

UbiFEX: Modelagem de Características para Linhas de Produtos de Software Sensíveis ao Contexto

Paula Fernandes, Cláudia Werner

PESC/COPPE – Universidade Federal do Rio de Janeiro
Caixa Postal 68.511 – 21945-970 – Rio de Janeiro – RJ – Brasil

{paulacibele,werner}@cos.ufrj.br

***Resumo.** Este artigo apresenta uma ferramenta para modelagem de características para linhas de produtos de software sensíveis ao contexto. UbiFEX utiliza como base a notação de mesmo nome, que permite a representação de forma explícita das entidades e informações de contexto relevantes para o domínio e a influência desse tipo de informação na variabilidade dos produtos. Além disso, uma ferramenta para simulação de contextos foi desenvolvida com o objetivo de verificar em tempo de desenvolvimento a consistência da reconfiguração dinâmica dos produtos.*

1. Introdução

A computação ubíqua [WEISER 1991] compreende um universo formado por uma grande variedade de dispositivos computacionais que estão se tornando cada vez mais presentes no cotidiano das pessoas, como telefones celulares, PDAs, entre outros. Um dos propósitos desse novo paradigma da computação é ampliar as atividades humanas com novos serviços que possam se adaptar às circunstâncias em que serão utilizados [COUTAZ et al. 2005].

Nesse cenário, uma classe de aplicações, chamadas de sensíveis ao contexto (*context-aware*) [BALDAUF et al. 2007], utilizam informações de contexto para fornecer serviços adaptados e relevantes na realização de tarefas dos seus usuários. O desenvolvimento de software para esse tipo de cenário apresenta muitos desafios, entre eles o fato de terem que ser suportados por diferentes dispositivos em diferentes contextos.

O paradigma de Linha de Produtos de Software (LPS) tem como proposta ser um guia para que as organizações possam explorar as características comuns e variáveis dos seus produtos de software para alcançar economia em sua produção [CLEMENTS e NORTHROP 2002]. Um dos benefícios de LPS é o seu alto nível de reúso. Dessa forma, o desenvolvimento de famílias de aplicações sensíveis ao contexto pode se beneficiar desse paradigma em termos de configurabilidade e reusabilidade.

Porém, grande parte das abordagens existentes para LPS trata as variabilidades de cada produto de forma estática, focando apenas em uma adaptação em tempo de desenvolvimento, onde diferentes configurações de um produto são geradas para atender às necessidades dos clientes e às características do ambiente de execução. Essas abordagens não são adequadas para lidar com o dinamismo exigido pelas aplicações sensíveis ao contexto, sendo necessário estendê-las para que sejam capazes de acomodar essas características.

A modelagem de características é uma das primeiras atividades no desenvolvimento de uma linha de produtos. Ela possui como objetivo identificar as características externamente visíveis dos produtos que farão parte da LPS [LEE e MUTHIG 2006], capturando os pontos comuns e variáveis existentes entre eles. Essas características são organizadas em um modelo denominado modelo de características, que é utilizado como ponto de partida para o recorte necessário à instanciação de novos produtos.

Para uma LPS sensível ao contexto é importante que as informações de contexto relevantes para o domínio sejam representadas no modelo de características. Sem essa representação não é possível identificar de forma explícita como esse tipo de informação impacta na seleção de características em tempo de execução.

O objetivo desse artigo é apresentar uma ferramenta para modelagem de características para linhas de produtos de software sensíveis ao contexto. Essa ferramenta utiliza a notação UbiFEX [FERNANDES et al. 2008] e foi desenvolvida no contexto do ambiente Odyssey [ODYSSEY 2008]. A notação UbiFEX estende a notação para modelagem de características do ambiente Odyssey, com o propósito de permitir a representação de informações de contexto e da influência desse tipo de informação na variabilidade dos produtos e decisões de adaptação. Para isso, foram definidos novos tipos de características, regras e expressões que representam essa relação entre características e contextos. Além disso, uma ferramenta para simulação foi desenvolvida com o objetivo de verificar a consistência dos modelos criados com essa notação.

