

Uma extensão da UML para modelagem de contexto

Vivian Pedó, Cristiano André Costa, Kleinner Farias

Programa de Pós-Graduação em Computação Aplicada
Universidade do Vale do Rio dos Sinos (UNISINOS)

vivian@framers.com.br, {cac, kleinnerfarias}@unisinós.br

Abstract. *One of the main problems of the UML is the inaccuracy to represent the concept of context, in addition to being rigid to incorporate new models. The same way, the current modeling tools do not support the validation and representation of context concepts. In this scenario, the paper proposes an extension of UML for modeling context, which aims not only to allow the representation and validation of the main aspects of context, but also develop a domain-specific modeling environment. Thus, it is expected to enable a more appropriate support for the context modeling, assisting in the development of context-aware systems.*

Resumo. *Um dos principais problemas da UML é a imprecisão para representar o conceito de contexto, além de ser rígida para incorporar novos modelos. Da mesma forma, as ferramentas de modelagem atuais não dão suporte à representação e à validação de modelos de contexto. Nesse âmbito, o trabalho propõe uma extensão da UML para modelagem de contexto, a qual visa não só permitir a representação e validação dos principais aspectos de contexto, como também desenvolver um ambiente de modelagem específico de domínio. Com isso, espera-se permitir uma modelagem mais adequada para o contexto, auxiliando no desenvolvimento de sistemas cientes de contexto.*

1. Introdução

Sistemas de informação móveis e ubíquos vêm ganhando atenção significativa em várias disciplinas, devido ao rápido desenvolvimento de uma ampla gama de tecnologias móveis e sem fio [Weiser 1991]. Estas aplicações em computação móvel e ubíqua devem estar cientes de seus contextos de interesse e se adaptar às mudanças neles. Entretanto o desenvolvimento de aplicações que interpretem o ambiente, se adaptem ao contexto e permaneçam funcionando enquanto o usuário se locomove é um dos grandes desafios de pesquisa na área [Costa et al. 2008] e [Nadoveza 2014]. Neste cenário de desafios, foi abordado a forma de expressar contexto. Um sistema ciente de contexto requer um modelo de apropriado para representar e manipular informações adquiridas do ambiente [Dey 2001]. Para tentar captar as necessidades desse conceito utilizam-se tecnologias para suportar o processamento semântico, como ontologias que são um instrumento promissor para especificar conceitos e inter-relações, e são adequados para apoiar os serviços de raciocínio no âmbito das aplicações ubíquas [Costa et al. 2008] e [Gu et al. 2004] [Strang e Linnhoff 2004]. Sendo assim, busca-se nas ontologias o vocabulário específico para a compreensão do contexto.

Visando facilitar o entendimento e o trabalho dos desenvolvedores sobre o contexto, busca-se efetuar a modelagem através da linguagem UML¹, visto que a mesma é familiar aos desenvolvedores de software [Booch 2000]. No entanto, a incapacidade da UML em representar os conceitos de ciência de contexto ainda é um desafio. No entanto, a UML proporciona mecanismos de extensão para acomodar tais novos conceitos [Farias et al. 2009]. Este trabalho, portanto, propõe uma extensão do metamodelo da UML para dar suporte a modelagem de contexto. Nesse âmbito, esse artigo propõe a UML2Context, uma linguagem para modelagem de contexto baseado na UML. Além disso, é proposto um ambiente específico de domínio para tornar possível a modelagem de sistemas cientes de contexto utilizando a linguagem proposta.

O artigo é organizado da seguinte forma. A seção 2 descreve os principais conceitos sobre modelagem de contexto. A seção 3 apresenta a extensão da UML proposta, descrevendo como o metamodelo da UML foi estendido. A seção 4 introduz o ambiente de modelagem específico de domínio para dar suporte à modelagem de contexto. A seção 5 contrasta o trabalho proposto com a literatura atual. Por final, a seção 6 apresenta algumas conclusões sobre o trabalho desenvolvido.

