. Assim, além de descrever ou documentar um software, os modelos também atuam no seu desenvolvimento, manutenção e operação [Schmidt 2006]. Os modelos são geralmente construídos utilizando-se linguagens de modelagem específicas de domínio (do inglês, *Domain-Specific Modeling Language* - DSML), que, por sua vez, são definidas por um meta-modelo [López-Fernández et al. 2015].

Propõe-se neste trabalho um metamodelo construído sobre os conceitos da MDE, para modelagem de cenários de computação ubíqua compostos de espaços inteligentes pessoais e fixos, objetos inteligentes, aplicações ubíquas e políticas de acesso aos objetos inteligentes e aplicações ubíquas. O metamodelo proposto visa (i) facilitar a modelagem de espaços inteligentes; (ii) permitir tratar os usuários como entidades de primeira classe, desacoplados do espaço fixo e acompanhados de seu próprio PSS; e (iii) possibilitar vocabulário e representação gráfica comuns entre os pesquisadores da área para os conceitos específicos do domínio de espaços inteligentes.

No presente trabalho, o termo *cenário de computação ubíqua* refere-se a:

Cenário de computação ubíqua. *Um conjunto modelado de espaços inteligentes tanto pessoais quanto fixos, pessoas, aplicações ubíquas, objetos inteligentes e suas inter-relações.*

O restante do trabalho está organizado da seguinte forma: a Seção 2 apresenta um cenário para facilitar o entendimento da proposta e detalha o metamodelo proposto para modelagem de cenários de computação ubíqua; a Seção 3 apresenta uma ferramenta de modelagem gráfica e, em seguida, demonstra sua utilização para modelagem do cenário descrito; a Seção 4 apresenta e discute sobre os trabalhos relacionados; por fim, a Seção 5 traz as conclusões acerca do presente trabalho e indica possíveis trabalhos futuros.

2. Proposta

Como forma de facilitar o entendimento da proposta e seu contexto, esta seção traz primeiramente a descrição de um cenário, em sua Subseção 2.1, o qual também serve de base para a validação da proposta, detalhada na Seção 3. Na Subseção 2.2 encontra-se o conceito de sobreposição de espaços inteligentes utilizado neste trabalho e, na Subseção 2.3, é apresentado o metamodelo para modelagem de cenários de computação ubíqua.

2.1. Cenário

A seguir é descrito o cenário de uma casa inteligente na qual residem dois pacientes idosos que necessitam de cuidados médicos. Este cenário possui considerável relevância, que é corroborada pela preocupação de diversos autores em fornecer ferramentas de suporte à assistência domiciliar e permitir o monitoramento remoto de pacientes (e.g., [Carvalho et al. 2011], [Carvalho et al. 2012], [Vieira and Carvalho 2015]).

O Senhor Genaro e a Dona Rute formam um casal de idosos que vivem juntos. Sr. Genaro possui hipertensão arterial, e, por conta disso, desenvolveu recentemente arritmia cardíaca. D. Rute também é hipertensa, além de sofrer de osteoporose. Ambos tomam medicação de forma contínua e recebem recomendação médica para realizar exercícios físicos diários. Na residência do casal foram instalados vários objetos inteligentes, tais como sensores e atuadores fixos, conforme ilustrado na Figura 1. Dispositivos médicos vestíveis são usados por eles de acordo com suas enfermidades. Esses dispositivos enviam informações para uma aplicação ubíqua de saúde pessoal, instalada em seus smartphones. Tanto o Sr. Genaro quanto a D. Rute possuem os seus próprios espaços inteligentes pessoais, compostos por seus dispositivos vestíveis, seus smartphones e a aplicação ubíqua de saúde pessoal. Demais objetos inteligentes são adicionados ao espaço inteligente pessoal sob demanda, como é o caso da TV da sala de estar, que é usada para exibir informações e alertas da aplicação ubíqua de saúde pessoal de cada um dos moradores, tais como mensagens com o horário para tomar as medicações e lembretes para a realização de exercícios físicos ou para aferir seu peso na balança digital, que é compartilhada entre o casal. Além disso, há ainda na casa três outras aplicações ubíquas (de segurança, de bem-estar, e de economia de energia), que estão hospedadas em um espaço inteligente fixo que abrange todos os cômodos da casa.

