

6 Conclusão

Neste Capítulo apresentamos um resumo de nossa abordagem (Seção 6.1), bem como suas contribuições (Seção 6.2) e diferenças das abordagens relacionadas (Seção 6.3). Em seguida, apresentamos suas limitações e os trabalhos futuros (Seção 6.4).

6.1. Resumo

Esta tese aborda os problemas de entrelaçamento e espalhamento de características transversais durante a modelagem de requisitos.

Fomos influenciados por abordagens de programação orientadas a aspectos na definição de uma estratégia de modelagem que considera atividades e mecanismos de separação e composição de características transversais. Por outro lado, considerando abordagens utilizadas em engenharia de requisitos para gerenciar a complexidade de modelos, adotamos como essencial um mecanismo de visualização.

Assim, nossa estratégia consiste de mecanismos para separação, composição e visualização de características. Estes mecanismos funcionam com base em uma linguagem de modelagem (LMROA) que permite a descrição antecipada de como as características estão entrelaçadas e espalhadas. LMROA é um modelo conceitual que define um novo tipo de relacionamento, denominado relacionamento transversal, para ser adicionado a linguagens tradicionais de modelagem de requisitos.

Para validar nossa estratégia, instanciamos LMROA por utilizar o V-graph como modelo de componentes e definimos (e implementamos) os mecanismos de separação, composição e visualização para esta instância. Além disto, dois estudos de caso foram realizados, ilustrando os ganhos obtidos por aplicar uma abordagem orientada a aspectos durante a definição de requisitos.

6.2. Contribuições

Esta tese mostra os problemas de espalhamento e entrelaçamento em modelos de requisitos. Estes problemas aumentam a complexidade dos modelos, dificultando a modelagem, rastreabilidade, evolução e reuso de requisitos. Assim, nós exploramos as vantagens de explicitar e tratar características transversais durante a modelagem. Como citamos no Capítulo 1, as contribuições desta tese são:

1) **Metamodelo de integração de características transversais** – O metamodelo define as atividades de separação, composição e visualização como sendo essenciais para abordar os problemas de espalhamento e entrelaçamento de características durante a modelagem de requisitos. Este metamodelo pode ser utilizado como base em outras fases do desenvolvimento para definição de mecanismos práticos de redução dos danos causados pelos problemas de espalhamento e entrelaçamento. O metamodelo é apresentado no Capítulo 4, Seção 4.1.

2) **Linguagem de modelagem de requisitos orientada a aspectos** – Em LMROA definimos o “relacionamento transversal”, como um novo construto para estender linguagens tradicionais de modelagem de requisitos. Diferentemente das abordagens orientadas a aspectos, nossa abordagem define um relacionamento como o elemento responsável por descrever como as características se entrelaçam e se espalham. Desta forma, como relatamos no Capítulo 4, Seção 4.2.3, todas as características, transversais ou não, são descritas utilizando apenas os construtos do modelo de componentes escolhidos. Assim, a extensão realizada é menos intrusiva à linguagem de modelagem de requisitos escolhida; retardamos a decisão e descrição das informações relativas à transversalidade de requisitos; facilitamos o reuso de características que em alguns projetos têm esta propriedade de transversalidade e em outros não a têm; separamos as informações referentes a “o quê” é transversal, representada pelos elementos do modelo de componentes, das referentes a “como” é transversal, representadas no relacionamento transversal. LMROA é descrita no Capítulo 4, Seção 4.2

3) **Criação de AOV-graph** – AOV-graph é uma extensão de V-graph que permite explicitar como as características estão transversais. Definimos diretrizes

para separação de características, semântica para os construtos de LMROA, e visões essenciais para a análise e evolução destes modelos de metas. Esta nova linguagem permite, além da descrição e análise de RNFs e RFs, a descrição e análise de características transversais. Por meio do relacionamento transversal é possível reduzir o número de relacionamentos entre metas, *softmetas* e tarefas e assim, diminuir o acoplamento entre estes elementos, ou pelo menos, facilitar a visualização e manipulação do modelo. AOV-graph é apresentado no Capítulo 4, Seção 4.3.