Este artigo está organizado em cinco seções. Na Seção 2, são apresentados os principais conceitos envolvidos neste trabalho. Na Seção 3, é apresentada a notação proposta e o mecanismo de simulação de contextos. Na Seção 4, é apresentado um exemplo de utilização dessa notação. Por fim, na Seção 5, são destacadas as contribuições e trabalhos futuros.

2. Contextualização

2.1. Linha de Produtos de Software

O paradigma de LPS explora os pontos comuns e variáveis em um conjunto de aplicações para um domínio específico. Ao longo do processo de desenvolvimento de uma linha de produtos, aspectos particulares de cada produto podem ser explicitados.

Uma das primeiras atividades no desenvolvimento de uma linha de produtos é a análise de características. O modelo de características representa as características de uma família de sistemas em um domínio, suas semelhanças, diferenças e as relações entre elas. Além disso, é utilizado como ponto de partida para o recorte necessário à instanciação de novos produtos.

Alguns conceitos precisam ser definidos para um melhor entendimento desse tipo de modelo. Pontos de variação são determinados pontos em um sistema de software em que decisões são tomadas, por exemplo, a respeito de qual variante será utilizada. Variantes são alternativas disponíveis para um ponto de variação. Já invariantes compreendem os elementos que não são configuráveis no domínio.

Em relação à opcionalidade, as características podem ser classificadas como: opcionais, quando podem estar ou não presentes em produtos desenvolvidos em uma

linha de produtos; e mandatórias, quando devem obrigatoriamente estar presentes em todos os produtos.

A notação utilizada no ambiente Odyssey é a Odyssey-FEX [OLIVEIRA 2006]. Essa notação busca suprir as deficiências encontradas em outras notações, provendo um alto grau de expressividade na modelagem de características. Ela define a semântica dos elementos que representam conceitos, funcionalidades e tecnologias utilizadas em um domínio, incluindo sua variabilidade, bem como os relacionamentos entre eles. Além disso, são definidas regras de composição que representam as dependências e exclusividades entre esses elementos. Essas regras são chamadas, respectivamente, de inclusivas e exclusivas.

2.2. Sistemas Sensíveis ao Contexto

Sistemas sensíveis ao contexto são capazes de adaptar seu comportamento para um contexto específico sem a intervenção explícita dos usuários. Eles utilizam informações de contexto para prover serviços e informações relevantes.

Muitas definições para contexto podem ser encontradas na literatura. Dey e Abowd (1999) definem contexto como qualquer informação que pode ser usada para caracterizar a situação de uma entidade que é considerada relevante para a interação entre o usuário e a aplicação, incluindo o próprio usuário e a aplicação.

Segundo Coutaz et al. (2005), o contexto fornece uma visão estruturada e unificada do mundo em que o sistema opera e não deve ser visto simplesmente como um estado, mas como parte de um processo. Dessa forma, não é suficiente que o sistema se comporte corretamente em um dado instante, ele deve se comportar assim durante todo o processo em que os usuários estão envolvidos.

3. UbiFEX

A notação UbiFEX estende a notação Odyssey-FEX com o propósito de permitir a modelagem de linhas de produtos sensíveis ao contexto. Para a representação de forma explícita das informações de contexto no modelo de características, UbiFEX introduz dois novos tipos de categorias para a classificação de características: entidade de contexto e informação de contexto.

A característica entidade de contexto tem como objetivo identificar as entidades de contexto relevantes para o domínio. Essas entidades podem ser representadas por lugares (e.g. sala), pessoas ou objetos (e.g. dispositivo móvel). Essa característica possui como propriedades um nome e uma descrição.

Por sua vez, a característica informação de contexto representa as informações que devem ser coletadas do ambiente para descrever as entidades de contexto identificadas. Ela tem como propriedades: nome, descrição, tipo (estática ou dinâmica) e tipo base do valor que ela representa (e.g. string, inteiro).

Porém, apenas a criação dessas novas categorias não é suficiente para representar o impacto dessas informações em relação à variabilidade dos produtos e às decisões de adaptação. Para permitir essa representação, novas expressões e regras foram definidas, as quais foram denominadas: definições de contexto e regras de contexto.

