

1

Introdução

Métodos e técnicas para o desenvolvimento de famílias de sistemas e linhas de produto [27, 33, 100, 117] têm demonstrado ao longo dos últimos anos ser uma abordagem promissora para tratar de problemas centrais da engenharia de software, tais como, aumento da produtividade e melhoria na qualidade do software produzido. Uma família de sistemas pode ser caracterizada como um conjunto de programas que possuem propriedades comuns e propriedades específicas que variam de acordo com o membro da família sendo considerado [98]. Uma linha de produto é definido como um conjunto de sistemas de software que compartilham um conjunto de funcionalidades comuns e que satisfazem necessidades específicas de um determinado segmento de mercado [27].

Abordagens de desenvolvimento de famílias de sistemas ou linhas de produto [27, 33, 100, 117] definem como aspecto central o projeto e implementação de uma arquitetura comum para todas as aplicações dentro de um escopo específico. Tal arquitetura contempla as funcionalidades comuns e variáveis compartilhadas por aplicações pertencentes ao escopo/domínio definido pela família de sistemas. Ela define um conjunto de componentes que são necessários para construir cada um dos sistemas (ou produtos) presentes em uma família de sistemas (ou linha de produto).

Frameworks Orientados a Objetos (OO) [37, 38, 64] representam atualmente uma tecnologia comum e relevante para a implementação de arquiteturas de família de sistemas ou linhas de produto. Eles permitem o reuso em larga escala e modular através: (i) do encapsulamento de interesses recorrentes de um dado domínio; e (ii) da disponibilização de diversas variabilidades e opções de configuração para as aplicações alvos. No desenvolvimento baseado em frameworks, aplicações são implementadas através do reuso da implementação da arquitetura definida pelo framework e da extensão de seus respectivos pontos de extensão/variação (*hot-spots*) [37]. Conseqüentemente a adoção da tecnologia de

framework traz, em geral, produtividade e qualidade para o desenvolvimento de novas aplicações.

1.1. Problema

Apesar dos benefícios trazidos pela tecnologia de frameworks OO, diversos pesquisadores e desenvolvedores têm identificado e relatado recentemente [5, 7, 14, 73, 80, 88, 89, 94] a inadequação de mecanismos OO para lidar com a modularização e composição de muitas características¹ (*features*) transversais encontradas em frameworks e arquiteturas de famílias de sistemas, trazendo dificuldades para a definição de variabilidades² no framework e integração com outros artefatos existentes. Exemplos de tais funcionalidades são: características opcionais, características alternativas transversais e características de integração transversal com outros artefatos (frameworks, componentes ou bibliotecas). Tal limitação na modularização de determinadas funcionalidades e propriedades de um framework OO traz como consequência: (i) o aumento da complexidade do framework com a inclusão de várias funcionalidades que não são úteis a muitas de suas instâncias; e (ii) dificuldade para configuração e customização do framework para atender diferentes cenários de reuso.

Trabalhos recentes [5, 7, 8, 73, 89, 94, 118] têm explorado o uso de técnicas de desenvolvimento de software orientadas a aspectos para facilitar a implementação de arquiteturas de famílias de sistemas flexíveis e customizáveis. Desenvolvimento de Software Orientado a Aspectos (DSOA) [42, 68] foi proposta como uma abordagem de engenharia de software que oferece mecanismos avançados para modularizar os chamados interesses transversais. Interesses transversais são funcionalidades ou propriedades do sistema que usualmente entrecortam diversos módulos em um sistema de software. Acredita-se que o isolamento dos interesses transversais possa trazer melhorias para o reuso e evolução de sistemas. DSOA pode ser vista como uma técnica complementar à

¹ Nesta tese de doutorado, uma característica é definida como sendo uma funcionalidade ou propriedade de uma família de sistemas ou linha de produto.

