Uma Proposta para Geração de uma Arquitetura de Linha de Produtos de Software Dinâmica

Position Paper

Eldânae Nogueira Teixeira, Cláudia M. L. Werner, Paula Fernandes

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

{danny,werner,paulacibele}@cos.ufrj.br

Abstract. *The context-aware system approach is becoming an opportunity to attend the need to access computational environment, anytime and anywhere, in a relevant manner to the user. However, the construction of these systems is not supported by Software Engineering techniques to runtime adaptations based on the system current context. The goal of this work is to achieve an architecture to support the Dynamic Software Product Line approach, allowing a mapping among different abstraction levels.*

Resumo. *A abordagem de sistemas sensíveis ao contexto vem surgindo como uma oportunidade para atender as necessidades de acesso a ambientes computacionais, em qualquer lugar e a qualquer momento, de forma relevante ao usuário. No entanto, a construção desses sistemas enfrenta dificuldades pela falta de suporte da Engenharia de Software a questões de adaptação em tempo de execução de acordo com o contexto corrente do sistema. O objetivo deste trabalho é prover uma arquitetura de suporte a abordagem de Linha de Produtos de Software Dinâmica, viabilizando um mapeamento entre diferentes níveis de abstração.*

1. Introdução

Segundo Weiser (1991), um dos precursores da Computação Ubíqua, o objetivo deste paradigma é possibilitar ao usuário o acesso a ambientes computacionais, em qualquer lugar e a qualquer momento, mas de forma que o usuário não perceba a interação com a máquina, sendo esta autônoma, interativa e relevante. Essa demanda por tecnologias que atendam aos conceitos de ubiquidade promove o crescimento de estudos no campo de sistemas sensíveis ao contexto. Esses sistemas são capazes de adaptar suas operações ao contexto corrente sem intervenção explícita do usuário e, portanto, buscam aumentar sua usabilidade e efetividade [BALDAUF *et al.*, 2007]. Existem na literatura diferentes definições para este tipo de aplicações, podendo destacar a de Dey (2001), que as define como sistemas que usam o contexto para prover informações relevantes e/ou serviços ao usuário, onde a relevância depende da tarefa do usuário.

A falta de um maior suporte da Engenharia de Software gera dificuldades no desenvolvimento desses sistemas, uma tarefa complexa, que envolve diversas questões, como: o tratamento das informações contextuais, da heterogeneidade de equipamentos envolvidos, além de procedimentos para análise e resolução de reconfigurações e falhas em tempo de execução. Esses sistemas podem ser implementados de diferentes formas, que variam de acordo com o propósito do sistema e as suas funcionalidades básicas.

Motivadas por este cenário, encontramos uma variedade de abordagens com foco no suporte do desenvolvimento desses sistemas. O trabalho de Baldauf *et al.* (2007) descreve uma arquitetura conceitual dividida em cinco camadas, que trabalham as questões de detecção e uso do contexto. Outros trabalhos envolvem a definição de uma arquitetura, como o trabalho de Henricksen e Indulska (2006), que propõe um *framework* de suporte ao desenvolvimento e a modelagem de contexto através de uma notação de modelagem gráfica denominada CML (*Context Modeling Language*), e o trabalho de Santos (2008), que define um arquitetura contextual que trata três elementos principais: Fonte de Contexto (*Context Source*), Gerenciador de Contexto (*Context Manager*) e Consumidor de Contexto (*Context Consumer*).

No entanto, grande parte desses estudos são feitos de forma isolada e sem um fórum formal para discussão de suas diversas facetas. Além disso, as soluções propostas são específicas para um determinado problema, não tendo o reúso como foco principal. A reutilização pode ser definida como um processo de criação de sistemas de software a partir de software preexistente [KRUEGER, 1992], e traz como promessa de benefícios o aumento da produtividade e da qualidade, além da redução dos custos e do tempo de desenvolvimento. Os sistemas sensíveis ao contexto podem compartilhar diversas similaridades que poderiam ser tratadas através de abordagens de reutilização, como Linha de Produtos de Software (LPS). Essa abordagem visa efetivar a reutilização de forma sistemática baseada em uma família de sistemas que compartilham um conjunto de características comuns e controladas [NORTHROP, 2002]. Além dos níveis de abstração, o nível de granularidade é um fator importante, que pode tratar a unidade reutilizável como um componente. O paradigma de Desenvolvimento Baseado em Componentes (DBC) trata desse nível, tendo como objetivo, em primeira instância, o desenvolvimento de software a partir de componentes pré-concebidos, que são reutilizáveis em várias aplicações e facilmente personalizáveis para produzir novas funcionalidades [HEINEMAN & COUNCILL, 2001].

