

7 Conclusão

Neste capítulo são apresentadas as conclusões do trabalho com um resumo das contribuições, discussões acerca de lições aprendidas com o uso da abordagem e uma lista de trabalhos futuros.

Esse trabalho propôs uma abordagem baseada em modelos e anotações de código para derivação automática de produtos de software. Tal abordagem é composta por três modelos (implementação, configuração e características). Estes modelos representam o conhecimento de configuração de uma arquitetura de LPS e possibilitam que produtos sejam automaticamente derivados a partir de um núcleo de artefatos de software. Uma versão inicial desses modelos pode ser criada a partir de anotações marcadas no código-fonte dos elementos de implementação. As anotações definidas permitem a criação automática de características no modelo de características e mapeamentos entre as características criadas e elementos de implementação anotados no modelo de configuração. A abordagem foi exemplificada e validada por meio de dois estudos de caso: (i) o framework JUnit; e (ii) uma linha de produtos de software para jogos J2ME.

O trabalho também descreveu um estudo sobre diversas tecnologias que facilitam o desenvolvimento de ferramentas baseadas em modelos e com geração de código. A plataforma extensível do Eclipse foi brevemente apresentada juntamente com três plug-ins que oferecem diversas funcionalidades para desenvolvimento baseado em modelos: (i) EMF; (ii) oAW; e (iii) FMP. A plataforma Eclipse juntamente com estes plug-ins formam a base de implementação da ferramenta GenArch. Essa escolha trouxe diversos benefícios, tais como: (i) facilidade na implementação da infra-estrutura para manipulação dos modelos (leitura, escrita e persistência); (ii) um processador de templates que atendeu bem as necessidades demandadas da ferramenta; e (iii) a reutilização completa de um editor de modelo de características para ser usado na ferramenta.

Finalmente, a dissertação descreveu uma extensão dos modelos e das funcionalidades para endereçar a derivação automática de LPS implementadas com as tecnologias de componentes Spring e OSGi. Em particular, foi

apresentado: (i) como novas abstrações relacionadas com as tecnologias Spring e OSGi, foram incorporadas aos modelos de implementação e configuração; (ii) como elementos de implementação anotados e arquivos de configuração de tais tecnologias foram utilizados na criação de uma versão inicial dos modelos; e (iii) como essas novas informações são utilizadas para derivação, configuração e *deployment* de produtos baseado nas tecnologias Spring e OSGi. Um novo estudo de caso, denominado JPetStore, foi realizado para validar as novas extensões da ferramenta GenArch.

7.1. Contribuições

As seguintes contribuições são resultados diretos dessa dissertação:

- **Uma abordagem para derivação de produtos de software.** Definição de uma abordagem baseada em três modelos (implementação, configuração e características) para a preparação de LPS ou *frameworks*, para serem derivadas automaticamente. A abordagem também define como versões iniciais dos modelos podem ser geradas automaticamente a partir: (i) da especificação de anotações no código de artefatos de implementação que agregam as variabilidades da LPS; ou (ii) da análise gramatical (*parsing*) de arquivos de configuração que especificam os componentes da aplicação e suas propriedades.;

- **Ferramenta GenArch.** Projeto e implementação de uma ferramenta para derivação automática de membros de uma LPS ou instâncias de um *framework*. A ferramenta GenArch foi totalmente desenvolvida com tecnologias de código livre (*open-source*). Ela foi projetada e implementada como um plug-in da plataforma Eclipse (Shavor et al. 2003) com o suporte de três plug-ins existentes para esta plataforma: EMF, oAW e FMP;

- **Estudo e exploração do uso de novas tecnologias na implementação de ferramentas baseadas em modelos.** A plataforma extensível do Eclipse foi brevemente apresentada juntamente com três plug-ins (EMF, oAW e FMP) que oferecem diversas funcionalidades para desenvolvimento baseado em modelos. Também foi apresentado como esses plug-ins foram utilizados e combinados na implementação da ferramenta GenArch;

- **Estudos de Caso.** Realização de 3 estudos de caso de preparação e derivação de LPS (JUnit, Jogo Rain of Fire e JPetStore) usando a ferramenta/abordagem proposta.

- **Customização de aplicações corporativas.** Projeto e implementação de uma extensão da ferramenta GenArch para endereçar a derivação de produtos baseados nas tecnologias de componentes Spring e OSGi. Essa extensão permite a customização automática de propriedades nos descritores dos componentes (*beans* e *bundles*) e seleção de quais componentes constituirão o produto final gerado.

7.2. Discussões e Lições Aprendidas

Como parte da realização do trabalho, algumas lições aprendidas e discussões foram obtidas baseados na experiência de desenvolvimento e uso preliminar da ferramenta GenArch. Nesta seção são discutidas e apresentadas algumas dessas lições aprendidas.