² Variabilidade é uma característica de uma família de sistemas ou linha de produto que varia de acordo com o membro da família ou produto sendo instanciado. Esse trabalho se concentra, principalmente, em variabilidades internas que são aquelas diretamente relacionadas com o projeto e implementação da arquitetura da família de sistemas ou linha de produto.

aquelas propostas por OO, a qual permite a modularização dos interesses transversais encontrados em sistemas OO.

Nesse contexto, este trabalho explora problemas existentes na modularização de determinadas características em frameworks OO e realiza uma análise sistemática de como técnicas orientadas a aspectos (OA) podem ajudar na sua resolução.

1.2. Limitações das Abordagens Atuais

Diversas abordagens baseadas em técnicas de modularização avançadas têm sido propostas ao longo dos últimos anos. Muitas delas se fundamentam em técnicas orientadas a aspectos. Zhang & Jacobsen [118] propõem um conjunto de princípios e refatorações para a modularização de arquiteturas OA flexíveis e configuráveis. Apel & Batory [7] demonstram como programação orientada a características [15, 111] pode ser usada em conjunção com orientação a aspectos para oferecer um melhor suporte a modularização de características em linhas de produto. Embora várias das abordagens propostas apresentem diretrizes gerais para o uso de aspectos na modularização de arquiteturas mais flexíveis e configuráveis, nenhuma delas investiga sistematicamente como a programação orientada a aspectos pode ser usada de forma complementar a técnica de frameworks OO auxiliando na modularização de suas características.

Também poucas das abordagens propostas discutem ou investigam mecanismos automáticos para a instanciação de arquiteturas de famílias de sistemas ou linha de produto considerando o uso da abstração de aspectos. A investigação de como novas abordagens de desenvolvimento de famílias de sistemas (exemplos: Desenvolvimento Generativo [32, 33] e Fábricas de Software [53, 54]) podem ser combinadas com técnicas OA para habilitar a instanciação de frameworks e suas variabilidades OO e OA também não tem sido tratada. Uma das poucas exceções é o trabalho de Lougran & Rashid [89]. Eles propõem a abordagem *Framed Aspects* que combina *frames* e aspectos para a implementação de arquiteturas flexíveis e usa os mecanismos de parametrização e geração de código oferecido pela técnica de *frames* para habilitar a instanciação automática das variabilidades presentes na arquitetura. *Frame* [11] é uma tecnologia que

permite particionar o código-fonte de um sistema ou componente em uma hierarquia de unidades menores, os chamados *frames*. Cada *frame* especifica a funcionalidade comum e variável de um dado elemento de implementação.

1.3. Solução Proposta

Esse trabalho propõe uma abordagem sistemática para o desenvolvimento de frameworks OO usando técnicas de DSOA [42, 68, 77]. O objetivo central da abordagem é melhorar a capacidade de extensão e configuração de frameworks OO para diferentes cenários, através de uma melhor gerência de suas características. A abordagem é composta por: (i) um conjunto de diretrizes para o projeto e implementação de frameworks usando programação orientada a aspectos; e (ii) um modelo generativo usado para a instanciação automática do framework e suas variabilidades, incluindo aquelas implementadas usando aspectos.

As diretrizes para o projeto e implementação de frameworks usando aspectos, são centradas no conceito de Pontos de Junção de Extensão (EJPs, do inglês, *Extension Join Points*) [73, 75, 82, 84] como uma forma unificada de projetar e documentar pontos de extensão transversais de frameworks. Os EJPs oferecem novas formas de extensão do núcleo do framework, através da exposição de pontos específicos da sua execução (eventos gerados ou estados atingidos), os quais podem ter sua funcionalidade estendida por meio da codificação de aspectos. A adição de novas funcionalidades no framework é feita através do uso de aspectos de variabilidade e de integração. Considerando que os aspectos podem ser facilmente adicionados e removidos do núcleo do framework, nossa abordagem facilita dessa forma a customização do framework para necessidades específicas. Os EJPs podem também ser vistos como a especialização do conceito de interfaces transversais (XPIs) para auxiliar a modularização e composição de frameworks OO. As XPIs são usadas para especificar interfaces entre o código base³ e um conjunto de aspectos.