4) **Geração das visões Cenários, MER e diagrama de classes a partir de informações descritas no modelo AOV-graph** – O modelo de metas AOV-Graph é uma visão intencional do sistema. Porém, as informações retratadas em seus construtos (*softmetas*, metas, tarefas, tipo, tópico e relacionamentos) podem ser decompostas e representadas de outras formas, tais como: cenários, MER e diagrama de classes. O mecanismo de visualização que definimos para AOV-graph inclui estas formas de organizar as informações do sistema. Para isto, elaboramos regras que transformam as informações em AOV-graph no modelo de cenários, MER e diagrama de classes. As regras para gerar estas visões são descritas no Capítulo 4, Seção 4.3.3.

5) **Conjunto de características transversais reutilizáveis** – Realizamos dois estudos de casos utilizando AOV-graph. Nestes estudos de caso, modelamos separadamente as características Segurança, Persistência, Usabilidade e Confiabilidade. Estes modelos podem ser reutilizados em outros projetos, ou pelo menos, usados como ponto de partida para melhor elaboração destas características. Os relacionamentos transversais contidos neles precisam ser modificados para que se relacionem aos novos pontos afetados. Contudo, algumas informações contidas em cada *intertype* e *advice* podem ser reaproveitadas.

Estas contribuições estão correlacionadas ao reuso, rastreabilidade, evolução e ao desenvolvimento orientado a aspectos da seguinte maneira:

1) **Reuso** – é melhorado porque cada característica pode ser modelada separadamente, muitas vezes, de maneira mais abstrata, com o intuito antecipado de utilizá-la em diferentes projetos;

2) **Rastreabilidade** – parte do rastreamento é realizada antecipadamente, enquanto o engenheiro modela as características usando o relacionamento transversal, e não após elas terem sido modeladas, como acontece usualmente;

3) **Evolução** – a evolução é afetada por três razões: i) devido a separação, cada modelo é menos complexo, então é mais fácil modificar cada um separadamente; ii) devido ao uso do relacionamento transversal, com a centralização de informações sobre o entrelaçamento e espalhamento é mais fácil controlar e modificar as interações que envolvem características transversais, e iii) com os mecanismos de composição e visualização qualquer mudança nos relacionamentos transversais é automaticamente propagada pelos modelos (visões) do sistema;

4) **Desenvolvimento de software orientado a aspectos** – com a utilização de instâncias de LMROA em projetos reais, tal como fizemos com AOV-graph, é possível analisar se as características modeladas como transversais (aquelas que são origem de relacionamentos transversais) durante a definição de requisitos são também transversais nas fases posteriores de desenvolvimento e, assim, ter maior conhecimento sobre “o que são aspectos” nesta fase. Desta forma, consideramos que nossa abordagem pode ajudar a abordagens de identificação, mapeamento e análise de características transversais.

6.3. Comparação com Trabalhos Relacionados

Na Tabela 11, listamos os principais trabalhos relacionados ao nosso (descritos nos Capítulos 2 e 3). Estes trabalhos têm em comum ao nosso a preocupação explícita com a separação e composição de características transversais durante a definição de requisitos.

Tabela 10. Trabalhos relacionados

Referência	Título	Foco
Rashid, 2002 (Capítulo 3)	Early aspects: a model for aspect-oriented requirements engineering	Processo
Moreira, 2002 (Capítulo 3)	Crosscutting quality attributes for requirements engineering	Processo
Rashid, 2003 (Capítulo 3)	Modularization and composition of aspectual requirements	Processo - identificação, modelagem, e mapeamento
Sousa, 2003 (Capítulo 3)	Adapting the NFR framework to aspect-oriented requirements engineering	Processo – modelagem
Sousa, 2004 (Capítulo 3)	Separation of crosscutting concerns from requirements to design: Adapting the use case driven approach	Processo – modelagem
Brito, 2004 (Capítulo 3)	Integrating the NFR framework in a RE model	Processo – identificação
Moreira, 2005 (Capítulo 3)	Multi-Dimensional Separation of Concerns in Requirements Engineering	Modelagem
Leite, 2005 (Capítulo 2)	Quality-Based Software Reuse	Reuso e modelagem

Em resumo, nossa abordagem se distingue destes trabalhos tanto na motivação quanto na maneira de realizar a integração de características transversais:

Quanto à motivação

Estes trabalhos tentam, durante a definição de requisitos, identificar características que **mais tarde** possam ser aspectos, i.e., **mais tarde** elas estarão espalhadas e entrelaçadas; enquanto nossa abordagem tenta diminuir o entrelaçamento e espalhamento **durante** a modelagem de requisitos. Isto significa que estas abordagens buscam favorecer primeiramente ao desenvolvimento orientado a aspectos, enquanto nós buscamos primeiramente ajudar a Engenharia de Requisitos, considerando que todas as atividades (elicitação, modelagem, análise, reuso, evolução, rastreamento) durante esta etapa se beneficiam da abordagem orientada a aspectos.

