

1 Introdução

O princípio de separação de *concerns* permeia todo o processo de desenvolvimento de software. Ele cuida das limitações humanas em tratar a complexidade, permitindo que o desenvolvedor se concentre em um assunto ou característica do sistema por vez (Dijkstra, 1976), e desta forma melhora sua compreensão sobre o sistema e aumenta a capacidade para evolução e reuso.

Não há uma clara definição para *concern*. Algumas definições possíveis são: “é uma propriedade observável desejada”, “é uma *feature*”, “é alguma responsabilidade”, “é um subproblema” (Tekinerdođan, 2004). Neste trabalho, chamamos *concerns* de características¹. Elas podem ser qualquer característica (instâncias de: requisito, dado, função, restrição, objeto, custo, dentre outras) do sistema ou do domínio, que vale a pena analisar isoladamente ou em conjunto com outras.

Em Engenharia de Software, o princípio de separação de características está relacionado à decomposição e modularização (Parnas, 1972); sistemas de software complexos devem ser decompostos em unidades modulares menores, cada uma tratando uma única característica. Como definido pela programação modular, independentemente do método utilizado, um módulo é uma unidade compilável e integrável para formar o programa, e possui as seguintes propriedades (Staa, 2000):

- encapsulamento – o módulo deve tornar invisível para os módulos clientes sua implementação interna, disponibilizando seus serviços apenas por sua interface;

¹ Em (Piveta, 2004) a tradução dada para “*concerns*” é “interesses”. Nós não adotamos esta tradução porque consideramos que “interesses” em engenharia de requisitos têm uma conotação de que há alguém interessado, podendo, algumas vezes, ser confundida com “pontos de vista”. Neste trabalho, queremos ter uma conotação mais ampla e independente de haver ou não um interessado, daí escolhemos a palavra “característica” e utilizamos o termo “características transversais” como tradução para “*crosscutting concerns*”

- acoplamento – o módulo deve ter menor acoplamento com os demais, i.e., menor dependência; e
- coesão – os elementos que constituem um módulo devem ter um forte inter-relacionamento, devendo relacionar-se a um único conceito ou característica.

As propriedades de encapsulamento, acoplamento e coesão são essenciais para gerenciar a complexidade de partes do sistema e dele como um todo. Entretanto, cada método de modelagem e programação utilizado no desenvolvimento impõe uma maneira dominante de decomposição e modularização, explicitando algumas características e ocultando outras, igualmente importantes, tais como objetos, funções, tempo, dados, dentre outras (Tarr, 1999).

Na Figura 1, Chavez (2004) ilustra os problemas de espalhamento e entrelaçamento causados pelas abordagens procedural e orientada a objetos. Na Figura 1a, o mesmo dado pode ser utilizado em diferentes funções (procedimentos), e cada função trata de diferentes dados. Na Figura 1b, o mesmo comportamento (método) é desempenhado em diferentes componentes e cada componente desempenha comportamentos diferentes.

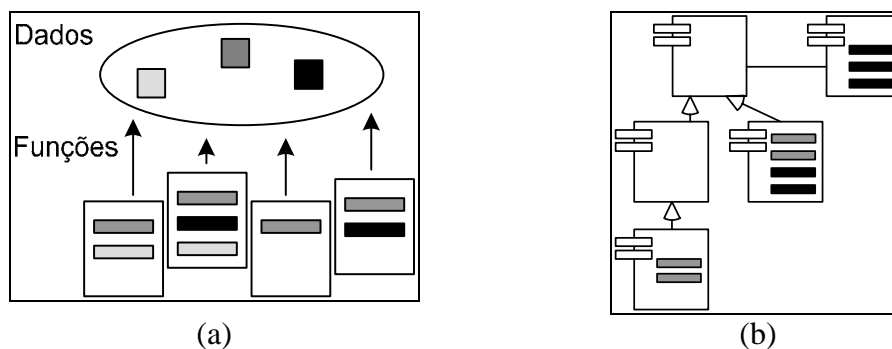


Figura 1. Problemas de espalhamento e entrelaçamento em abordagens (a) funcionais e (b) orientadas a objetos

Tarr (1999) argumenta que a decomposição dominante, seja funcional ou orientada a objetos, influencia negativamente a rastreabilidade, evolução e reuso, e argumenta sobre a importância de termos abordagens que tratem, simultaneamente, da separação de características em múltiplas dimensões. Desta forma, pode-se abordar os problemas de espalhamento e entrelaçamento que são ocasionados pela utilização de modelos parciais de decomposição (objetos, funções, dentre outros).