³ Em DSOA, o termo código base é usado para se referir às classes do sistema que são potencialmente afetadas pelos aspectos.

Nosso trabalho também se fundamenta em técnicas de Desenvolvimento Generativo (DG) [32, 33]. DG envolve o estudo de métodos e ferramentas que habilitam a produção automática de membros de uma família de software a partir de especificações de alto nível. Nossa abordagem contempla a definição de um modelo generativo [32, 33] o qual permite instanciar automaticamente arquiteturas orientadas a aspectos, permitindo a geração e customização de variabilidades OO e OA. O modelo generativo pode ser adotado na derivação de instâncias de frameworks OA projetados e implementados usando nossas diretrizes. Ele é centrado na definição de: (i) um modelo de característica – usado para especificar e coletar informações necessárias para a customização de variabilidades OO e OA, existentes em classes e aspectos; (ii) um modelo de arquitetura – que agrega os elementos de implementação de uma arquitetura de família de sistemas, na forma de um conjunto de componentes; e (iii) um modelo de configuração – que permite definir o mapeamento entre características e elementos de implementação, presentes nos modelos de características e arquitetura, respectivamente. Nosso modelo generativo trata explicitamente da instanciação de variabilidades OA por meio da customização de pontos de corte de aspectos usando relações transversais entre características e mapeamento de características <<joinpoint>> em pontos de corte definidos nos EJPs.

1.4.

Organização do Texto

Essa tese de doutorado está organizada em mais 8 capítulos, além deste introdutório.

O Capítulo 2 apresenta as três principais abordagens de desenvolvimento de famílias de sistemas investigadas nesse trabalho: (i) frameworks orientados a objetos, (ii) desenvolvimento de software orientado a aspectos; e (iii) desenvolvimento generativo. O capítulo apresenta as deficiências trazidas por mecanismos OO para implementação de características transversais existentes em frameworks, e mostra o potencial que DSOA e DG trazem para a resolução de muitos desses problemas.

O Capítulo 3 traz uma visão geral da abordagem orientada a aspectos de desenvolvimento de frameworks proposta neste trabalho. Ela é composta de: (i)

um conjunto de diretrizes para implementação de frameworks usando aspectos; e (ii) um modelo generativo que permite a instanciação automática de arquiteturas OA.

O Capítulo 4 detalha as diretrizes para implementação de características transversais existentes em frameworks usando técnicas orientadas a aspectos. É apresentado como os principais elementos da abordagem, tais como pontos de junção de extensão (EJPs) e aspectos de variabilidade e integração podem ser implementados em AspectJ.

O Capítulo 5 apresenta um modelo generativo a aspectos, cujo objetivo central é habilitar a instanciação automática de arquiteturas de linha de produto que contenham variabilidades OO e OA, a partir de informações coletadas por um modelo de característica.

O Capítulo 6 descreve três estudos de caso de frameworks e arquiteturas de linha de produto que foram implementados em AspectJ usando a abordagem proposta. O modelo generativo OA de cada um dos estudos de caso é também apresentado.

O Capítulo 7 apresenta uma análise geral da abordagem proposta a partir dos estudos de caso realizados. São apresentados os benefícios gerais trazidos pela abordagem, assim como um estudo qualitativo e quantitativo da aplicação da abordagem em um dos estudos de caso. Diversas discussões e lições aprendidas a partir da experiência de uso da abordagem são também apresentadas.

O Capítulo 8 discorre sobre trabalhos relacionados com a proposta apresentada nesta tese. Finalmente, o Capítulo 9 conclui esta tese resumizando a sua lista de contribuições e sugerindo direções para a realização de trabalhos futuros.