Quanto ao modelo de integração

Nenhum dos trabalhos relacionados se preocupa em oferecer diferentes visões do modelo integrado. Desta forma, após a composição, os modelos são mais complexos e, assim, é mais difícil analisá-los e modificá-los. Por outro lado, nossa abordagem é restrita a modelagem, não se preocupando com a identificação e mapeamento de características transversais, estas atividades estão incluídas em nossos trabalhos futuros.

Quanto à linguagem de modelagem

Na Tabela 11, resumimos a comparação entre nossa linguagem de modelagem e as demais, utilizando as propriedades definidas em (Chavez, 2004). As colunas denominadas Q7, Sousa, Rashid e Moreira se referem às linguagens definidas, respectivamente, em (Leite, 2004; Sousa, 2004; Rashid, 2003; e Moreira, 2005). Nos concentramos nestes trabalhos porque eles abordam a modelagem de maneira mais concreta.

1) Modelo de componentes – LMROA pode ser aplicada a diferentes modelos de componentes. AOV-graph é uma instância de LMROA que utiliza um modelo de metas como modelo de componentes, denominado V-graph, seus componentes são: metas, *softmetas* e tarefas. Em Q7, Sousa, Rashid e Moreira, as estratégias para separação e composição são específicas, respectivamente, para os modelos de componentes: V-graph, casos de uso, sentenças de requisitos em pontos de vista e sentenças de requisitos. Além disto, Sousa e Rashid incluem extensões para

representar de maneira diferenciada os elementos transversais, denominadas casos de uso transversais e *concerns*, respectivamente.

Tabela 11. Comparação de linguagens para integração de características transversais

Características	LMROA	AOV-graph	Q7	Sousa	Rashid	Moreira
Modelo de componentes	Qualquer	Goal model	Goal model	Casos de uso e casos de usos transversais (RNFs)	Baseado na decomposição de pontos de vista e <i>concerns</i> (RNFs) em requisitos	Baseado na decomposição de <i>concerns</i> em requisitos
Componente	Qualquer	Tarefa, meta, <i>softmeta</i>	Tarefa, meta, <i>softmeta</i>	Casos de uso	Pontos de vista, <i>concerns</i> e requisitos	<i>Concerns</i> e requisitos
Linguagem de componentes	Qualquer	V-graph	V-graph	Diagrama de Casos de uso	Sentenças de requisitos	Sentenças de requisitos
Modelo de <i>joinpoint</i>						
<i>Joinpoints</i> dinâmicos	-	-	-	-	-	-
Joinpoints estáticos	Depende do modelo de componentes	Tarefas, metas, <i>softmetas</i> , tipos, tópicos	Tarefas, metas e <i>softmetas</i>	Casos de uso	Sentenças de requisitos nos pontos de vista	Sentenças de requisitos
Modelo núcleo						
Aspecto	Relacionamento transversal	Relacionamento transversal	<i>Template</i> 5w2h	Template para composição	Template pra composição	<i>Template</i> para composição
Interface transversal	Origem e <i>pointcuts</i>	Tarefas, metas, <i>softmetas</i> Expressões regulares	Where: meta, <i>softmeta</i> e tarefa	caso de uso e ação afetada	requisito no ponto de vista afetado	requisito que é afetado
<i>Feature</i> transversal	<i>Advice</i> , <i>intertype declaration</i>	Tarefas, metas, <i>softmetas</i> e novos atributos ou elementos	What: meta, <i>softmeta</i> e tarefa	Ações do caso de uso transversal	Requirements no <i>concern</i>	Requirements no <i>concern</i>
Modelo de composição	Estático	Estático	Estático	Estático	Estático	Estático
Dicotomia	Não	Não	Não	Não	Não	Não
Quantificação	Sim	Sim	Não	Não	Não	Não
Transparência	Sim	Sim	Sim	Sim	Sim	Sim