Assim, um dos principais problemas é a construção de uma arquitetura que suporte essa dinamicidade de adaptação e a rastreabilidade entre os diferentes níveis de abstração. Essas arquiteturas, quando especificadas através de elementos especializados no domínio e compostas em um padrão de estrutura eficaz para a construção de uma família de aplicações, são definidas como Arquiteturas de Referência de Domínio (DSSA – *Domain Specific Software Architecture*) [HAYES, 1994].

Desta forma, o objetivo deste trabalho consiste em propor uma abordagem de apoio a um projeto arquitetural baseado em componentes dentro de uma LPS Dinâmica (LPSD), que pode ser entendida como uma LPS onde os produtos são sistemas adaptativos (i.e., podem se adaptar pró-ativamente quando mudanças são executadas em seu ambiente) [CETINA *et al.*, 2008]. A proposta visa identificar os elementos arquiteturais que devem fazer parte de uma DSSA nesse domínio de aplicações e apoiar a especificação desta arquitetura através de mecanismos para geração e organização destes componentes derivados a partir de um modelo de características e de um modelo contextual, resultados da fase de análise do domínio. O modelo de características pode ser entendido como um modelo de alto nível de abstração que representa os requisitos dos produtos de uma LPS. Já o modelo contextual representa explicitamente as informações contextuais e as regras de contexto, que designam reconfigurações baseadas no contexto corrente da aplicação, representando possíveis decisões de adaptação.

Este artigo está organizado em três seções. Na seção 2, a abordagem proposta é discutida. Finalmente, a seção 3 apresenta as conclusões.

2 – Abordagem Proposta

O objetivo desta abordagem é definir um arcabouço genérico para o desenvolvimento de aplicações sensíveis ao contexto. Este deverá estar fundamentado na especificação de uma arquitetura de referência, baseada em componentes, para este domínio de aplicações. Para isso, a abordagem deve fornecer mecanismos que auxiliem na criação, especificação e organização dos elementos arquiteturais, derivados a partir de modelos de características e modelos de contexto. Deve-se levar em consideração a definição de heurísticas e a necessidade de manutenção da consistência dos conceitos envolvidos na especificação dos requisitos do domínio. O processo de criação de uma arquitetura de referência requer grande esforço do Engenheiro de Domínio devido a sua generalização, devendo este recorrer a várias fontes de informação [BRAGA, 2000]. Uma avaliação da combinação de estratégias apoiadas em um direcionamento *top-down* (i.e., da análise para o projeto), e estratégias baseadas na extração de conhecimento a partir de arquiteturas existentes ou sistemas legados, deve ser realizada para determinar uma DSSA dentro deste domínio de aplicações.

A partir de estudos preliminares, a abordagem foi decomposta em três etapas: (1) geração da arquitetura conceitual; (2) geração da arquitetura lógica e (3) geração da arquitetura operacional.