2. Modelagem de Contexto

A utilização do contexto vem evoluindo a cada ano paralelamente ao desenvolvimento de sistemas cientes de contexto [Nadoveza 2014]. Uma aplicação é dita ciente de contexto quando utiliza informações de contexto a fim de fornecer serviço ou informação relevantes ao usuário de acordo com a ação que será executada [Dey 2001]. No artigo de Lopes et al. (2013), o contexto é definido como qualquer informação que pode ser usada para caracterizar a situação de uma entidade. De acordo com [Dey e Abowd 2000], uma entidade é uma pessoa, lugar ou objeto considerados relevantes para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a aplicação.

Baseado nos principais conceitos de contexto, o trabalho buscou estudar modelos de contexto uniformes que possuam os termos mais utilizados em sua composição. Partindo da classificação de [Strang e Linnhoff 2004], que organiza seis classificações de abordagem de contexto. Também foi empregado o modelo gráfico da UML (*Unified Modeling Language*) com seus diagramas, que pela sua estrutura genérica, pode ser adaptado para modelar o contexto [UMLBase 2014]. Foca-se assim na modelagem do contexto a partir da UML para auxiliar no desenvolvimento de aplicações cientes de contexto.

2.1. Definições de Termos para Modelagem

A informação do contexto pode ser obtida em uma grande variedade de formas, sendo assim, são necessários meios eficientes e eficazes de modelagem da informação [Nadoveza 2014]. Para auxiliar na modelagem de sistemas cientes de contexto primeiramente é necessário identificar os principais termos utilizados para definir contexto. Então utilizou-se uma base ontológica em alto nível para guiar a montagem do metamodelo, pois as ontologias são um instrumento promissor para especificar conceitos e inter-relações [Strang e Linnhoff 2004]. Ontologia de alto nível ou infraestruturas de ontologia superior são definidas como um conjunto genérico de

¹ <http://www.uml.org>

classes e as relações de propriedades que podem ser usadas em diferentes aplicações e domínios [Nadoveza e Kiritsis 2014 e 2013].

No artigo [Dey e Abowd 2000] foram explanados os elementos de contexto onde a ontologia superior define o conceito da entidade contexto e os conceitos básicos de pessoa, localização, entidade computacional e atividade. No artigo [Bulcão e Pimentel 2005] o modelo de contexto representa os conceitos básicos de atores, localização, tempo, atividades e dispositivos, bem como as relações entre estes conceitos. Leva-se também em consideração as cinco dimensões semânticas chamada de W4H [Bulcão e Pimentel 2005]. Essa abordagem engloba contextos a serem considerados: identidade (quem), localização (onde), tempo (quando), atividade (o que) e perfis de dispositivos (como). Chega-se assim a um modelo de contexto semântico independente de domínio com ontologias de alto nível.

De acordo com os principais pontos levantados nas ontologias analisadas é incluído no modelo um vocabulário capaz de compreender diferentes situações de contexto. O vocabulário foi definido focado nos itens que se repetem nos modelos ontológicos analisados e que são genéricos podendo ser utilizado amplamente. São eles: o próprio contexto, lugar, usuário, dispositivo, atividade, localização e tempo.

3. A Linguagem UML2Context

Definindo a notação que é a parte gráfica, precisamos então definir a sintaxe da linguagem de modelagem, onde estabelecemos o conceito de contexto que será modelado com os elementos que o compõem. Utilizando os mecanismos de extensibilidade são definidos os estereótipos, que estendem o vocabulário UML, permitindo a criação de novos tipos de blocos derivados dos já existentes [Fowler e Scotts 2000]. Esse novos tipos de blocos serão empregados especificamente para a definição de contexto.