As definições de contexto descrevem situações ou contextos relevantes para o domínio. Elas têm como propriedades um nome e uma expressão. A notação BNF que define essa expressão é apresentada na Figura 1. Uma expressão pode ser formada por uma característica informação de contexto (CIC), previamente modelada no modelo de características, um operador relacional e um valor. Além disso, uma expressão pode ser uma composição de expressões por meio do uso de operadores lógicos. Dessa forma, um contexto está no estado ativo para um dado ambiente, quando a avaliação da expressão que o descreve retorna um valor verdadeiro.

```

<definição-contexto> ::= <expressão>
<expressão> ::= <CIC><operador-relacional> <valor>
               | <expressão> <operador-lógico> <expressão>
               | NOT <expressão>
<operador-relacional> ::= > | < | >= | <= | = | <>
<operador-lógico> ::= AND | OR
<valor> ::= <string> | <inteiro> | <float> | <booleano>

```

Figura 1. BNF para as definições de contexto.

Após a definição dos contextos relevantes para o domínio, as regras de contexto podem ser especificadas. Essas regras representam como um contexto, previamente definido, impacta na configuração dos produtos da LPS em tempo de execução, determinando, por exemplo, decisões a respeito da seleção de variantes em um ponto de variação.

Uma regra de contexto é formada por um antecedente, o operador implica (*implies*) e um conseqüente. O antecedente é representado por uma expressão que pode ser formada pela combinação de definições de contexto, características e operadores lógicos. Já o conseqüente é definido por uma expressão que pode ser formada pela combinação de características e operadores lógicos. O operador implica determina que se a expressão que define o antecedente for avaliada como verdadeira, então, as características presentes no conseqüente devem ser ou não selecionadas para a nova configuração do produto, dependendo dos operadores lógicos envolvidos.

As regras de contexto podem ser utilizadas tanto para representar mudanças essenciais na configuração do produto quanto para prover funcionalidades relevantes, com base nos contextos que estão ativos no ambiente de execução de um determinado produto. As características que fazem parte do conseqüente de uma regra de contexto são marcadas no canto superior esquerdo com o identificador da regra, permitindo de forma clara a identificação das características que sofrem influência de contexto.

Todas as expressões utilizadas na representação tanto das definições de contexto quanto das regras de contexto foram implementadas segundo o padrão de projeto Interpreter [GAMMA et al. 1995], permitindo que a avaliação das mesmas seja feita de forma recursiva.

A Figura 2 ilustra o ambiente para modelagem de características do ambiente Odyssey, após as extensões propostas por UbiFEX. Na barra de ferramentas, novos ícones foram adicionados para permitir a criação das features entidade e informação de contexto. Na barra lateral esquerda, foram adicionados novos elementos para a criação

das definições e regras de contexto. A figura também apresenta parte do modelo de características de uma LPS no domínio de guias turísticos móveis.