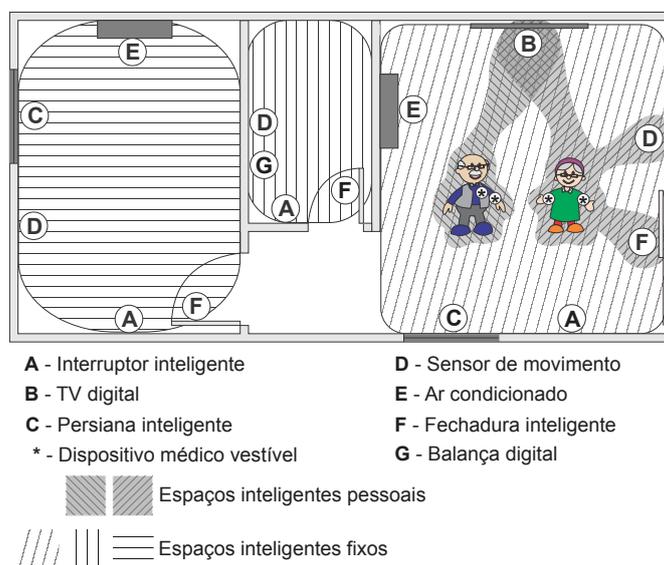


Figura 1. Casa inteligente, na qual dois usuários compartilham objetos inteligentes simultaneamente configurados em diferentes espaços inteligentes.

2.2. Sobreposição de Espaços Inteligentes

Apesar de ser um problema típico e facilmente observável da área de espaços inteligentes, não há uma definição clara sobre o conceito de sobreposição de espaços inteligente. Di-

ante disso, este trabalho propõe e usa a seguinte definição:

Sobreposição de espaços inteligentes. *A sobreposição de espaços inteligentes ocorre quando um ou mais objetos inteligentes estão configurados em espaços inteligentes distintos, sejam estes pessoais ou fixos.*

Por “configurado” considera-se que o objeto inteligente está pronto para ser utilizado pela aplicação ubíqua. Em outras palavras, “configurado” significa que o objeto inteligente foi localizado, instalado e configurado pela aplicação, tornando seus serviços disponíveis perfeitamente utilizáveis.

A sobreposição de espaços inteligentes pode ocorrer pela mobilidade de um usuário que carrega consigo seu espaço inteligente pessoal. Dessa forma, ele pode interagir com outros espaços, tanto fixos quanto pessoais de outro usuário, requisitando acesso a dispositivos que estão configurados em outros espaços inteligentes. Outra forma de sobreposição ocorre quando um único dispositivo está configurado em diferentes espaços inteligentes, novamente, fixos ou pessoais.

2.3. Metamodelo para Cenários de Computação Ubíqua

Esta seção detalha o metamodelo proposto para configuração de cenários compostos por espaços inteligentes pessoais e espaços inteligentes fixos, além de considerar a questão da sobreposição de espaços inteligentes. A arquitetura, suas abstrações e relacionamentos expressam sua sintaxe abstrata, apresentada na Figura 2. Como forma de facilitar a modelagem de instâncias do metamodelo, sua sintaxe concreta foi definida como um conjunto de ícones gráficos. A sintaxe concreta do metamodelo é apresentada na Tabela 1.