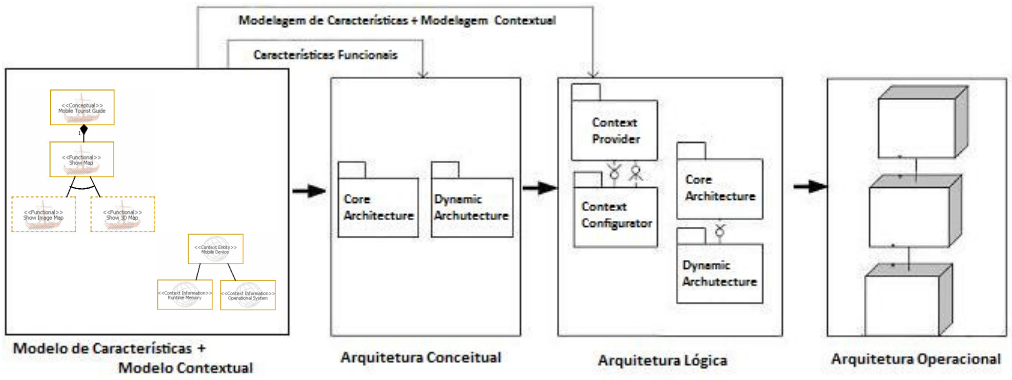

Figura 1 - Etapas do mapeamento

2.1. Geração da Arquitetura Conceitual

Essa etapa é responsável pela identificação dos componentes do sistema, das responsabilidades e dos relacionamentos em termos dos conceitos do domínio de aplicação.

A definição desta arquitetura prevê a necessidade de avaliação do modelo de características e de seu processo de construção. As propostas existentes de realização deste mapeamento são baseadas na premissa que uma característica é diretamente mapeada para um componente e que a modelagem deve ser estruturada de acordo com esta visão. No entanto, o objetivo é avaliar quais as cardinalidades que podem ser trabalhadas, mantendo a rastreabilidade e a identificação das funcionalidades do sistema e suas unidades de implementação.

Esta etapa envolve a definição de uma arquitetura *core* e uma arquitetura dinâmica. A arquitetura *core* corresponde aos componentes que representam as caracterísi-

cas mandatórias, aquelas compartilhadas por todos os produtos derivados da linha de produtos. No entanto, algumas características, ainda que não sejam mandatórias, apresentam alto grau de similaridade e uma avaliação deve determinar a sua inclusão na arquitetura *core*. Trabalhos nesta área estão sendo desenvolvidos [BENAVIDES *et al.* 2005] e devem ser avaliados no desenvolvimento desta abordagem. A arquitetura dinâmica irá trabalhar com as características consideradas não mandatórias. Outro aspecto a ser avaliado é a necessidade de definição de componentes que gerenciam os relacionamentos entre as características e, consequentemente, entre os componentes da arquitetura, como, por exemplo, os relacionamentos entre componentes da arquitetura *core* e da arquitetura dinâmica. Esses "componentes-relacionamento" trabalhariam como intermediários, gerenciando a conexão e as interfaces providas e requeridas, reduzindo o acoplamento entre os componentes. A Figura 2 mostra uma visão inicial da arquitetura para um pequeno fragmento do modelo de características de um domínio de guia turístico móvel [BAUS *et al*., 2005] .