Além disto, um sistema é naturalmente composto de diferentes

características e entendê-las em conjunto é igualmente importante. Se por um lado precisamos separar as características, por outro precisamos compô-las. Muitas vezes estas características são fortemente relacionadas, entrelaçadas e/ou sobrepostas, influenciando ou restringindo umas às outras, sendo chamadas de **características transversais** (Tekinerdođan, 2004). Isto torna difícil a separação e análise das partes do sistema bem como a análise do impacto que umas exercem sobre as outras.

Para melhorar a separação e composição de características transversais, nos últimos anos, houve um investimento no paradigma de orientação a aspectos (Kiczales, 1997). Linguagens de programação orientadas a aspectos dão suporte à composição de características separadas e encapsuladas em “aspectos”², tornando a tarefa de composição transparente para o programador. Entretanto, tratar características transversais somente no nível de implementação não é suficiente porque muitas decisões de planejamento e de desenho³ já foram tomadas sem levar em consideração esta natureza transversal (Nuseibeh, 2004). Além disso, muitas vezes já não se tem mais o rastro entre os “aspectos” e o que deu origem a eles, dificultando o entendimento de onde mudanças em requisitos afetam.

Assim, têm surgido linguagens e métodos de modelagem considerando estes novos elementos e relacionamentos de desenho, permitindo a separação e composição de características transversais em um nível mais alto de abstração (Chavez, 2004). Identificar, separar e compor características transversais de alto nível de abstração provê: melhor entendimento do problema e da solução, rastreabilidade entre os diversos modelos construídos durante o desenvolvimento e entre as diversas características, reusabilidade de soluções de maior granularidade, e diminuição da distância entre as fases de desenvolvimento (Rashid, 2004; Baniassad, 2004; Brito, 2004).

Além disto, os fundamentos de separação e composição de características transversais inspiram o desenvolvimento de ferramentas aptas a extrair diferentes características de modelos projetados em qualquer linguagem de programação ou de desenho. Neste sentido, podemos prover, de maneira menos intrusiva, os

² Neste Capítulo, utilizamos o termo “aspecto” entre aspas, num sentido geral, referindo elementos utilizados em linguagens de programação para modularizar características transversais.

³ Nesta tese, utilizamos o termo “desenho” como tradução para o termo “design”, no sentido da atividade de desenvolvimento que resulta no projeto arquitetural e detalhado.

benefícios da separação e composição de características aos métodos e técnicas já utilizados para desenvolvimento de software. Desta forma, a separação e composição sugerem que podemos decompor o sistema utilizando abordagens diferentes ou que podemos decompor utilizando uma única abordagem, mas estamos aptos a visualizar e manipular outras formas de decomposição.

Esta tese tem por objetivo prover mecanismos de separação, composição e visualização de características transversais para melhorar o entendimento, manutenibilidade e reusabilidade de modelos de requisitos. Na Seção 1.1 explicamos o problema a ser tratado nesta tese. Na Seção 1.2 apresentamos as limitações das abordagens existentes. Na Seção 1.3 resumimos a solução proposta nesta tese. Na Seção 1.4 citamos as contribuições oferecidas. Na Seção 1.5 descrevemos como o restante desta tese está organizado.

1.1. Problema

Em documentos de requisitos, seja considerando um conjunto de modelos ou considerando apenas um tipo de modelo de requisitos, percebemos o espalhamento e entrelaçamento de muitas características. Na maioria das vezes, estas características são classificadas e separadas (decompostas) de maneiras diferentes, baseadas, por exemplo: em se são requisitos funcionais, não funcionais ou inversos; nas pessoas que estão interessadas no serviço, que o requisitaram, ou que o implementam; em opiniões dadas pelos requerentes; ou no tipo de serviço que oferecem; dentre outras.

A classificação e separação de características são úteis para facilitar a leitura do modelo utilizado e identificar nele as partes importantes em cada momento do processo, seja para elicitar novas características, validar com o usuário, analisar conflitos ou simplesmente entregar a especificação aos desenvolvedores. Entretanto, muitas vezes estas características estão transversais; por exemplo, um requisito funcional oferece um tipo de serviço e pode ter sido requisitado por diferentes pessoas interessadas nele e cada pessoa pode dar uma opinião complementar ou conflitante sobre o que ele é.

Neste caso, uma única categorização não é suficiente para satisfazer às necessidades dos engenheiros de requisitos, e utilizar diferentes categorizações requer maior trabalho em identificar as partes que interessam em cada momento

do processo de definição. As informações estão entrelaçadas e espalhadas, e quanto mais tentamos torná-las menos entrelaçadas mais as espalhamos, e vice-versa.