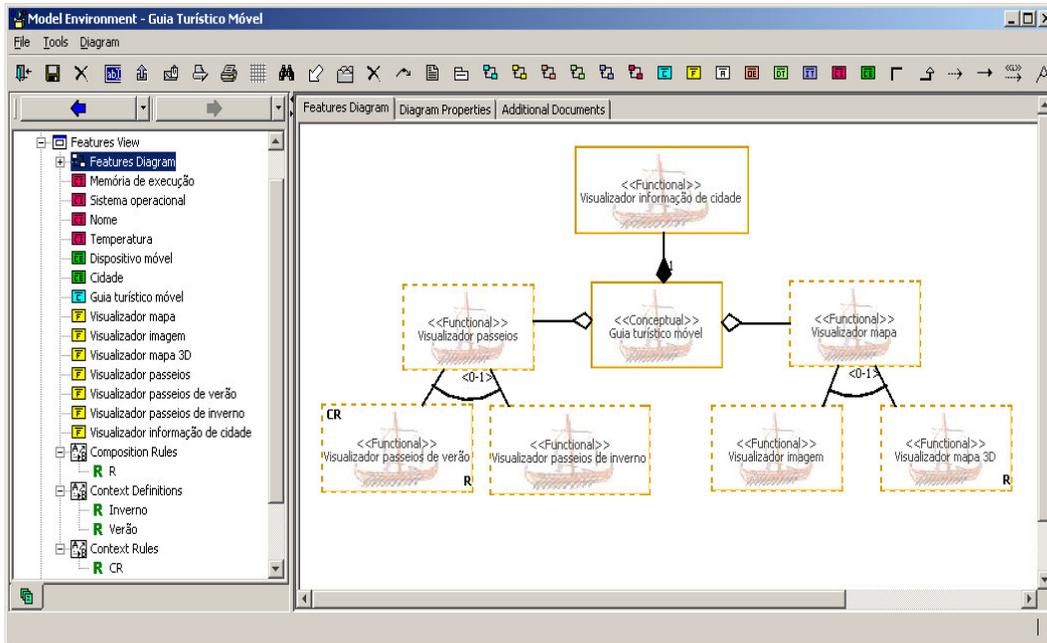


Figura 2. Extensão do ambiente para modelagem de características do Odyssey.

Outra ferramenta foi desenvolvida com o objetivo de verificar a consistência dos modelos criados utilizando a notação UbiFEX. Ela permite que em tempo de desenvolvimento a consistência, tanto inter-regra quanto intra-regra, das regras de contexto seja verificada, além dos conflitos que possam surgir entre essas regras e as regras de composição. Esses conflitos podem ocorrer no caso de uma das regras exigir a inclusão de uma determinada características na configuração do produto enquanto a outra exige a exclusão da mesma. Essa verificação é feita por meio da simulação dos contextos definidos para o domínio. O objetivo é tentar reduzir as falhas que eventualmente ocorreriam na execução do produto. A Figura 3 ilustra as principais etapas desse processo.

Inicialmente, os valores para a simulação precisam ser definidos para cada característica do tipo informação de contexto modelada. Esses valores podem ser definidos de três formas: uma lista de valores; um intervalo; ou de forma automática. Nesse último caso, os valores são calculados com base nas expressões que representam as definições de contexto, incluindo valores que testem tanto o valor verdadeiro quanto falso das expressões.

Após essa etapa, a configuração inicial do produto deve ser selecionada e as regras de composição verificadas para essa configuração. O próximo passo é a simulação dos valores, onde são atribuídos valores específicos para cada característica do tipo informação de contexto com base nos valores definidos na etapa inicial. Em seguida, as definições de contexto são analisadas para identificar os contextos ativos. Com base nesse resultado, as regras de contexto também são analisadas e são identificadas as alterações necessárias na configuração do produto. Finalmente, essas modificações são aplicadas tendo como base a configuração inicial e são verificadas as possíveis inconsistências.

cias dessas modificações em relação às regras de composição. Esse ciclo se repete até que todas as combinações de valores sejam simuladas. Ao final do processo, um relatório com as inconsistências encontradas é gerado.

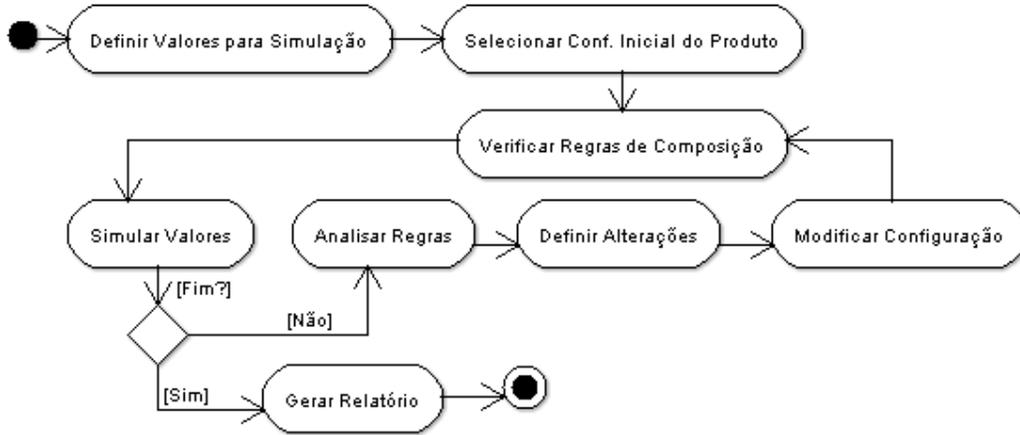


Figura 3. Etapas do processo de simulação.