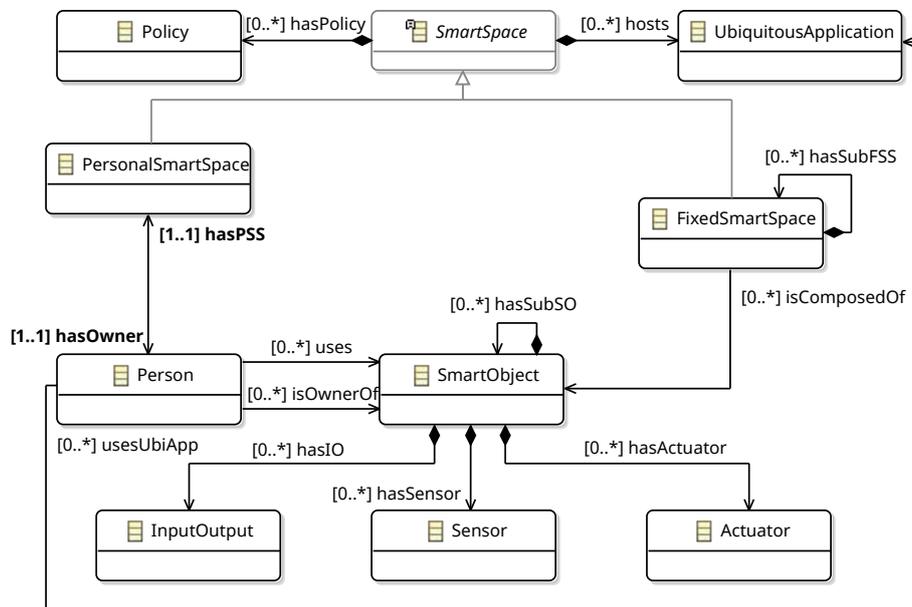


Figura 2. Metamodelo proposto para modelagem de cenários compostos por espaços inteligentes pessoais e fixos.

O metamodelo foi desenvolvido no *Eclipse Modeling Framework (EMF)*¹ sendo, portanto, uma instância do meta-metamodelo Ecore. A seguir apresenta-se a descrição de cada um dos componentes do metamodelo:

¹<http://www.eclipse.org/modeling/emf/>

Tabela 1. Representação gráfica da sintaxe concreta do metamodelo para cenários de computação ubíqua.

Entidade	Representação Gráfica
Policy	
UbiquitousApplication	
PersonalSmartSpace	
Person	
FixedSmartSpace	
SmartObject	
Sensor	
Actuator	
InputOutput	
Associações: hasPolicy, hosts, hasPSS, hasOwner, hasSubFSS, hasSubSO, hasIO, hasSensor, hasActuator	
Associações: uses, isOwnerOf, isComposedOf	

- SmartSpace: um espaço inteligente é uma generalização de espaços inteligentes pessoais (PersonalSmartSpace) e espaços inteligentes fixos (FixedSmartSpace), que, por sua vez, hospedam as aplicações ubíquas (UbiquitousApplication). Essa é uma metaclassse abstrata.
- Policy: um espaço inteligente pode possuir políticas (Policy) para a utilização dos objetos inteligentes (SmartObject) pelas aplicações ubíquas e pelas pessoas (Person) ou para a utilização das aplicações ubíquas pelas pessoas. Estas políticas podem ser, por exemplo, do tipo evento-condição-ação (do inglês, *event-condition-action* - ECA). O detalhamento desta metaclassse foge do escopo deste trabalho. Contudo, é possível adaptar e usar abordagens já existentes em conjunto com o metamodelo.
- UbiquitousApplication: representa uma aplicação projetada para ser executada em um espaço inteligente pessoal (PersonalSmartSpace) ou fixo (FixedSmartSpace). Uma aplicação ubíqua é naturalmente sensível ao contexto, ou seja, ela pode reagir a mudanças “sensoriadas” no ambiente. Essas informações são repassadas à aplicação por meio de um serviço de contexto externo, como o *Hermes* [Veiga et al. 2014] ou *Context Toolkit* [Dey et al. 1999]. A aplicação ubíqua pode se basear no contexto obtido para executar alguma ação e essa ação pode levar em consideração as políticas (Policy) definidas.
- PersonalSmartSpace: define um espaço inteligente pessoal. Cada pessoa, representada por uma entidade Person, possui seu próprio espaço inteligente pessoal e este pode ser estendido, na medida em que novos objetos inteligentes, pertencentes ou não a outros espaços inteligentes, são nele configurados.
- Person: define uma pessoa. As associações possíveis entre uma pessoa e um objeto inteligente são:
 - uses: define que um objeto inteligente pode eventualmente ser compartilhado e usado simultaneamente entre espaços inteligentes, configurando

na sobreposição de espaços inteligentes.