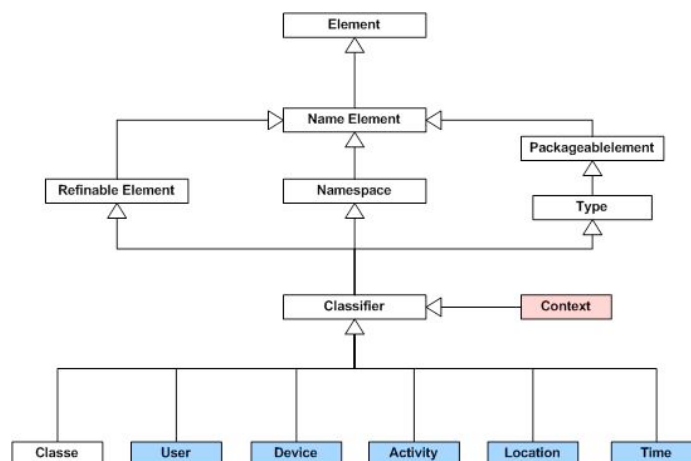


Figura 1. Novas metaclasses inseridas

3.1. Estereótipos e Metaclasses

Os conceitos definidos então são: *Context*, *Location*, *User*, *Device*, *Activity* e *Time*. Entendemos que estes são conceitos-chave para o contexto no desenvolvimento de sistemas ciente de contexto, merecendo ser conceitos de primeira classe no modelo UML2Context. Assim, para cada conceito foi introduzido uma metaclasses, ou seja,

pode-se dizer que cada elemento foi estereotipado como uma metaclassa. A Figura 1 ilustra as metaclasses criadas para representar os conceitos definidos, e que estão destacadas na cor azul. Elas foram inseridas no metamodelo padrão da UML criando nosso próprio metamodelo. As metaclasses definidas possuem o objetivo de especificar o contexto que é uma metaclassa mais ampla que engloba outros elementos em sua composição.

3.2. Responsabilidades

Cada elemento definido como uma metaclassa possui *responsabilidades* dentro do metamodelo, que são obrigações de uma determinada classe. As responsabilidades dos elementos são:

- *User*: é responsável pelo conhecimento acerca das informações sobre o usuário final que está acessando o sistema e suas características relevantes;
- *Device*: é responsável pelo conhecimento acerca do equipamento que está sendo usado como telefone, computador e *gadgets*;
- *Activity*: é responsável pelo conhecimento acerca da ação ou evento que está sendo efetuado, como uma compra, uma consulta, uma dica;
- *Location*: é responsável pelo conhecimento acerca do local físico onde está ocorrendo as atividades ou ações do usuário e da localização do software, localização geográfica, ambiente, setor;
- *Time*: é responsável pelas informações de tempo, data e hora que ocorreu o evento, duração e outros;
- *Context*: é responsável por englobar e tratar as informações das metaclasses anteriores.

3.3. Atributos

Estes elementos também possuem informações importantes que serão consideradas pelo contexto. Essas informações são os *atributos* da classe que também contém operações próprias, são eles:

- *User* - Atributos: nome, idade, sexo, estado civil, quantidade de filhos, endereço, cidade, estado, país, profissão, religião;
- *Device* - Atributos: aparelho, modelo, sistema operacional, operadora;
- *Activity* - Atributos: ação, informação fornecida;
- *Location* - Atributos: posição geográfica, data, hora, temperatura;
- *Time* - Atributos: hora inicial, hora final, duração.

3.4. Operações

As metaclasses vão receber as informações coletadas do ambiente e vão informar a metaclassa contexto essas informações. Sendo assim é proposto os seguintes métodos para cada metaclassa:

- Ler: coleta e lê um valor de um atributo;
- Modificar: acrescenta o valor no campo, alimentando os atributos;

- Informar: informa o valor do atributo quando consultado.