As ferramentas apresentadas foram desenvolvidas em Java. A ferramenta para modelagem foi implementada no núcleo do ambiente Odyssey, estendendo a notação para modelagem de características previamente existente. Já a ferramenta de simulação de contexto foi implementada na forma de plug-in para o Odyssey, podendo ser carregada por demanda [FERNANDES et al. 2007].

4. Exemplo de utilização

Para ilustrar a utilização das ferramentas, iremos utilizar como exemplo uma LPS para o domínio de guias turísticos móveis. Esses guias servem como apoio para os turistas, provendo informações relevantes sobre pontos turísticos, restaurantes, passeios, levando em consideração, por exemplo, as condições da cidade visitada ou a localização e preferências do usuário.

A Figura 2 descreve parte do modelo de características desse domínio. A característica “Visualizador informação de cidade” é mandatória, enquanto que “Visualizador passeios” e “Visualizador mapas” são opcionais e representam pontos de variação. Para esse domínio, definimos a seguinte regra de composição inclusiva (R): “Visualizador passeios verão” *requires* “Visualizador mapa 3D”.

O modelo de características de contexto pode ser representado no mesmo diagrama da Figura 2, porém para explicitar a sua visualização, optamos por utilizar um novo diagrama (Figura 4). Nesse modelo temos, por exemplo, a entidade de contexto “Cidade”, caracterizada pelas informações de contexto “Nome” e “Temperatura”.

Foram criadas duas definições de contexto, denominadas “Verão” e “Inverno”, que são representadas, respectivamente, pelas expressões (“Cidade.Temperatura” ≥ 25) e (“Cidade.Temperatura” < 25). Dessa forma, se a temperatura da cidade for maior ou igual a 25, dizemos que a definição de contexto “Verão” está ativa.

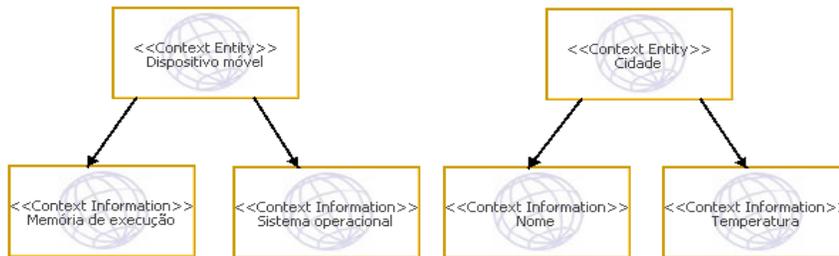


Figura 4. Modelo de características de contexto.

Após a definição dos contextos do domínio, as regras de contexto podem ser criadas. Como exemplo, definimos a seguinte regra de contexto (CR): “Verão” *implies* “Visualizador passeios verão”. Essa regra indica que se a definição de contexto “Verão” estiver ativa, então a decisão a ser tomada para o ponto de variação “Visualizador passeios” deve ser “Visualizador passeios verão”.

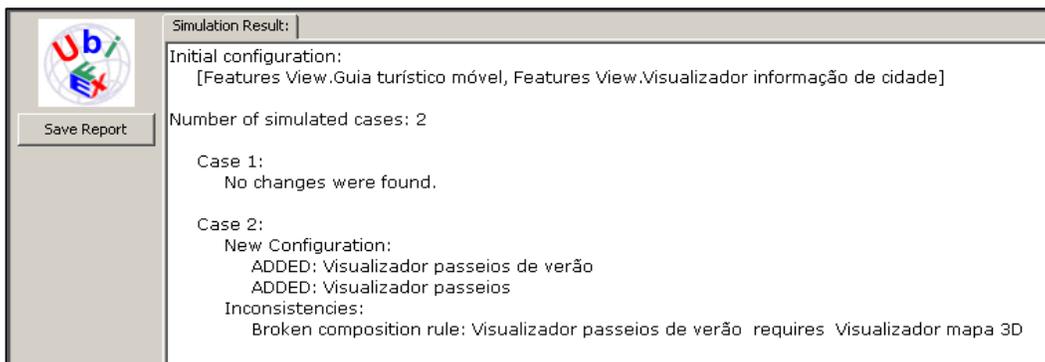


Figura 5. Relatório de simulação.