Se considerarmos o processo de requisitos isoladamente, os problemas de espalhamento e entrelaçamento em documentos de requisitos trazem sérias dificuldades para o entendimento de cada requisito, para manipulação do documento, para a reutilização de partes da especificação e para a modificação nos requisitos. Estes problemas trazem dificuldades ainda maiores se consideramos o processo de desenvolvimento de software inteiro, pois este depende da boa elaboração dos requisitos para o efetivo desenvolvimento e aceitação do produto (Sommerville, 2005).

O problema abordado nesta tese é o espalhamento e entrelaçamento de características transversais em modelos de requisitos. Consideramos que eles são problemas naturais, decorrentes da complexidade dos sistemas e, portanto, que é necessário sabermos amenizar seus danos para podermos conviver com eles.

1.2. Limitações dos Trabalhos Relacionados

Durante a modelagem de requisitos, a separação e composição de características têm sido tratadas parcialmente por abordagens tradicionais, tais como:

- Separação - definição de linguagens que permitem a descrição de requisitos funcionais (RFs), requisitos não funcionais (RNFs), ou ambos, seqüência de ações, intencionalidade, influência entre requisitos, interação entre interessados e funcionalidades, dentre outros; descrição de opiniões de cada interessado no sistema (pontos de vista) (Finkelstein, 1992, Nuseibeh, 1994); descrição modular dos requisitos, principalmente de RNFs, que são catalogados de acordo com algum método, para posterior reutilização (Cysneiros, 2003).
- Composição - frameworks para integração de modelos (Nuseibeh, 1994); e processos para integração de RNFs e RFs (Cysneiros, 2001).

Estas abordagens de separação não são suficientes porque permitem apenas uma forma de decomposição de requisitos e não deixam explícitas as características transversais, nem como elas interagem umas com as outras.

Enquanto que estas abordagens de composição se baseiam apenas nos construtos das linguagens de modelagem para realizar a integração entre modelos, entre opiniões ou entre RFs e RNFs; elas não prevêm a necessidade de compor características transversais aos construtos das linguagens. Além disto, estas abordagens são primordialmente ou de separação, ou de composição. Elas não abrangem as duas atividades porque a separação não prevê a descrição de como compor os modelos separados.

Com o crescimento do interesse pelo desenvolvimento orientado a aspectos, têm surgido abordagens de separação que se preocupam, também, com a composição. É possível descrever características transversais separadamente, porque há mecanismos para juntá-las (Baniassad, 2004; Moreira, 2005; Rashid, 2002; Sousa, 2003, 2004a, 2004b). Entretanto, estes mecanismos não são automatizados (Baniassad, 2004; Sousa, 2003, 2004), são restritos a integração de RNFs (Rashid, 2002), não permitem a visualização separada das características após a composição, ou são específicos para um único tipo de linguagem de modelagem (Baniassad, 2004; Moreira, 2005; Rashid, 2002; Sousa, 2003, 2004a, 2004b).

Nesta tese, mostramos a necessidade de trabalhos que envolvam a separação, a composição e a visualização como uma abordagem mínima para modelagem de requisitos. Nossa abordagem consiste em um modelo para integração de características transversais, que pode ser utilizado por diferentes linguagens de modelagem de requisitos para abordar os problemas de espalhamento e entrelaçamento de características transversais.

1.3. Solução Proposta

Para abordar os problemas de espalhamento e entrelaçamento de características durante a definição de requisitos, propomos um metamodelo de integração de características transversais. Este modelo é fundamentado nas idéias implementadas nas linguagens de programação orientadas a “aspectos”: AspectJ (Kiczales, 2001) e Hyper/J (Tarr, 2000).

Assim como nestas linguagens, o modelo de integração dá apoio à separação e composição de características transversais, por prover elementos para modelagem de características separadamente e um mecanismo de composição

automático que facilita a inclusão e a exclusão de características da especificação. Além disto, o modelo de integração dá apoio à extração de diferentes visões (visualização) dos modelos separados e compostos. Nesta etapa do desenvolvimento, a visualização é essencial porque facilita a análise e especificação do problema, bem como a definição do sistema, veja Figura 2.

