3
Uma Abordagem Orientada a Aspectos para o Desenvolvimento de Frameworks

Este capítulo apresenta uma visão geral da contribuição principal deste trabalho: uma abordagem orientada a aspectos para o desenvolvimento de frameworks. A abordagem tem como objetivo uma melhor modularização de características transversais encontradas em frameworks de forma a facilitar a sua customização para diferentes cenários. Ela é composta por: (i) um conjunto de diretrizes que auxiliam o desenvolvedor a definir quais funcionalidades do framework devem ser projetadas e implementadas usando aspectos; e (ii) um modelo generativo que permite a instanciação automática das variabilidades do framework, incluindo aquelas implementadas usando programação orientada a aspectos, a partir de um modelo de características.

PUC-Rio - Certificação Digital Nº 0310904/CA

3.1.
Diretrizes para Implementação de Frameworks com Aspectos

Essa seção apresenta os elementos que compõem as diretrizes para modularização de frameworks usando aspectos, sendo eles: (i) pontos de junção de extensão; (ii) núcleo do framework; e (iii) aspectos de extensão.

3.1.1.
Pontos de Junção de Extensão (EJPs)

Os pontos de extensão (*hot-spots*) de um framework OO são tipicamente implementados através de classes abstratas ou interfaces. Eles possibilitam a extensão do comportamento comum de colaboração fornecido pelo framework oferecendo implementações concretas para seus respectivos métodos abstratos.

Na abordagem proposta, um framework OO especifica e implementa não apenas suas funcionalidades comuns e variáveis usando classes, mas também expõe um conjunto de pontos de junção de extensão [73, 75, 84] (EJPs, do inglês *Extension Join Points*). Os EJPs expõem pontos específicos da execução do

framework, os quais podem ter sua funcionalidade estendida por meio da codificação de aspectos. Dessa forma, eles são adotados como uma forma de facilitar a implementação de variabilidades transversais e de integração. Os EJPs também estabelecem contratos entre as classes do framework e o conjunto de aspectos que estendem sua funcionalidade básica. Eles podem ser usados para dois diferentes propósitos:

(i) expor um conjunto de eventos do framework que podem ser usados para notificar ou facilitar uma integração transversal com outros módulos de software, tais como, frameworks, componentes ou bibliotecas;

(ii) oferecer pontos de execução pré-definidos que estão espalhados e/ou entrelaçados no framework e nos quais pode ser incluída a implementação de características opcionais ou alternativas transversais.

Dessa forma, os EJPs documentam pontos de extensão transversais para desenvolvedores de software que desejam instanciar e estender o framework. Eles podem também ser vistos como um conjunto de restrições no espaço total de pontos de junção existentes no framework. O Capítulo 4 mostra como os EJPs podem ser implementados em AspectJ.

PUC-Rio - Certificação Digital Nº 0310904/CA

3.1.2. Núcleo do Framework e Aspectos de Extensão

Nossa abordagem promove o desenvolvimento de frameworks como uma composição de uma estrutura núcleo e um conjunto de aspectos de extensão. Um aspecto de extensão pode endereçar: (i) a implementação de características opcionais ou alternativas transversais do framework; ou (ii) a integração transversal com um componente, biblioteca ou framework adicional. A composição entre o núcleo do framework e suas extensões é realizada por diferentes tipos de aspectos. Cada aspecto define uma composição transversal com o framework através dos EJPs expostos. A seguir, os principais elementos da abordagem são descritos:

(i) **núcleo do framework** – implementa a funcionalidade obrigatória de uma família de software. De forma similar a um framework OO tradicional, a estrutura do núcleo contém a implementação de classes que representam seus pontos fixos (*frozen-spots*) com a funcionalidade comum da família de software, e de classes

que representam pontos flexíveis (*hot-spots*) não transversais do domínio endereçado;

(ii) **aspectos do núcleo** – implementam e modularizam interesses ou papéis tranversais existentes nas classes do núcleo do framework. Eles representam o uso tradicional de POA para simplificar o entendimento e evolução do núcleo;

(iii) **aspectos de variabilidade** – implementam características opcionais e alternativas transversais existentes no núcleo do framework. Tais elementos estendem os EJPs do framework com algum comportamento transversal adicional;

(iv) **aspectos de integração** – definem composições transversais entre o núcleo do framework e outras extensões existentes, tais como, uma biblioteca, um componente ou um framework. Esses elementos também dependem dos EJPs para definir sua implementação.