Para ilustrar a ferramenta de simulação, definimos dois valores para a informação de contexto “Cidade.Temperatura”, um que torna a expressão que representa “Verão” verdadeira (25) e outro que a torna falsa (15). A Figura 5 apresenta o relatório gerado após a execução da simulação. Para o caso onde a definição de contexto não está ativa, nenhuma mudança foi encontrada. No outro caso, a regra de contexto adicionou a variante “Visualizador passeios verão”, porém essa característica requer uma outra característica que não estava presente na configuração final, o que gerou uma inconsistência nessa configuração.

5. Conclusões

As informações de contexto são elementos chave no desenvolvimento de aplicações auto-adaptativas para a computação ubíqua. Na literatura podemos encontrar alguns trabalhos que tratam de LPS dinâmicas [LEE e KANG 2006], porém eles não se preocupam com a forma como essas informações são identificadas e representadas. Dessa forma, UbiFEX permite um melhor entendimento e representação das entidades e informações de contexto relevantes para o domínio. Além disso, a ferramenta de simulação permite a verificação em tempo de desenvolvimento de inconsistências que provavelmente só seriam identificadas durante a execução dos produtos.

Na versão atual, a ferramenta de modelagem permite apenas a representação de informações de contexto simples por limitação da notação UbiFEX. Além disso, a ferramenta de simulação suporta somente tipos inteiros e booleanos.

Como trabalhos futuros, prevemos a avaliação das ferramentas apresentadas em um domínio real de maior escala, como forma de analisar a representatividade da notação proposta e a escalabilidade da ferramenta de simulação.

Agradecimentos

Os autores gostariam de agradecer à CAPES e ao CNPq pelo apoio financeiro.

Referências

- BALDAUF, M., DUSTDAR, S., ROSENBERG, F. (2007) *A Survey on Context-Aware Systems*, Technical Report TUV-1841-2004-24, Information Systems Institute of the Technical University of Vienna.
- CLEMENTS, P., NORTHROP, L. (2002) *Software Product Lines: Practices and Patterns*, Addison-Wesley.
- COUTAZ, J., CROWLEY, J.L., DOBSON, S., et al. (2005) "Context is Key", *Communications of the ACM*, v. 48, n. 3 (March), pp. 49-53.
- DEY, A.K., ABOWD, G.D. (1999) "Towards a Better Understanding of Context and Context-Awareness". In: *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, pp. 304-307, Karlsruhe, Germany, September.
- FERNANDES, P., PRUDÊNCIO, J.G., MARINHO, A., et al. (2007) "Carga Dinâmica de Componentes via Biblioteca Brechó". In: *Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software (SBCARS 2007), Sessão de Ferramentas*, pp. 1-8, Campinas, São Paulo, Brasil, Agosto.
- FERNANDES, P., WERNER, C., MURTA, L. (2008) "Feature Modeling for Context-Aware Software Product Lines". In: *Twentieth International Conference on Software Engineering and Knowledge Engineering (SEKE'08)*, pp. 758-763, Redwood City, California, USA, July.
- GAMMA, E., HELM, R., JOHNSON, R., TESTE, T. (1995) *Padrões de Projeto - Soluções Reutilizáveis de Software Orientado a Objetos*, Ed. Bookman.
- LEE, J., KANG, K. (2006) "A Feature-Oriented Approach to Developing Dynamically Reconfigurable Products in Product Line Engineering". In: *10th International Conference on Software Product Line*, pp. 131-140, Baltimore, Maryland, USA, August.
- LEE, J., MUTHIG, D. (2006) "Feature-Oriented Variability Management in Product Line Engineering", *Communications of the ACM*, v. 49, n.12 (December), pp. 55-59.
- ODYSSEY (2008) "Projeto Odyssey". In: <http://reuse.cos.ufrj.br/odyssey>, acessado em 11/06/2008.
- OLIVEIRA, R.F., 2006, *Formalização e Verificação de Consistência na Representação de Variabilidades*, Dissertação de M.Sc., COPPE, UFRJ, Rio de Janeiro, Brasil.
- WEISER, M. (1991) "The Computer for the 21st Century", *Scientific American*, v. 265, n. 3 (September), pp. 94-104.