- `isOwnerOf`: define que um determinado objeto inteligente é de uso particular de uma pessoa. Dessa forma, por padrão, um objeto inteligente ligado a uma pessoa por meio da associação `isOwnerOf` não pode ser compartilhado com (*i.e.*, utilizado por) outra pessoa que não seja seu dono.
- `FixedSmartSpace`: define um espaço inteligente fixo. Um espaço inteligente fixo pode ser constituído por outros espaços inteligentes fixos. Por exemplo: cada sala do Instituto de Informática de uma determinada universidade pode ser um espaço inteligente fixo (ou *Fixed Smart Space* - FSS). O prédio do Instituto de Informática, por sua vez, pode ser um FSS, constituído por suas várias salas. E cada instituto de uma universidade pode fazer parte do FSS de um de seus câmpus.
 - `isComposedOf`: um espaço inteligente é composto por objetos inteligentes, que são compartilhados (ou não) entre as pessoas que nele entram e saem.
- `SmartObject`: representa um objeto inteligente. Um objeto inteligente pode ser composto por sensores (`Sensor`), atuadores (`Actuator`) ou interfaces de entrada e saída (`InputOutput`). Um *smartphone*, por exemplo, é um objeto inteligente que possui diversos sensores, tais como acelerômetro e bússola; atuadores, como *vibracall* e alto-falantes; e interfaces de entrada e saída, como tela e botões físicos (liga/desliga, aumenta/diminui volume, etc.). Além disso, um objeto inteligente pode ser constituído por outros objetos inteligentes. Pode-se ainda desejar modelar cada um destes dispositivos como um objeto inteligente independente, por exemplo, um sensor de temperatura. Dessa forma, é possível criar modelos em diferentes níveis de granularidade.

3. Validação da Proposta

Como forma de identificar como os pesquisadores da área de metamodelagem têm validado e avaliado suas propostas de metamodelos, foi conduzida uma Revisão Sistemática da Literatura (RSL). Foram identificados 438 trabalhos nas bibliotecas digitais das bases de pesquisa definidas no protocolo da RSL, dos quais 60 foram detectados como duplicados e 281 foram removidos pelos critérios de exclusão. Dentre os 97 trabalhos incluídos, constatou-se que a forma mais comum de validação dos metamodelos é a construção de uma ferramenta para produzir modelos em conformidade com o metamodelo proposto, em conjunto com a sua utilização para modelagem de cenários reais ou fictícios.

A validação do metamodelo apresentado neste trabalho seguiu, portanto, as tendências adotadas pelos pesquisadores da área. Por limitações de espaço, apenas o cenário apresentado na Subseção 2.1 foi modelado como uma instância do metamodelo proposto. Sendo assim, um editor gráfico foi implementado utilizando o *Eclipse Graphical Modeling Framework* (GMF)². Essa ferramenta foi incrementada com o uso de linguagens da família *Epsilon*³ e sua construção foi apoiada pela ferramenta *Eugenia*, que também integra a família *Epsilon*.

A Figura 3 apresenta a ferramenta de modelagem sendo utilizada para a configuração de um cenário de computação ubíqua. A janela da ferramenta é dividida

²<https://www.eclipse.org/gmf-tooling/>

³<https://www.eclipse.org/epsilon/>

em duas partes: (i) e (ii). Em (i), há a área reservada para a construção do modelo. Em (ii), estão os ícones que representam os conceitos do metamodelo, agrupados em uma barra de ferramentas (ou paleta). A barra de ferramentas é dividida em duas partes: (a) ferramentas para criação de elementos, ou seja, instâncias dos tipos definidos no metamodelo; e (b) ferramentas para criação de associações entre os elementos.

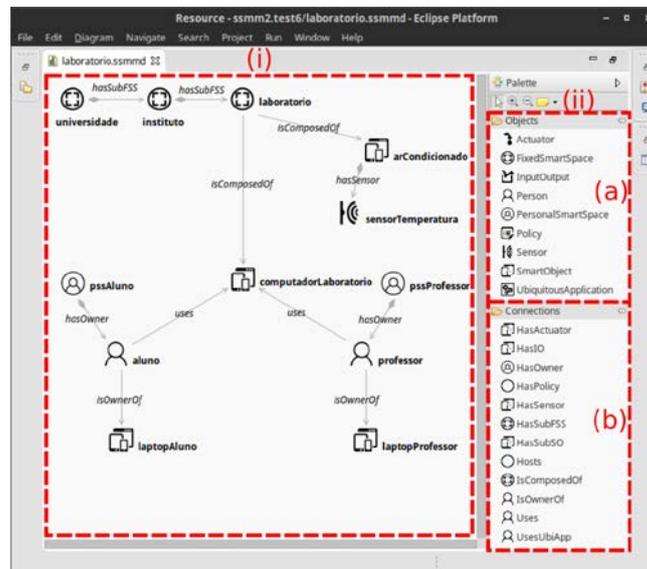


Figura 3. Ferramenta de modelagem sendo utilizada para modelagem de um cenário de computação ubíqua.