3.5. Relacionamentos

Cada elemento do contexto conterà um relacionamento da metaclassa contexto com as outras metaclasses. E as associações já impõem restrições quando definimos nas mesmas os seus limites [UMLBase 2014]. Sendo assim seguem os relacionamentos propostos e suas restrições:

- Relação *Context-User*: associação da metaclassa *Context* com a metaclassa *User*; Restrições: cada Contexto terá um ou mais Usuários.
- Relação *Context-Location*: associação da metaclassa *Context* com a metaclassa *Location*; Restrições: cada Contexto terá uma Localização.
- Relação *Context-Device*: associação da metaclassa *Context* com a metaclassa *Device*; Restrições: cada Contexto terá um ou mais Dispositivos;
- Relação *Context-Activity*: associação da metaclassa *Context* com a metaclassa *Activity*; Restrições: cada Contexto terá uma ou mais Atividades.
- Relação *Context-Time*: associação da metaclassa *Context* com a metaclassa *Time*. Restrições: cada Contexto terá um Tempo.

3.6. Extensão da UML

Com a notação definida foi estabelecido o metamodelo, o diagrama que define a notação [UMLBase 2014]. Na Figura 2 é possível visualizar o metamodelo proposto com as metaclasses inseridas. Toda a informação que descreve o contexto de usuário é representada pelos estereótipos *ComposableContext*. O elemento *ComposableContext* é uma classe abstrata e subclasse do elemento nomeado (*Namespace* da UML), que é um elemento central no metamodelo UML. Estes elementos determinam a semântica dos elementos do modelo, que podem ser utilizados para representar contexto.

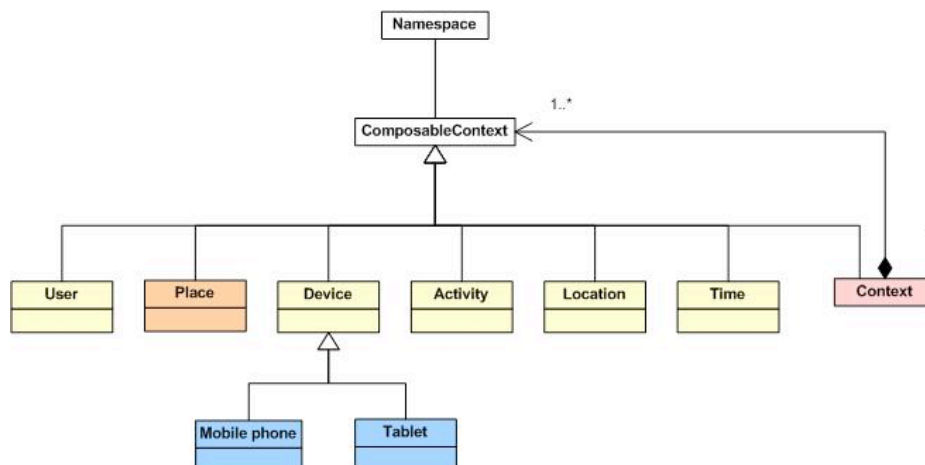


Figure 2. Metamodelo com novos elementos

Estes elementos fazem parte da definição de contexto *ComposableContext*, que é um *Namespace* da UML. A direita do diagrama é possível ver o elemento *Context* no metamodelo, ele é composto pelas entidades que estão no mesmo nível que ele e pode conter também ele mesmo, característica que pode futuramente ser trabalhada para incluir um controle de trilhas de contexto.

Este modelo também serve para que seja possível ampliá-lo uma vez que dependendo da aplicação poder haver necessidade de incluir outros elementos relevantes ao contexto. Assim o modelo pode ser estendido incluindo novos elementos no primeiro nível. Pode-se ver na Figura 2 o exemplo da inclusão do elemento *Place* no metamodelo proposto, onde pode ser adicionado novas informações sobre o Lugar conforme necessidade. O metamodelo proposto também pode incluir a generalização, que são relacionamentos entre elementos gerais chamados superclasses ou classes de primeiro nível e tipos mais específicos desses elementos chamados subclasses ou classes-filhas. Pode-se por exemplo incluir o *Device* que pode ser especializado em outros elementos que serão derivados da mesma classe como *Mobile Phone* e *Tablet*.