2) Modelo de *joinpoint* – em todas as abordagens apenas *joinpoints* estáticos são definidos. LMROA define explicitamente quais são os pontos a serem

afetados, através do elemento *joinpoint*. No caso de AOV-graph, *softmetas*, metas e tarefas podem afetar *softmetas*, metas, tarefas, tipos e tópicos, permitindo a separação e composição de RNFs, RFs, dados e ações. Em Q7, apenas *softmetas*, metas e tarefas podem ser afetadas. Em Sousa e Rashid, apenas RNFs podem afetar RFs, sendo que em Rashid, RFs estão relacionados a pontos de vista. Assim como em nossa abordagem, em Moreira, tanto RFs quanto RNFs podem afetar e/ou ser afetados.

3) Modelo núcleo – em LMROA definimos o relacionamento transversal como o relacionamento responsável por registrar a transversalidade dos componentes. Este relacionamento pode ser aplicado a diferentes modelos de componentes. Sua interface transversal é descrita pelos *pointcuts* e as *features* pelas *intertype declarations* e *advice*. Em AOV-graph, os *pointcuts* podem ser qualquer elemento dos tipos meta, *softmeta*, tarefas, tipos e tópicos, e as *features* podem ser elementos destes mesmos tipos ou novos tipos descritos nas *intertype declarations*. Em Q7, as interfaces e *features* transversais podem ser elementos dos tipos, meta, *softmeta* e tarefa. Em Sousa, as interfaces são os identificadores dos casos de uso mais a ação afetada nele, e as *features* são definidas por ações do caso de uso transversal. Em Rashid e Moreira, as interfaces e *features* transversais são definidas apenas por elementos do tipo sentenças de requisitos.

4) Modelo de composição – Em todas as abordagens o modelo de composição: gera um novo modelo (seguindo o mesmo formato do modelo de componentes) com as informações integradas; a composição ocorre diretamente na descrição “estática” de requisitos. Em AOV-graph, definimos um modelo de composição automatizado, que permite o uso de expressões regulares para identificação dos pontos a serem atingidos. Em Moreira e Rashid o modelo de composição também é automático. Enquanto Q7 e Sousa não disponibilizam um modelo de composição automatizado. Em Sousa as informações sobre a composição são descritas informalmente, servindo apenas como guia para composição.

5) Dicotomia – Nenhuma das abordagens modularizam características transversais em um único elemento. Elas utilizam elementos do modelo de componentes para descrever as características e elementos de composição para descrever o entrelaçamento. Em AOV-graph e Q7 as características são definidas pelos próprios elementos do modelo de componentes (metas, *softmetas* e tarefas) e a transversalidade pelo relacionamento transversal e 5W2H, respectivamente. Em

Sousa as características transversais são definidas por casos de uso de RNFs transversais (casos de uso transversais) e a transversalidade por um template de composição. Em Rashid e Moreira, as características são representadas pelo template de *concerns* e a transversalidade por templates de composição.

6) Quantificação – Nossa abordagem permite a quantificação através de expressões regulares. Em Rashid e Moreira é possível especificar que um ou todos os requisitos de uma característica corta todos ou um requisito de uma outra característica, mas para especificar mais de um requisito (que não sejam todos) é necessário definir um a um. Em Sousa e Q7, a quantificação também não é abordada.

7) Transparência – Todas as abordagens assumem que não é necessário realizar nenhuma mudança na descrição dos elementos afetados.

Como podemos observar na Tabela, LMROA é um **modelo genérico** que pode ser utilizado para estender linguagens de modelagem de requisitos, provendo o tratamento de características transversais. Apesar de AOV-graph e Q7 usarem o mesmo modelo de componentes, AOV-graph permite a criação de novos elementos e a modificação de tipos e tópicos além dos permitidos em Q7. Com a nossa abordagem o modelo de componentes é estendido apenas pelo relacionamento transversal, enquanto as abordagens definidas em (Sousa, 2004; Rashid, 2003) criam extensões para representar tanto as características transversais em si (RNFs), quanto para representar a transversalidade destas características. A abordagem descrita em (Moreira, 2005) utiliza um template de composição para relacionar quaisquer requisitos, enquanto em AOV-graph podemos utilizar, também, as correlações e contribuições de seu modelo de componentes. Além disto, em AOV-graph os elementos metas, *softmetas*, tarefas, tipos e tópicos possibilitam mais formas de separação e composição do que apenas sentenças não estruturadas de requisitos, como utilizado em (Moreira, 2005).