Estratégias de Adoção de LPS. Diferentes tipos de estratégias de adoção podem ser utilizadas no desenvolvimento de LPSs. Independente da estratégia adotada, uma ferramenta de derivação sempre precisa reduzir o custo de instanciação de arquiteturas de LPSs complexas. Acreditamos que a ferramenta GenArch pode ser utilizada em conjunto com as abordagens de adoção de LPSs: pró-ativa, extrativa e incremental (Seção 2.1). Na abordagem pró-ativa, a ferramenta pode ser usada na anotação dos elementos de implementação produzidos durante a fase de engenharia de domínio de maneira a preparar estes elementos para serem automaticamente derivados durante a engenharia de aplicação. A abordagem extrativa também pode demandar a introdução de anotações GenArch em classes, interfaces ou aspectos, sempre que novos pontos de extensão são expostos, gradualmente, durante a transformação de produtos existentes em uma LPS. Finalmente, a abordagem reativa requer uma funcionalidade de sincronização, porque esta pode envolver cenários complexos de evolução. Ambas as abordagens extrativa e reativa requerem o uso de ferramentas de refatoração no auxílio à extração e incorporação de características a uma LPS (Alves et al. 2005).

Integração com ferramentas de refatoração. Aplicação de técnicas de refatoração são comuns atualmente no desenvolvimento de sistemas de software. No contexto do desenvolvimento de LPSs, técnicas de refatoração precisam considerar (Alves et al. 2006), por exemplo: (i) se mudanças aplicadas na estrutura dos elementos de implementação de uma LPS não diminui o conjunto de todas as configurações possíveis (produtos) endereçadas pela LPS;

e (ii) cenários complexos de incorporação de programas já existentes em uma LPS. Embora muitas refatorações propostas introduzam pontos de extensão ou variabilidades (Alves et al. 2005), as ferramentas de refatoração existentes não estão bem integradas com ferramentas de modelagem e derivação de LPS. Isso pode trazer dificuldades ou inconsistências quando ambas as ferramentas são utilizadas em conjunto no desenvolvimento de LPS. A integração da ferramenta GenArch com ferramentas de refatoração existentes envolve diversos desafios, como, por exemplo: (i) permitir a criação de anotações `@Feature` para todas as refatorações que expõem ou criam uma nova característica variável, de forma que tal característica se torne presente no modelo de características para permitir sua derivação automática; e (ii) refatorações que introduzem novos pontos de extensão (como, classe ou aspecto abstratos ou interface) devem ser integradas com a ferramenta GenArch para permitir a inserção automática da anotação `@Variability`. A funcionalidade de sincronização entre modelos, código e anotações também é fundamental nesse contexto de integração com ferramentas de refatoração, pois ela garante que todas as refatorações aplicadas em elementos de implementação existentes, e que ocasionem a criação de novas anotações ou a modificação das já existentes, serão sincronizadas com os modelos de derivação.

Customização Dinâmica de LPS. Muitas infra-estruturas de tecnologias de componentes desenvolvidas nos últimos anos vêm enfatizando a necessidade de suporte a adaptação e customização dinâmica de aplicações. A tecnologia de componentes J2EE (Roman et al. 2004), por exemplo, permite o *deployment* dinâmico de Enterprise Java Beans (Roman et al. 2004). O *deployment* de novos componentes ou o gerenciamento (atualização ou remoção) dos já existentes é suportado por mecanismos fornecidos pelo próprio servidor de aplicação, como o JBoss (JBoss 2008). A especificação OSGi também permite o gerenciamento dinâmico de componentes. No entanto, esta tecnologia ainda não é muito explorada no contexto de aplicação Java corporativas. O *Spring Dynamic Modules* (SpringDynamicModules 2008) é uma iniciativa neste sentido. A extensão Spring/OSGi da ferramenta GenArch (Capítulo 5) já representa um avanço no uso dessas tecnologias na implementação de arquiteturas de LPS. Outra linha de pesquisa interessante para ser investigada é a customização dinâmica de LPS baseada em modelos, como por exemplo, o de características. Em uma abordagem desse tipo, uma ferramenta de derivação baseada no modelo de características poderia guiar também a customização dinâmica baseada na seleção de características. As

características selecionadas pelo engenheiro de aplicação demandariam a instalação (*deployment*) automática (remoção e atualização) de diversos componentes associados às características selecionadas.

7.3. Trabalhos Futuros

Essa seção apresenta uma lista de trabalhos futuros como continuidade dessa dissertação de mestrado. Alguns desses trabalhos já estão em andamento como continuidade das pesquisas do Laboratório de Engenharia de Software (LES) da PUC-Rio.