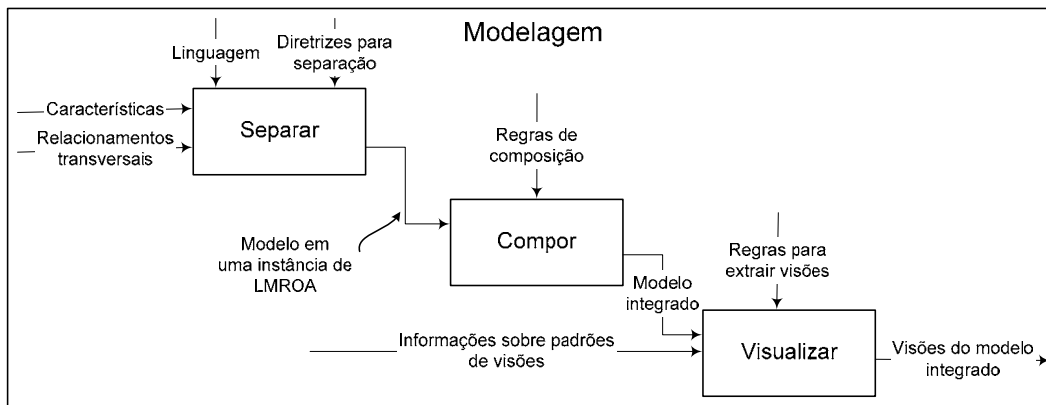


Figura 2. Metamodelo de integração de características transversais

Para realizar a separação e a composição, definimos uma linguagem de modelagem de requisitos orientada a “aspectos” (LMROA). LMROA oferece um novo tipo de relacionamento, denominado “relacionamento transversal”, para estender linguagens de modelagem de requisitos. O relacionamento transversal é utilizado para registrar a interação entre características, indicando que elas se espalham e/ou se entrelaçam às demais e registrando como tais características influenciam ou modificam umas as outras.

A composição é realizada por um mecanismo que processa as informações contidas no relacionamento transversal, propagando as modificações nele descritas. Cada construto do relacionamento transversal indica uma maneira de modificar o modelo e também novas características, por meio das quais pode-se extrair visões. A visualização é responsável pela geração de diferentes visões do modelo integrado, permitindo que o engenheiro de requisitos possa analisar partes ou a modelagem inteira, sob diferentes perspectivas.

Os mecanismos de composição e visualização dependem da linguagem de modelagem adotada. Desta forma, aplicamos a extensão definida por LMROA ao modelo de metas V-graph (Yu, 2004), e assim, instanciamos os mecanismos de composição e visualização.

1.4. Contribuições da Tese

Por abordar os problemas de entrelaçamento e espalhamento em modelos de requisitos, nossa abordagem provê facilidades para que os engenheiros de requisitos modelem características transversais e suas interações, e entendam, analisem, modifiquem e reusem partes do modelo de requisitos. Dentre os produtos gerados estão: metamodelo para integração de características transversais; linguagem de modelagem de requisitos orientada a “aspectos” (LMROA); extensão do modelo de metas V-graph; regras para extração de um conjunto de visões que podem ser geradas a partir do V-graph; e um conjunto preliminar de características transversais reusáveis.

Além disso, consideramos que nossa abordagem provê a infra-estrutura básica para pesquisarmos a natureza de características transversais, além daquelas originadas em requisitos não funcionais. Desta forma, contribuímos para o desenvolvimento orientado a aspectos. No Capítulo 6, detalhamos como estas contribuições são alcançadas em nossa abordagem.

1.5. Organização da Tese

O restante desta tese está organizado da seguinte maneira. No **Capítulo 2** apresentamos os conceitos de engenharia de requisitos necessários para o entendimento do contexto e problema abordados nesta tese. No **Capítulo 3**, relatamos os fundamentos da separação de características, soluções utilizadas no nível de implementação e abordagens em engenharia de requisitos orientada a aspectos, que contribuíram para a criação de nossa estratégia.

No **Capítulo 4**, descrevemos nossa estratégia para os problemas de entrelaçamento e espalhamento de características transversais durante a modelagem de requisitos – um metamodelo para integração de características transversais. Detalhamos a utilização deste modelo com o V-graph, mostrando os benefícios desta abordagem e sua implementação. Além disto, apresentamos como esta estratégia pode ser utilizada com outras linguagens de modelagem e qual o seu impacto no processo de engenharia de requisitos como um todo.

No **Capítulo 5**, apresentamos os estudos de caso realizados para demonstrar e validar nossa abordagem. Finalmente, no **Capítulo 6**, apresentamos as contribuições e limitações de nossa abordagem, uma comparação com os trabalhos relacionados, bem como os trabalhos futuros e as tendências apontadas pelo tema desta tese.

No Apêndice A, apresentamos a linguagem de modelagem de requisitos orientada a aspectos, utilizando os padrões BNF e DTD. No Apêndice B, apresentamos a implementação dos componentes do mecanismo de composição e visualização. Nos Apêndices C e D, resumimos a modelagem e algumas visões de nossos estudos de caso. No Anexo A, apresentamos o documento de requisitos utilizado em um dos estudos de caso.