4. A Ferramenta UML2Context

Para avaliação do modelo, foi desenvolvida uma ferramenta que permite a modelagem de contexto. O UML2Context *tool* trata-se de um ambiente específico de domínio que atenderá a modelagem de contexto de acordo com a especificação do metamodelo definido. A ferramenta funciona como um plugin do Eclipse² e foi criada utilizando os seguintes plugins:

- EMF (*Eclipse Metamodel Framework*): um framework de modelagem e de geração de código para construir aplicações baseadas em modelos;
- EMOF (*Essencial Meta-Object Facility*): linguagem de definição de metamodelo usada para definições da UML que é parte integrante do *plug-in* EMF;
- GMF (*Graphical Modeling Framework*): *plug-in* do eclipse que fornece um componente generativo e uma infraestrutura *runtime* para desenvolver editores gráficos baseado no EMF e GEF;
- GEF (*Graphical Eclipse Framework*): fornece tecnologia para criar ricos editores gráficos no Eclipse.

Na Figura 3 é possível ter uma ideia da interface do UML2Context com seus componentes de trabalho para modelagem de um simples diagrama. A ferramenta UML2Context é um ambiente de modelagem de contexto com base nos conceitos de linguagem UML2Context, e tem os seguintes componentes de trabalho:

- (A) Package Explorer: Possui o objetivo de permitir a organização de arquivos em uma estrutura de árvore a fim de que seja possível um melhor gerenciamento e manipulação destes;
- (B) Modeling View: Possui o objetivo de permitir aos desenvolvedores visualizar e editar os modelos de forma interativa. Pois os modelos que são criados precisam ser visualizados para atender dois requisitos básicos de modelos que são compreensibilidade e a comunicação;
- (C) Nodes Palette: Possui o objetivo de permitir aos desenvolvedores criar instâncias de construtores, que podem fazer parte do diagrama de classes proposto pelo modelo;

² <http://eclipse.org>

- (D) Relationship Palette: Possui o objetivo de permitir a visualização dos relacionamentos dos construtores, que são visualizados na Modeling View e são instanciados na Nodes *Palette*;
- (E) Properties View: Possui o objetivo de permitir a manipulação de forma precisa das propriedades dos modelos. Ela vai exibir as propriedades definidas no metamodelo;
- (F) Problems View: Possui o objetivo de disponibilizar uma funcionalidade de “*validar modelo*”, que valida os modelos criados em relação ao metamodelo da linguagem. Se existir alguma inconsistência ou item de atenção, ela será mostrada nesse componente. Assim é possível especificar se existe alguma inconsistência no modelo criado em relação à definição do metamodelo da linguagem.