A Figura 8 mostra o projeto do framework OO com aspectos seguindo as diretrizes de nossa abordagem. Como podemos ver na figura, os aspectos de integração e variabilidade podem atuar apenas nos EJPs oferecidos pelo framework e devem obedecer todas as restrições definidas pelos mesmos.

PUC-Rio - Certificação Digital Nº 0310904/CA

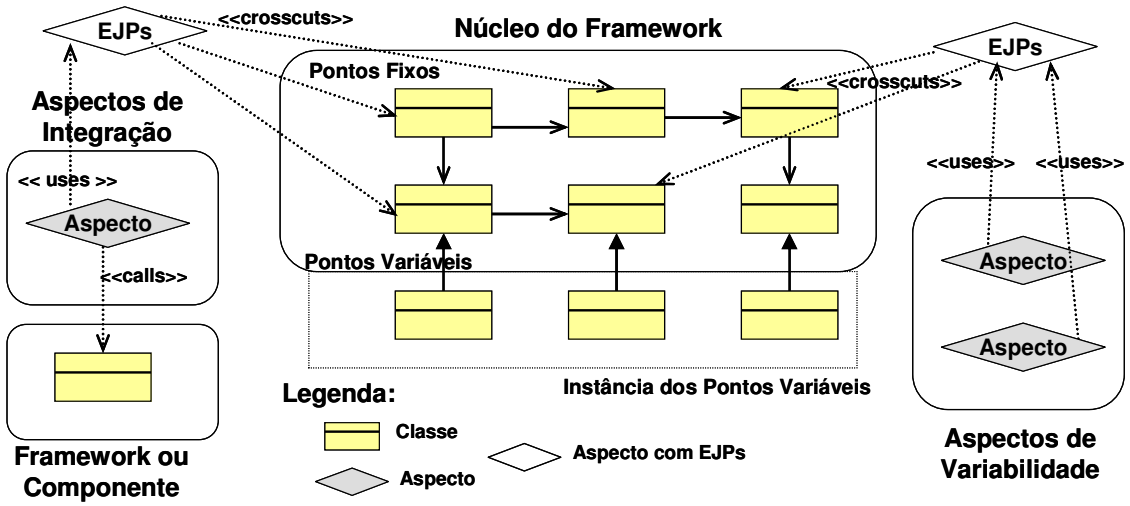

Figura 8. Elementos de Implementação da Abordagem Proposta

3.2. Um Modelo Generativo Orientado a Aspectos

O resultado do projeto e implementação de uma arquitetura de família de sistemas ou linha de produto usando as diretrizes para modularização de características de frameworks é um conjunto de artefatos (classes, aspectos, arquivos de configuração, etc) os quais endereçam as características comuns e variáveis de um dado domínio. Muito das classes e aspectos definidos para a arquitetura serão instanciados apenas se eles forem necessários para implementar uma aplicação ou produto específico. Dessa forma, a seleção manual e customização de tais elementos pode se tornar uma atividade complexa a qual torna extremamente custoso ou até impraticável o uso da abordagem. Para facilitar a instanciação do framework e seus respetivos aspectos de extensão, um modelo generativo orientado a aspectos é proposto. O objetivo principal de tal modelo generativo é permitir a configuração de uma instância do framework a partir de um modelo de características.

Nosso modelo generativo OA [74, 76, 81] segue a estrutura geral apresentada por Czarnecki e Eisenecker [33] (Seção 2.3). Entretanto, é proposta a extensão de tal modelo para suportar a instanciação de arquiteturas OA, permitindo a geração e customização de variabilidades OO e OA. Nosso modelo é composto por:

(i) **um modelo de característica** – este modelo funciona como uma linguagem específica de domínio de configuração [33]. Ele é responsável pela especificação e coleta de informação necessária para a customização das variabilidades OO e OA, existentes em classes e aspectos. Um conjunto de relacionamentos tranversais entre características é usado em tal modelo para auxiliar na customização de pontos de corte de aspectos;