A ferramenta de modelagem permite validar o modelo construído para garantir sua conformidade com o seu metamodelo. Caso alguma inconsistência seja encontrada, ela é apontada, permitindo sua correção. Para estender a validação dos modelos, foram definidas regras por meio da linguagem EVL (*Epsilon Validation Language*)⁴, permitindo incrementar as regras sintáticas definidas pelo metamodelo com restrições adicionais, tais como impedir o autorreferenciamento para certos tipos de elementos.

O Código 1 apresenta trechos do código do modelo do cenário, construído por meio da ferramenta de modelagem. Esse código pode ser utilizado, por exemplo, como base para a geração automática de código em alguma linguagem de programação ou ainda como base para sua transformação em outro modelo, este por sua vez baseado em metamodelo próprio. A Figura 4 ilustra a representação gráfica do cenário, tal qual este é modelado na ferramenta. Os rótulos em **negrito** representam os nomes dos elementos na forma de instâncias de tipos do metamodelo. Os rótulos em *itálico* representam as associações entre esses elementos.

Código 1. Instância do metamodelo de cenários de computação ubíqua, representando a configuração do cenário (trechos).

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <ssmm:SmartSpaceDiagram xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI" xmlns:xsi="http
  ://www.w3.org/2001/XMLSchema-instance" xmlns:ssmm="http://www.example.org/ssmm">
3   <SmartSpace xsi:type="ssmm:PersonalSmartSpace" hasOwner="//@Person.0" name="pssP1"/>
4   <SmartSpace xsi:type="ssmm:PersonalSmartSpace" hasOwner="//@Person.1" name="pssP2"/>
5   <SmartSpace xsi:type="ssmm:FixedSmartSpace" name="house"/>
6   /* suprimido */

```

⁴<https://www.eclipse.org/epsilon/doc/evl/>