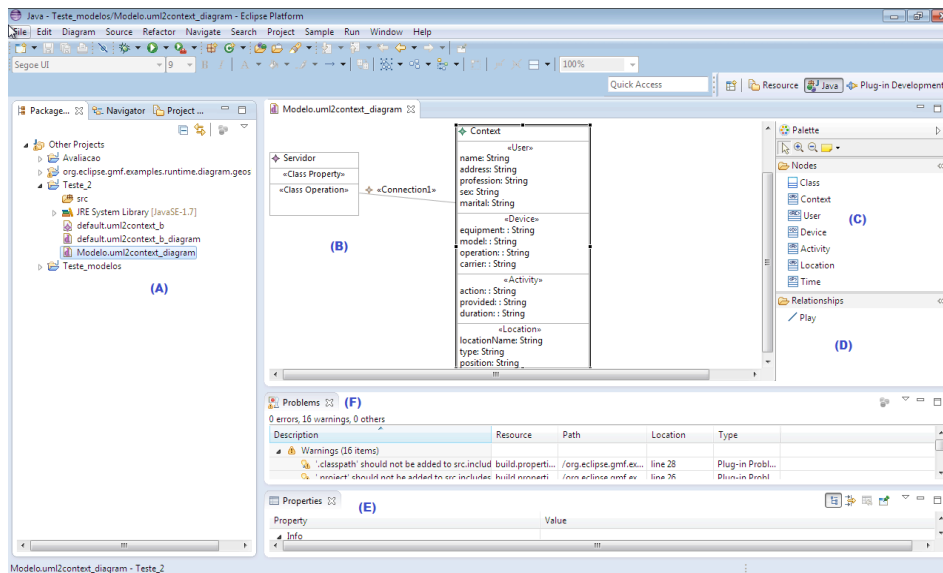


Figura 3. Interface de trabalho do UML2Context

Uma preocupação constante durante o desenvolvimento do metamodelo contexto, juntamente com a extensão da UML é a identificação e representação das características dos conceitos presentes no paradigma de computação ciente de contexto no contexto de metaclasses (estas características são incluídas como propriedades de metaclasses).

4.1. Exemplo de uso da Ferramenta Proposta

O metamodelo define a semântica de como os elementos do modelo se instanciam em um modelo e também pode ser exemplificado por uma metaclassa. Um exemplo de como pode-se fazer uso desse metamodelo em aplicações cientes de contexto é mostrado na Figura 4, onde é ilustrada a metaclassa contexto incluindo as outras metaclasses. Foi feita a aplicação da extensão UML para expressar a relação de composição em tempo de modelagem. Esta relação de composição pode ser então resumida em um elemento Context que é composto de metaclasses com suas características próprias. Nesse exemplo escolhemos como principais elementos: *user*, *device*, *activity*, *location* e *time*. Cada um destes elementos possui suas características como *name*, *age*, *type*, etc.

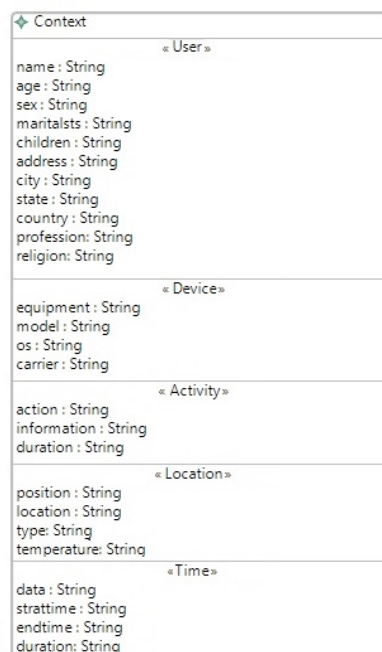


Figure 4. Metaclass Context na UML2Context Tool

5. Trabalhos Relacionados

Alguns artigos pesquisados tiveram abordagens relacionadas, porém com diferentes focos. No Método de Divisão de Sun et al. (2013) foi proposto uma modelagem fatiando o contexto com algoritmos, mas seu objetivo era chegar em uma técnica de análise de modelo, envolvendo a verificação de uma sequência de invocações de operações para descobrir violações de invariantes especificadas reduzindo o tempo de análise. Essa técnica de análise de contexto somente engloba a forma com que o contexto deve ser abordado e quais as variáveis que devem ser consideradas, não chega a efetuar uma extensão UML para comportar graficamente sua modelagem para auxílio direto no desenvolvimento.

Na UML CAProf de Benselim e Seridi-Bouchelaghem (2013) se estendeu a modelagem de contexto através de perfis inseridos por XML no StarUML, onde cada perfil se limita um estereótipo para uma configuração de contexto. Limitando o perfil ao seu estereótipo e estendendo apenas classe e associação.