(ii) **um modelo de arquitetura OA** – este modelo define os principais componentes de uma arquitetura de família de sistemas ou linha de produto. A arquitetura contém um conjunto de variabilidades que precisam ser customizadas para definir uma aplicação completa. Variabilidades transversais são implementadas como aspectos nessa arquitetura. Cada componente é definido como um conjunto de classes, aspectos, templates e arquivos extras. Os templates representam elementos (classes, aspectos, arquivos de configuração) que serão

PUC-Rio - Certificação Digital Nº 0310904/CA

customizados durante o processo de instanciação automático. As diretrizes para modularização de características em frameworks de nossa abordagem (Seção 3.1.1) são usadas como a base principal para implementação da arquitetura OA;

(iii) **um modelo de configuração** – esse modelo especifica o mapeamento entre os elementos definidos no modelo de característica e os componentes (e respectivos sub-elementos, tais como, classes, aspectos, arquivos extras ou templates) do modelo de arquitetura OA. Tal modelo também permite definir explicitamente o mapeamento entre features transversais e aspectos possibilitando a geração de pontos de corte específicos em aspectos. Todas as informações providas pelo modelo de configuração são usadas para auxiliar no processo de decidir quais componentes devem ser instanciados e quais customizações devem ser realizadas em tais componentes considerando uma aplicação específica.

Os elementos existentes (modelos de característica, de arquitetura e de configuração) em nosso modelo generativo são definidos para habilitar seu uso por uma ferramenta de customização e geração de código. O Capítulo 5 detalha cada um desses modelos e apresenta diretrizes de tecnologias que podem ser utilizadas para a sua definição. Um algoritmo de processamento de tais modelos que gera como resultado uma aplicação específica é também apresentado.

PUC-Rio - Certificação Digital Nº 0310904/CA

3.3. Fluxo de Atividades da Abordagem

Existem diversas atividades envolvidas no processo de desenvolvimento dos elementos de nossa abordagem. As Figuras 9 e 10 apresentam esse conjunto de atividades, organizadas sob as perspectivas da engenharia de domínio [33] e da engenharia de aplicação [33], respectivamente. Na engenharia de domínio, nossa abordagem endereça atualmente apenas as fases de projeto e implementação de domínio, oferecendo um conjunto de atividades para o desenvolvimento de frameworks usando mecanismos OO e OA (atividades I, II, III, IV e V). Também outras atividades endereçam a especificação dos diversos elementos que compõem o nosso modelo generativo (atividade V, VI, VII, VIII e IX), de forma a apoiar a instanciação automática do framework e suas variabilidades durante a engenharia de aplicação. A seguir são apresentadas descrições de cada uma das atividades da engenharia de domínio e de aplicação. Embora tais atividades

estejam organizadas de forma seqüencial, elas não necessariamente precisam ocorrer nessa ordem. A especificação do modelo de característica, por exemplo, pode ocorrer em paralelo com as demais atividades.