```

7 <SmartSpace xsi:type="ssmm:FixedSmartSpace" isComposedOf="//@SmartObject.4 //@SmartObject.5
  //@SmartObject.8 //@SmartObject.9" name="bathRoom"/>
8 <UbiquitousApplication name="personalHealth"/>
9 /* suprimido */
10 <SmartObject name="smartPhone"/>
11 <SmartObject name="ecg"/>
12 /* suprimido */
13 <Person uses="//@SmartObject.6 //@SmartObject.4" isOwnerOf="//@SmartObject.0 //@SmartObject
  .1 //@SmartObject.2" hasPSS="//@SmartSpace.0" name="patient1"/>
14 <Person uses="//@SmartObject.6 //@SmartObject.4" isOwnerOf="//@SmartObject.0 //@SmartObject
  .3 //@SmartObject.2" hasPSS="//@SmartSpace.1" name="patient2"/>
15 <Hosts source="//@SmartSpace.0" target="//@UbiquitousApplication.0"/>
16 /* suprimido */
17 <Hosts source="//@SmartSpace.2" target="//@UbiquitousApplication.1"/>
18 <HasSubFSS source="//@SmartSpace.2" target="//@SmartSpace.5"/>
19 <HasSubFSS source="//@SmartSpace.2" target="//@SmartSpace.4"/>
20 <HasSubFSS source="//@SmartSpace.2" target="//@SmartSpace.3"/>
21 </ssmm:SmartSpaceDiagram>

```

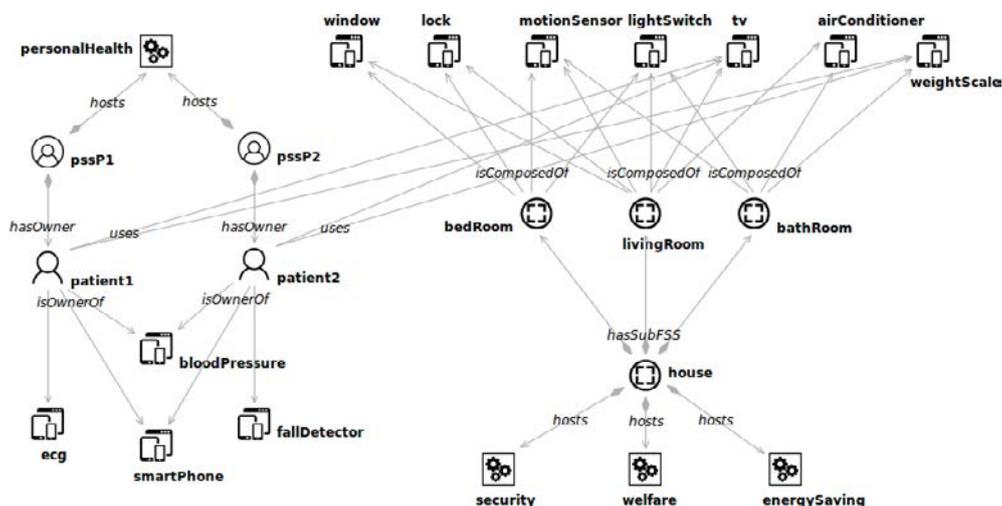


Figura 4. Modelagem do cenário utilizando a ferramenta de modelagem.

Na Figura 4, o usuário identificado por *patient1*, possui configurados em seu espaço inteligente pessoal um sensor de eletrocardiograma (*ecg*), um monitor de pressão arterial (*bloodPressure*) e um aparelho *smartphone*, todos do tipo *SmartObject* e definidos como de uso privado, conforme indicado pela associação *isOwnerOf*. Além disso, o usuário eventualmente utiliza uma balança (*weightScale*) para medir seu peso, a qual está localizada no espaço inteligente fixo *bathRoom* e é compartilhada pelo usuário *patient2*. Um televisor (*tv*), localizado no espaço inteligente fixo *livingRoom*, é utilizado para exibir as informações e alertas enviados pela aplicação ubíqua de saúde pessoal (*personalHealth*). Essas duas últimas associações são do tipo *uses*.

A configuração do espaço inteligente pessoal do usuário *patient2* é composta também por um *smartphone* e um monitor de pressão arterial, além de um sensor de quedas (*fallDetector*). O usuário ocasionalmente confere seu peso na balança compartilhada pelo casal. Semelhante ao que acontece com *patient1*, sua aplicação ubíqua de saúde pessoal utiliza a *SmartTV* da sala de estar para exibir informações e alertas.

Por fim, as aplicações ubíquas de segurança (*security*), de bem-estar (*welfare*) e de economia de energia (*energySaving*) são hospedadas por um espaço inteligente fixo (*house*), que compreende todos os cômodos da casa, abrangendo os espaços inteligentes fixos da sala de estar (*livingRoom*), banheiro (*bathRoom*) e quarto do casal (*bedRoom*).

4. Trabalhos Relacionados

O trabalho de Corredor *et al.* [Corredor et al. 2012] propõe uma metodologia chamada de *Resource-Oriented and Ontology-Driven Development* (ROOD), que disponibiliza um conjunto de ferramentas de modelagem e se baseia nos conceitos de ontologias e arquitetura dirigida a modelos (do inglês, *Model-Driven Engineering* - MDA) para apoiar o desenvolvimento de espaços inteligentes.

Em [Freitas et al. 2014] é apresentada a linguagem 2SML, que permite modelar espaços inteligentes em alto nível, usando a abordagem de modelos em tempo de execução. O 2SML divide a programação do espaço inteligente em dois modelos: modelo do engenheiro e modelo do usuário. Cada um desses modelos possui sua própria linguagem de modelagem, com sintaxes diferentes. Os modelos gerados devem ser processados por uma máquina própria, chamada 2SVM, que possui arquitetura em camadas, inspirada pela CVM (*Communication Virtual Machine*) [Deng et al. 2006].

O projeto PERSIST (*Personal Self Improving Smart Spaces*) [Dolinar et al. 2008] introduz a visão de que os espaços inteligentes pessoais fornecem uma interface entre o usuário e os vários serviços e objetos inteligentes que estão disponíveis, interagindo com outros espaços inteligentes pessoais para criar um ambiente poderoso e flexível.

A nossa proposta difere dos trabalhos apresentados, pois considera que cada pessoa pode formar seu próprio *espaço inteligente pessoal*, e que em determinados momentos pode haver a sobreposição deste com o espaço inteligente configurado em outra aplicação ubíqua (*e.g.*, segurança, bem-estar, economia de energia). O metamodelo para cenários de computação ubíqua proposto apresenta formas de modelar espaços inteligentes, os quais hospedam aplicações ubíquas capazes de lidar com estes fatores.