No CAMEL por Sindico e Grassi (2009) é feita uma extensão de UML para sistemas sensíveis ao contexto em Eclipse utilizando o ECORE, porém a modelagem de sistemas sensíveis ao contexto é limitada por camadas de trabalho separadas em sensoriamento, acionamento e adaptação trabalhando o relacionamento entre elas e não entrando em detalhes sobre o contexto e seus componentes de análise. Foi estendido os itens Operation e TypedElement.

Na linguagem MAS-ML Silva, Choren e Lucena (2008) desenvolveu uma linguagem de modelagem com detalhes sobre os componentes envolvidos e suas relações, estendendo a UML para comportar esses novos objetos e suas relações como agente, organização, papeis, objeto e comportamento. No entanto esta linguagem foi

desenvolvida voltada para sistemas que vão trabalhar com a abordagem de agentes inteligentes.

No UML2Context, se propõe uma extensão da UML para modelagem de contexto para auxiliar no desenvolvimento de sistemas cientes de contexto. Aprofunda-se nos conceitos e nos termos utilizados relevantes para contexto. Procurou-se nas ontologias o melhor vocabulário para utilização não limitando o desenvolvedor na criação de perfis para modelar o contexto e já prevendo a inclusão de variáveis que serão consideradas na modelagem contexto. O modelo proposto não foi limitado por camadas de trabalho e aborda de maneira gráfica a modelagem de contexto facilitando o trabalho dos desenvolvedores. Um trabalho foi feito por Silva, Choren e Lucena (2008) onde foi desenvolvido uma linguagem de modelagem, no entanto focou em agentes inteligentes e no UML2Context foi abordado especificamente o contexto para sistemas cientes de contexto.

6. Conclusão

Esse artigo apresentou a definição de uma linguagem para modelagem de contexto denominada UML2Context, para auxiliar na modelagem de sistemas cientes de contexto. Além disso, como forma de avaliar o modelo, foi desenvolvido uma ferramenta, a UML2Context Tool, que fornece uma perspectiva diferente de trabalho com o contexto. Embora a proposta através da linguagem de modelagem já proporcione uma nova visão e entendimento do uso do contexto nas aplicações cientes de contexto, sem o apoio de uma ferramenta, tornaria o trabalho mais difícil de ser realizado, já que as aplicações padrões que utilizam UML não suportam todos os conceitos levantados e a representação apenas conceitual, sem o uso de gráficos, não seja tão eficiente para os desenvolvedores. Os benefícios obtidos com a linguagem *UML2Context* e a ferramenta *UML2Context Tool* emergem diante da resolução de alguns desafios encontrados até então em ambiente acadêmico e profissional e que seguem descritos.

A principal contribuição científica se dá ao ponto que a linguagem de modelagem de contexto, baseada em ontologias que abrangem muitos dos requisitos para a aplicações cientes de contexto contribuem nos padrão de representação ou representações pré-definidas para o modelo de contexto evitando problemas onde o padrão não é seguido, evitando comprometer a compreensão e o bom entendimento dos modelos. E evitando também ambiguidade e inconsistências tão comuns de ocorrer em projetos [Farias et al., 2009]. A linguagem UML2Context e uma ferramenta para uso dos desenvolvedores onde os modelos terão uma representação gráfica padrão dos elementos que são utilizados no desenvolvimento da aplicação, torna o trabalho de desenvolvimento eficiente e eficaz e os modelos de fácil compreensão. O destaque frente aos trabalhos relacionados ocorre tendo em vista que reúne pontos positivos de cada um deles focando na modelagem de contexto de forma a trazer a mesma para a realidade do desenvolvimento.

Como resultado, pode-se observar que a linguagem UML2Context simplifica a modelagem de contexto já estabelecendo um padrão para modelagem. Os elementos principais estão destacados e com seus atributos definidos facilitando o entendimento dos desenvolvedores, e também permite a extensão de novos elementos. Como trabalho futuro pretende-se avaliar o uso da ferramenta com grupos de desenvolvedores com diferentes graus de experiência.