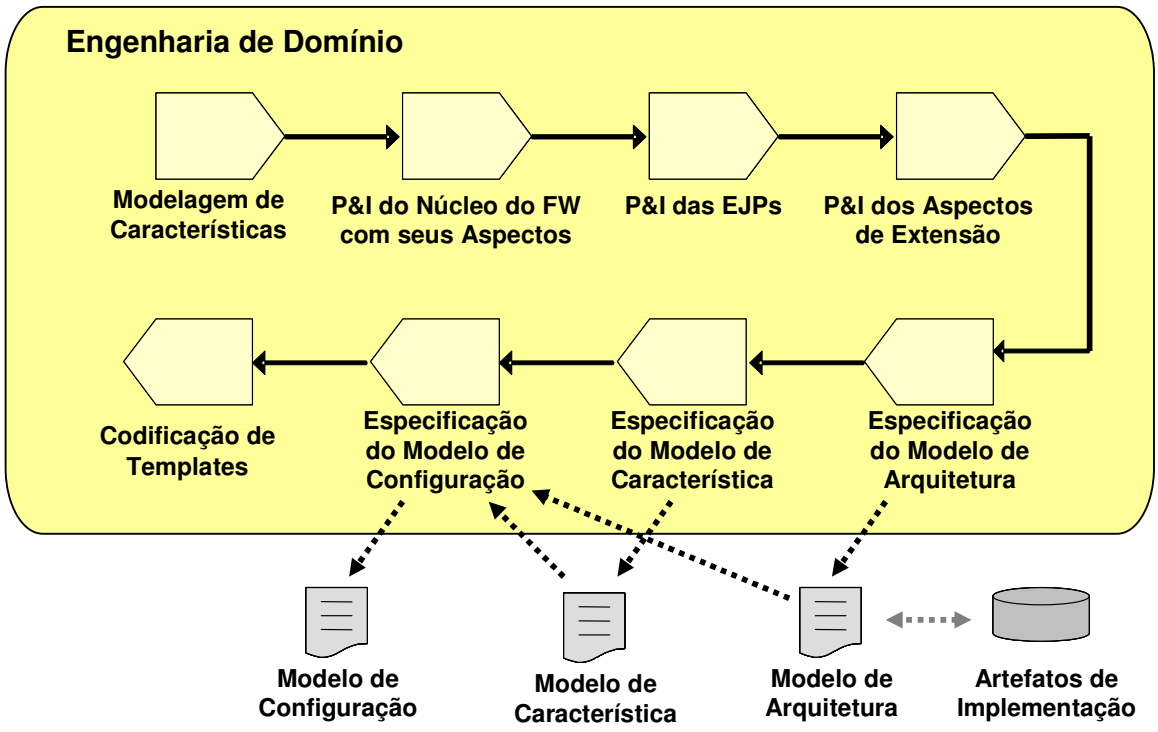

Figura 9. Atividades de Engenharia de Domínio da Abordagem

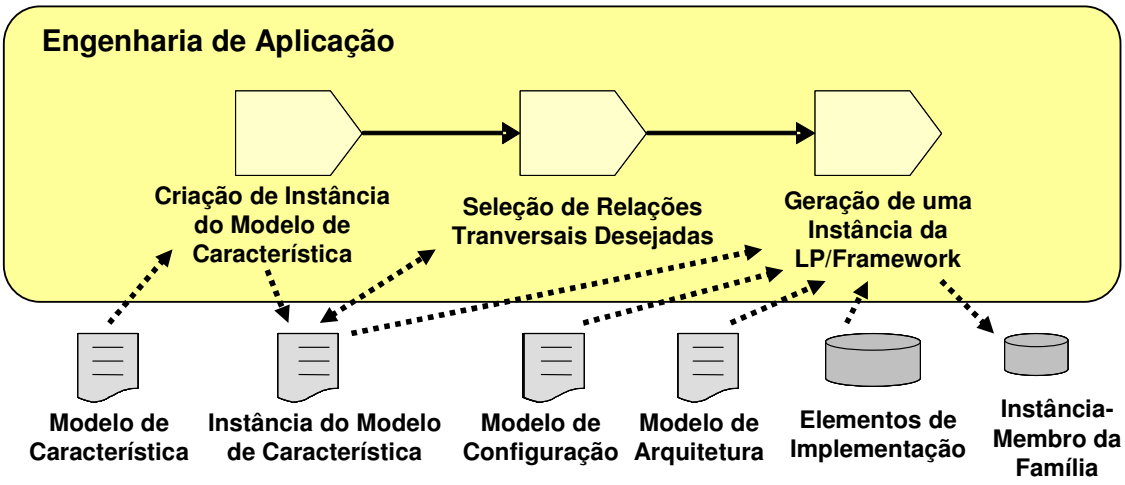

Figura 10. Atividades de Engenharia da Aplicação da Abordagem

PUC-Rio - Certificação Digital Nº 0310904/CA

FASE: ENGENHARIA DE DOMÍNIO

Atividades de Desenvolvimento do Framework

(I) **modelagem de características comuns e variáveis do framework** – essa atividade se concentra na representação de características comuns e variáveis a serem implementadas pelo framework, em um modelo de característica. Ela é, em geral, precedida por atividades de análise de domínio que buscam a identificação e elicitação das características comuns e variáveis existentes no domínio do framework;