5. Conclusão

Neste trabalho foram utilizados conceitos de Engenharia Dirigida a Modelos para propor um metamodelo que permite a modelagem em alto nível de um conjunto de espaços inteligentes pessoais e fixos e suas entidades constituintes, aqui denominado de cenário de computação ubíqua. O metamodelo possibilita modelar as formas de interação entre os usuários, representados por seus espaços inteligentes pessoais, e as aplicações ubíquas, objetos inteligentes e demais espaços inteligentes pessoais ou fixos, por meio de uma notação gráfica. Os modelos gerados podem ser utilizados para se ter uma visão geral em alto nível do cenário de computação ubíqua e identificar possíveis inconsistências antes de iniciar o processo de codificação das aplicações ubíquas. Por essa ótica, esses modelos gráficos agem também como forma de documentar o cenário de computação ubíqua.

A validação da proposta se deu por meio da modelagem de um cenário de computação ubíqua, na forma de uma instância dos metamodelo para modelagem de cenários de computação ubíqua proposto, utilizando uma ferramenta de modelagem gráfica desenvolvida para construir modelos em conformidade com esse metamodelo.

O principal trabalho futuro consiste na implementação de transformações M2T (*model-to-text*) na ferramenta de modelagem para permitir a geração de código em nível M0 a partir dos modelos nela construídos. A modelagem e tratamento das políticas de adaptação das aplicações ubíquas com base em informações de contexto obtidas do ambiente também constituem trabalhos futuros.

Agradecimentos

Os autores agradecem ao Instituto Federal de Educação, Ciência e Tecnologia Goiano (IF Goiano) pelo apoio financeiro recebido.

Referências

- Carvalho, S. T., Copetti, A., and Loques Filho, O. G. (2011). Sistema de computação ubíqua na assistência domiciliar à saúde. *Journal Of Health Informatics*, 3(2).
- Carvalho, S. T., Murta, L., and Loques, O. (2012). Variabilities as first-class elements in product line architectures of homecare systems. In *Software Engineering in Health Care (SEHC), 2012 4th International Workshop on*, pages 33–39.
- Corredor, I., Bernardos, A. M., Iglesias, J., and Casar, J. R. (2012). Model-driven methodology for rapid deployment of smart spaces based on resource-oriented architectures. *Sensors*, 12(7):9286–9335.
- Deng, Y., Sadjadi, S. M., Clarke, P. J., Zhang, C., Hristidis, V., Rangaswami, R., and Prabakar, N. (2006). A communication virtual machine. In *Computer Software and Applications Conference, 2006. COMPSAC'06. 30th Annual International*, volume 1, pages 521–531. IEEE.
- Dey, A. K., Salber, D., Abowd, G. D., and Futakawa, M. (1999). The conference assistant: Combining context-awareness with wearable computing. In *The Third International Symposium on Wearable Computers*, pages 21–28. IEEE.
- Dolinar, K., Porekar, J., McKitterick, D., Roussaki, I., Kalatzis, N., Liampotis, N., Papaioannou, I., Papadopoulou, E., Burney, S. M., Frank, K., Hayden, P., and Walsh, A. (2008). PERSIST Deliverable D3.1: Detailed Design for Personal Smart Spaces. <http://www.ict-persist.eu/?q=content/persist-deliverables-and-publications>. [Acessado em Abril-2015].
- Freitas, L. A., Costa, F. M., Rocha, R. C., and Allen, A. (2014). An architecture for a smart spaces virtual machine. In *Proceedings of the 9th Workshop on Middleware for Next Generation Internet Computing*, page 7. ACM.
- López-Fernández, J. J., Cuadrado, J. S., Guerra, E., and de Lara, J. (2015). Example-driven meta-model development. *Software & Systems Modeling*, 14(4):1323–1347.
- Schmidt, D. C. (2006). Guest editor’s introduction: Model-driven engineering. *Computer*, 39(2):0025–31.
- Veiga, E. F., Melo e Maranhão, G., and Bulcão Neto, R. F. (2014). Apoio ao Desenvolvimento de Aplicações de Tempo Real Sensíveis a Contexto Semântico. In *IX Workshop de Teses e Dissertações do XX Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia)*, pages 1–4, João Pessoa-PB.
- Vieira, M. A. and Carvalho, S. T. (2015). Configuração de Espaços Inteligentes para Sistemas Ubíquos de Monitoramento de Pacientes Domiciliares. In *Anais da III Escola Regional de Informática de Goiás (ERI-GO 2015)*, pages 19–30, Goiânia-GO, Brazil. Sociedade Brasileira de Computação (SBC).
- Weiser, M. (1991). The computer for the 21st century. *Scientific american*, 265(3):94–104.