Agradecimentos

Os autores gostariam de agradecer a CAPES, ao CNPq e a FAPERGS pelo apoio a essa pesquisa.

Referências

- Benselim, M. S. and Seridi-Bouchelaghem, H. (2013) “Extending UML Class Diagram Notation for the Development of Context-aware Applications”, *Journal of Emerging Technologies in Web Intelligence*, Vol. 5, No. 1.
- Booch, G., Rumbaugh, J. and Jacobson, I. (2000). *UML – Guia do Usuário (O mais avançado tutorial sobre Unified Modeling Language (UML) elaborado pelos próprios criadores da linguagem)*. Rio de Janeiro: Campus.
- Bulcão, N. R. F. and Pimentel, M. G. C. (2005) “Toward a Domain-Independent Semantic Model for Context-Aware Computing”, *Proceedings of the Third Latin American Web Congress (LA-WEB’05) Buenos Aires, Argentina* IEEE.
- Costa, C. A., Yamim, A. C. and Geyer, C. F. R. (2008) “Toward a General Software Infrastructure for Ubiquitous Computing”. *Pervasive computing* vol. 7 no.1.
- Dey, A. K. (2001) “Understanding and Using Context”, Springer-Verlag London Ltd *Personal and Ubiquitous Computing*.
- Dey, A. K. and Abowd, G. D. (2000) “The Context Toolkit: Aiding the Development of Context-Aware Applications”, *Proc. 3rd Int. Symposium on Wearable Computers*.
- Farias, K., Nunes, I., Silva, V. and Lucena, C.. *MAS-ML Tool: Um Ambiente de Modelagem de Sistemas Multi-Agentes*. V *Workshop on Software Engineering for Agent-oriented Systems*, 2009.
- Fowler, M. and Scotts, K. (2000). *UML Essencial – Um breve guia para a linguagem-padrão de modelagem de objetos*. São Paulo, segunda edição da Bookman.
- Gu, T., Wang, X. H., Pung, H. K. and Zhang, D. Q. (2004) “An Ontology-based Context Model in Intelligent Environments”, *Proc. Communication Networks and Distributed Systems Modeling and Simulation Conference*, San Diego, CA, USA.
- Lopes, J., Gusmão, M., Duarte, C., Davet, P., Souza, R., Pernas, A., Yamin, A. and Geyer, C. (2013) “Toward a distributed architecture for context awareness in ubiquitous computing”, *Journal Applied Computing Research*, 3(1):19-33.
- Nadoveza, D. and Kiritsis, D. (2014) “Ontology-based approach for context modeling in enterprise applications”, *Elsevier Science - Computer in Industry*.
- Silva, V. T., Choren, R. and Lucena, C. J. P. (2008) “MAS-ML: A Multiagent System Modelling Language”, *Departamento de Sistemas Informáticos Y Computación Universidad Complutense Madrid. Agent-Oriented Software Engineering V.2 No. 4*.
- Sindico, A. and Grassi, V. (2009) “Model driven development of context aware software systems”, *International Workshop on Context-Oriented Programming (COP 2009, July 7, Genova, Italy)*, pp.1-5.
- Strang, T. and Linnhoff-Popien, C. (2004) “A Context Modeling Survey”, *Workshop Proceedings*.
- Sun, W., France, R. B. and Ray, I. (2013) “Contract-Aware Slicing of UML Class Models”, Springer-Verlag Berlin Heidelberg: A. Moreira et al. (Eds.): *MODELS 2013, LNCS 8107*, pp. 724–739.
- UMLBase (2014) “Expand your UML Knowledge and get certification”, <http://umlbase.com/learn/fundamentals/the-uml-metamodel/>, dezembro.
- Weiser, M. (1991) “The Computer for the 21st Century”. *Scientific American Special Issue on Communications, Computers, and Networks*.