(II) **projeto e implementação do núcleo do framework** – essa atividade objetiva a implementação da funcionalidade comum a qualquer uma das instâncias do framework. Ela envolve o desenvolvimento das características obrigatórias (*frozen-spots*) do framework e também das características variáveis não transversais (*hot-spots*) que devem sempre ser instanciadas pelos usuários do framework, usando mecanismos OO tradicionais, tais como, herança, agregação e composição. Padrões de projeto (tais como, *Strategy* e *Template Method*) são bastante úteis para o projeto OO de características variáveis não transversais;

(III) **projeto e implementação de aspectos do núcleo** – essa atividade busca oferecer uma melhor modularização dos interesses transversais existentes no núcleo, aspectos podem ser codificados para facilitar o entendimento e/ou manutenção do núcleo do framework;

(IV) **projeto e implementação dos EJPs** – cada framework pode, além de oferecer pontos de extensão OO (por meio de interfaces ou classes abstratas), expor um conjunto de pontos de junção para a inclusão de extensões transversais. Essa atividade objetiva a identificação de eventos ou estados relevantes da execução do framework que possam ser relevantes para a realização de alguma composição transversal com o framework;

(V) **projeto e implementação dos aspectos de extensão** – nessa atividade, aspectos de extensão são implementados para endereçar as características transversais opcionais, alternativas e de integração do framework. Cada um dos aspectos de extensão deve atuar exclusivamente nos EJPs expostos pelo framework. Aspectos de extensão podem também por sua vez oferecer alguma variabilidade na sua implementação.

PUC-Rio - Certificação Digital Nº 0310904/CA

Atividades de Especificação do Modelo Generativo

(VI) **especificação do modelo de arquitetura** – essa atividade envolve a representação de todos os elementos implementados para a arquitetura da família de sistemas ou framework, como um conjunto de classes, aspectos, templates e arquivos extras. Uma ferramenta pode auxiliar na geração semi-automática de tal modelo, a partir do processamento do diretório contendo os artefatos de implementação da arquitetura;

(VII) **especificação do modelo de características** – após a implementação do framework, e seus EJPs e aspectos de extensão, é necessário revisitar e atualizar o modelo de característica produzido como resultado da atividade (I), com eventuais modificações realizadas nas variabilidades do framework;

(VIII) **especificação do modelo de configuração** – essa atividade envolve, principalmente, a definição de um conjunto de relações de dependência entre características e elementos de implementação do framework (classes, aspectos, templates, arquivos de configuração, etc);

(IX) **codificação de templates** – envolve o uso de atributos/propriedades capturadas pelo modelo de característica para codificação de um dado elemento (classe, aspecto ou arquivo de configuração), cuja customização depende de informações oferecidas por tais atributos/propriedades;

FASE: ENGENHARIA DA APLICAÇÃO

A engenharia de aplicação é simplificada devido à preparação do framework para ser automaticamente instanciado durante a engenharia de domínio. Essa fase é composta das seguintes atividades:

(I) **criação de uma instância do modelo de características** – criação de uma instância do modelo de características da família de sistemas, e seleção das variabilidades (características varíaveis) que se deseja incluir no framework, com a respectiva configuração de suas propriedades;

(II) **seleção de relações transversais** – envolve a escolha ou seleção de relações tranversais válidas no modelo de características. Ela permite a customização de pontos de junção específicos onde aspectos de extensão irão atuar;

PUC-Rio - Certificação Digital Nº 0310904/CA

(III) **criação de uma instância da arquitetura de linha de produto / framework** – essa atividade envolve a solicitação de criação de uma instância da arquitetura da família de sistemas, a partir: da instância do modelo de característica definido (atividade I) com respectivas relações traversais entre características (atividade II), e dos modelos de arquitetura e configuração da família ou linha de produto sendo considerada. Uma ferramenta de instanciação é responsável por processar tais modelos e gerar uma aplicação que atenda a solicitação do engenheiro de aplicação.

3.4. Sumário

Este capítulo apresentou uma visão geral da abordagem OA para desenvolvimento de framework proposta nesse trabalho. Ela é centrada na definição de um conjunto de pontos de junção de extensão (EJPs) no núcleo de framework de forma a facilitar a implementação de extensões transversais. Os capítulos seguintes detalham as diretrizes de modularização de frameworks usando aspectos e o modelo generativo proposto para instanciação de suas variabilidades.

PUC-Rio - Certificação Digital Nº 0310904/CA