6.4.

Limitações e Trabalhos Futuros

A seguir, citamos as limitações de nossa abordagem e perspectivas para trabalhos futuros:

Identificação e mapeamento de características transversais

O escopo desta tese é a modelagem de requisitos. Desta forma, as atividades de identificação e mapeamento de características transversais não são abordadas. Consideramos que estas atividades são independentes do meta-modelo de integração e do relacionamento transversal de LMROA. Elas são influenciadas pelo modelo de componentes utilizado. Desta forma, para cada instância de LMROA, heurísticas podem ser definidas para a identificação, modelagem e mapeamento ou adaptadas de outras abordagens, tais como (Sousa, 2004; Rashid, 2003).

Análise

Consideramos que nossa abordagem ajuda a atividade de análise de requisitos porque oferece visões, parciais ou não, mostrando diferentes propriedades do sistema modelado. Entretanto, por meio dos estudos de caso realizados, nos limitamos à análise de corretude e identificação de requisitos omissos. Entretanto, seria interessante agregar outros tipos de análise à nossa abordagem, tais como, análise de obstáculos (Lamsweerde, 2000); análise quantitativa de metas (Giorgini, 2002); análise de variabilidade (González, 2004; Park, 2004), dentre outras.

Apoio de ferramentas

Uma das principais limitações de abordagens de engenharia de requisitos orientada a aspectos é quanto ao apoio de ferramentas. Como relatamos no Capítulo 3, Seção 3.4, muitas das abordagens não apresentam suporte automatizado para realizar a composição. Nossa abordagem baseia-se em uma linguagem que permite a automação desta atividade, bem como a geração de visões. Entretanto, como apresentamos no Capítulo 4, Seção 4.3.4, nossa implementação não inclui ferramentas para edição gráfica de modelos em AOV-graph, nem interatividade com as visões geradas, de maneira que o engenheiro conseguisse minimizar, ocultar, realçar ou mesmo modificar os elementos de interesse a cada momento do processo de definição de requisitos.

Regras de transformação

Quanto às visões geradas com base em AOV-graph, as regras de transformação definidas no Capítulo 4, Seção 4.3.3.2, podem ser refinadas para obtenção de modelos mais precisos, tal como a inclusão de cardinalidade no MER e diagrama de classes, inferência do tipo de relacionamento no diagrama de classes, dentre outros.

Integração de métodos

Se as visões geradas pudessem ser modificadas diretamente, então um metamodelo definindo duplo fluxo de transformações é necessário. Pesquisas sobre a adequação ou adaptação do relacionamento transversal quando trabalhando com múltiplos modelos pode ser inspirada nas idéias de Nuseibeh (1994) ou Cysneiros (2001). Isto ampliaria nossa abordagem por incluir, também, a integração de métodos.

Armazenamento e recuperação de características

Como mostrado anteriormente, as características (ou serviços) podem ser separadas e reutilizadas em diferentes projetos. Entretanto, não discutimos como elas podem ser armazenadas e recuperadas. Em (Leite, 2004), define-se uma abordagem interessante para o reuso de características transversais baseada na linguagem Q7. Podemos unir esta abordagem à nossa, realizando uma transformação dos elementos descritos nos relacionamentos transversais para o formato Q7. Isto tornaria as informações em Q7 mais precisas e proveria nossa abordagem de um mecanismo de organização e recuperação para reuso.

Extensão da linguagem de modelagem

Definimos LMROA de forma genérica para que ela possa ser adaptada a modelos de componentes diferentes, tal como apresentamos no Capítulo 4, Seção 4.4. Acreditamos que esta descrição possa ser facilmente estendida caso haja necessidade. Por exemplo, em descrever qual a seqüência das composições, ou outras primitivas além de *include* e *substitute*.

Experimentação

Demonstramos através dos estudos de caso que nossa abordagem contribui positivamente para o reuso, rastreabilidade e evolução. Porém, mais estudos precisam ser realizados para obtermos mais conhecimento sobre o uso desta estratégia. A realização de experimentos utilizando nossa abordagem e abordagens mais tradicionais em paralelo, com métricas bem definidas, é importante para avaliarmos os ganhos de produtividade com o uso de nossa estratégia.