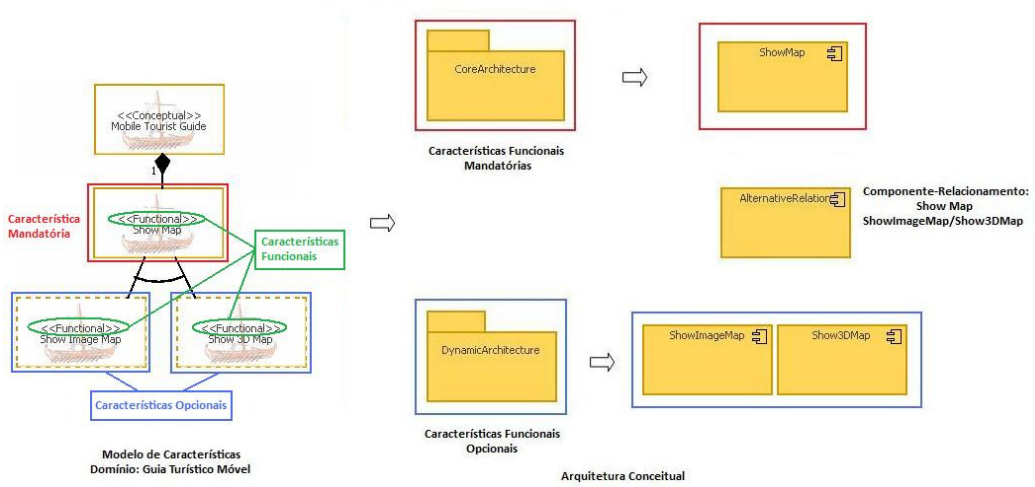

Figura 2 - Exemplo simplificado de Arquitetura Conceitual

2.2. Geração da Arquitetura Lógica

A arquitetura lógica é responsável pelo detalhamento da especificação da arquitetura conceitual, introduzindo mais estruturas lógicas e mecanismos que tratem as questões do contexto implementacional e das características não funcionais. Nesta fase, devem ser consideradas as questões relacionadas à determinação e atribuição das responsabilidades de cada componente através de uma série de interfaces. O modelo de interações entre componentes poderá ser definido através de serviços e eventos, sem referência explícita e direta a um componente em particular, possibilitando o desacoplamento entre os provedores de funcionalidade e consequentemente reduzindo impactos após as reconfigurações.

Os tópicos relacionados ao tratamento do contexto também devem ser avaliados nesta fase. Assim, devem ser incluídos na arquitetura, componentes provedores de contexto e componentes configuradores. Os componentes provedores de contexto devem trabalhar as questões de aquisição, interpretação e disseminação das informações de contexto. Esses componentes estão associados à existência de múltiplas fontes e à necessidade de gerenciamento e provimento das informações relevantes em um nível de abstração apropriado para a aplicação. Com relação aos componentes configuradores, estes devem ter conhecimento de todo o modelo de características e são responsáveis

pela especificação das estratégias de reconfiguração, avaliando o contexto corrente em relação às regras de contexto definidas na modelagem do domínio. Deve-se avaliar a necessidade da existência de um componente que trate as questões de validação e tratamento de impactos, ou se essas responsabilidades serão delegadas individualmente às características envolvidas, direta ou indiretamente no processo de reconfiguração.

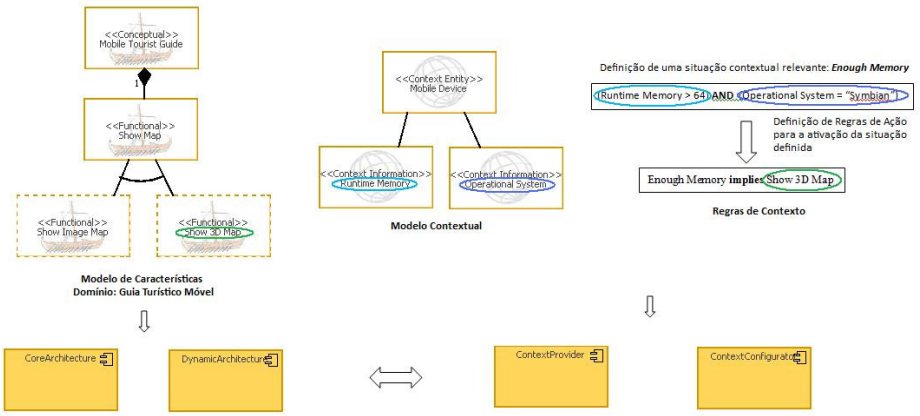

Figura 3 - Exemplo simplificado de Arquitetura Lógica

2.3. Geração da Arquitetura operacional

A arquitetural operacional foca na distribuição das funcionalidades em nós computacionais. Essa distribuição deve estar associada à disposição da estrutura de interação mantida entre os nós, com o objetivo de atender os requisitos expressos pelas características não-funcionais. Representa um mapeamento para unidades computacionais, que podem ser derivadas a partir da definição de unidades de características que representam conjuntos de características relacionadas em um modelo de características para realizar uma funcionalidade em comum.

Nesta fase, é necessário que sejam avaliadas questões de implementação dos componentes definidos, além de plataformas de suporte e middlewares que forneçam serviços, mecanismos e interfaces que implementem as funcionalidades requeridas ou o fornecimento de um arcabouço que viabilize o desenvolvimento das aplicações.