- **Integração do modelo de características com outras linguagens específicas de domínio.** Estabelecer mecanismos para o uso combinado entre o Modelo de Características e outras Linguagens Específicas de Domínio (gráficas, textuais e baseadas em modelos) é uma linha de trabalho que pretendemos explorar no contexto da ferramenta GenArch. A integração de várias DSLs a uma ferramenta de derivação de produto oferece mais expressividade para especificar as variabilidades de uma LPS, ampliando em muitos casos a sua capacidade de geração. Como parte dessa linha de trabalho, espera-se também definir um método sistemático para o projeto e implementação de LPSs baseado na composição de DSLs.

- **Refatoração da arquitetura da ferramenta GenArch para ser extensível.** A arquitetura atual da ferramenta GenArch está modularizada em módulos específicos e bem definidos. Entretanto, a ferramenta não oferece ainda pontos de extensão explícitos para a incorporação flexível de: (i) novos mecanismos para análise gramatical (*parsing*) de artefatos de tecnologias existentes; (ii) novas funcionalidades de extensão dos modelos de arquitetura e de configuração que contemplem novas tecnologias de implementação de LPS; (iii) novas estratégias de sincronização de modelos e artefatos de código; e (iv) permita a integração de novas DSLs na ferramenta.

- **Desenvolvimento de novos estudos de caso.** Embora alguns estudos de caso tenham sido apresentados nesta dissertação, novas LPS e frameworks de larga escala precisam ser desenvolvidos ou refatorados de forma a validar e evoluir a abordagem proposta. Os seguintes estudos de caso estão sendo atualmente explorados pelo grupo de pesquisa do LES/PUC-Rio: (i) uma LPS de sistemas multi-agentes que incorpora propriedades de agências a aplicações web; (ii) uma nova LPS de jogos J2ME que implementam suas variabilidades

usando também arquivos de *builds* do Ant; e (iii) uma linha de testes automatizados de segurança e desempenho para aplicações web.

- **Integração com ferramentas de refatoração.** Integrar a ferramenta GenArch com ferramentas de refatoração, de maneira que a primeira permita a evolução reativa da arquitetura da LPS a partir da criação de anotações `@Feature` para todas as refatorações que expõem novas variabilidades e anotações `@Variability` para todas as refatorações que introduzam novos pontos de extensão.

- **Instanciação de variabilidades presentes em códigos orientados a aspectos.** Programação orientada a aspectos pode ser usada para implementar variabilidades transversais específicas por meio da definição de comportamentos extensíveis. Diversos mecanismos para implementação dessas variabilidades estão presentes nas linguagens atuais, tais como: (i) a definição de aspectos, pontos de corte (*pointcuts*) e métodos abstratos em AspectJ (Kiczales et al. 2001); e (ii) especificação de pontos de corte concretos diretamente em arquivos de configuração Spring. Como trabalho futuro, pretendemos incorporar mecanismos na ferramenta GenArch que ofereça suporte para o gerenciamento e instanciação automática desses tipos de variabilidades. Em particular, esperamos incorporar à ferramenta, um modelo generativo orientado a aspectos anteriormente proposto (Kulesza et al. 2007). Essa integração envolve basicamente: (i) a definição de características *crosscutting* e *jointpoint* no modelo de características; e (ii) a especificação de mapeamentos entre características *crosscutting* e *pointcuts* no modelo de configuração.

7.4. Uso da Ferramenta GenArch na Indústria

A ferramenta GenArch foi experimentada em ambiente industrial dentro da empresa Meantime Mobile Creations. Tal empresa é incubada no Centro de Estudos e Sistemas Avançados do Recife (C.E.S.A.R) e pioneira no desenvolvimento e *publishing* de jogos para celular na América atina.

A experiência de uso da GenArch na Meantime consistiu na configuração de uma de suas LPS de jogos J2ME para celular. A configuração foi realizada com sucesso de forma que novos jogos (produtos) passaram a ser derivados automaticamente a partir do uso da ferramenta. Neste sentido, a experiência foi positiva e mostrou que é possível adotar a ferramenta em cenários reais de gerência de variações em linhas de produto de software. Vale ressaltar que em

tal cenário de uso, a ferramenta GenArch substituiu com sucesso a ferramenta líder mundial da área pure::variants, e foi considerada como uma alternativa a essa ferramenta devido ao seu alto custo. A simplicidade e agilidade da ferramenta agradaram bastante os engenheiros e arquitetos de software da Meantime, que estão iniciando a preparação para adotar a ferramenta na derivação de outras LPS, já de forma integrada com outras ferramentas disponíveis na empresa.

Essa experiência nos faz crer que a abordagem/ferramenta apresentada neste trabalho é atrativa e possui potencial para permitir a adoção da abordagem de LPS por parte de pequenas e médias empresas, que demandam a gerência de um grande número de variações em seus produtos.