3. Conclusão

Atualmente, a pesquisa encontra-se em um estágio inicial de definição da proposta. Alguns aspectos práticos deverão ser identificados durante o desenvolvimento desta abordagem. Inicialmente, este trabalho está inserido no contexto do ambiente Odyssey (ODYSSEY, 2009), uma infra-estrutura de reutilização baseada em modelos de domínio. Algumas análises devem ser realizadas para avaliar as representações disponíveis no ambiente e sua adequação à proposta como, por exemplo: representações de informações contextuais existentes e avaliação da necessidade de extensão dos conceitos presentes na modelagem definida dentro do ambiente Odyssey pela abordagem UbiFEX [FERNANDES E WERNER, 2008]. Estudos de áreas de arquitetura e desenvolvimento baseado em componentes, além da adequação dessas abordagens à área de computação sensível ao contexto, devem ser conduzidos. A expectativa é o levantamento de novos questionamentos e a possibilidade de mapeamento de requisitos e soluções para a geração da arquitetura proposta.

Agradecimentos

Os autores gostariam de agradecer à CAPES e ao CNPq pelo apoio financeiro, e à equipe de Reutilização da COPPE/UFRJ, que direta ou indiretamente contribuiu para a realização desse trabalho.

Referências

BALDAUF, M., DUSTDAR, S., ROSENBERG, F., 2007, *A Survey on Context-Aware Systems*, Relatório Técnico TUV-1841-2004-24, Information Systems Institute of the Technical University of Vienna.

BAUS, J., CHEVERST, K., AND KRAY, C. 2005. A survey of map-based mobile guides. Map-based mobile services - Theories, Methods, and Implementations, 197-216.

BENAVIDES, D., TRINIDAD, P., RUIZ-CORTÉS, A., 2005, "Automated Reasoning on Feature Models". In: *17th Conference on Advanced Information Systems Engineering (CAiSE'05)*, pp. 491-503, Porto, Portugal, June.

BRAGA, R., 2000, Busca e Recuperação de Componentes em Ambientes de Reutilização de Software, Tese de D.Sc., COPPE, UFRJ, Rio de Janeiro, Brasil.

CETINA, C., PELECHANO, V., TRINIDAD, P., RUIZ-CORTÉS, A., 2008, "An architectural discussion on DSPL". In: 2nd International Workshop on Dynamic Software Product Lines, Limerick, Irlanda, Setembro.

DEY, A., 2001, "Understanding and Using Context", *Personal and Ubiquitous Computing*, v. 5, n. 1 (February), pp. 4-7.

FERNANDES, P., WERNER, C., 2008, "Ubifex: Modeling context aware software product lines". In 2nd InternationalWorkshop on Dynamic Software Product Line Conference, Limerick, Ireland, 2008, pp. 3-8.

HAYES, R., 1994, Architecture-Based Acquisition and Development of Software Guidelines and Recommendations from the ARPA Domain-Specific Software Architecture (DSSA) Program Tecknowledge Federal System.

HEINEMAN, G.T., COUNCILL, W.T., 2001, "Component-Based Software Engineering: Putting the Pieces Together", Addison-Wesley, 2001.

HENRICKSEN, K., INDULSKA, J., 2006, "Developing Context-Aware Pervasive Computing Applications: Models and Approach", Pervasive and Mobile Computing Journal, v.2, n. 1, 2006, pp. 37-64.

KRUEGER, C.W., 1992, "Software Reuse", ACM Computing Surveys, v. 24, n. 2 (June), pp. 131-183.

NORTHROP, L., 2002, "SEI's software product line tenets", IEEE Software, vol. 19, 2002, pp. 32-40.

ODYSSEY, 2009, "Odyssey Project". http://reuse.cos.ufrj.br.

SANTOS, V., 2008, "*CEManTIKA: A Domain-Independent Framework for Designing Context-Sensitive Systems*". Tese de D.Sc., Centro de Informática, UFPE, Pernambuco, Brasil

WEISER, M., 1991, "*The Computer for the 21st Century*". In: Scientific American 265, Nr. 3, S. 94-101.