



Universidade Federal de Pernambuco
Centro de Informática
Departamento de Sistemas de Computação

Graduação em Ciência da Computação

**AUTOMAÇÃO DO PROCESSO DE IDENTIFICAÇÃO DE
ASPECTOS EM MODELOS I***

Cleviton Vinicius Fonseca Monteiro

PROPOSTA DE TRABALHO DE GRADUAÇÃO

Orientador: *Jaelson Brelaz de Castro*

Co-orientadora: *Fernanda Maria Ribeiro de Alencar*

| |
|--|
| Proposta de Trabalho de Graduação |
| Automação do Processo de Identificação de Aspectos em Modelos i* |

SUMÁRIO

| | |
|--|---|
| 1. Contexto | 2 |
| 2. Integração de Aspectos nos Modelos i* | 4 |
| 3. Objetivo do Trabalho Proposto | 6 |
| 4. Cronograma | 7 |
| Referências | 9 |

LISTA DE FIGURAS

| | |
|--|---|
| Figura 1 - Exemplo de Modelo i* | 4 |
| Figura 2 - Representação Expandida de um Candidato a Aspecto em i* | 5 |
| Figura 3 - Diagrama Resultante da Aplicação das Guidelines | 5 |

LISTA DE TABELAS

| | |
|--|---|
| Tabela 1- Cronograma de Atividades | 7 |
|--|---|

1. CONTEXTO

A constante inovação dos paradigmas de programação tem como objetivo tornar a engenharia de software cada vez mais gerenciável, permitindo que aplicações com alto nível de complexidade sejam desenvolvidas com sucesso. Seguindo esse contexto o paradigma de orientação a agentes surgiu como uma nova maneira de analisar, projetar e implementar sistemas de software complexos. Porém para utilizá-la largamente em aplicações comerciais e industriais faz-se necessário seguir um processo de engenharia de software que suporte esse novo paradigma. Então, para lidar com esse problema novas metodologias baseadas nos conceitos de agentes foram propostas [16, 17, 18], em particular pode-se destacar o Tropos [11, 12]. Tropos é uma metodologia fundamentada em requisitos que suporta engenharia de sistemas multi-agentes e visa tornar o paradigma de orientação a agentes uma tecnologia de ampla utilização em sistemas comerciais e industriais.

O Tropos tem o intuito de suportar também as fases iniciais do ciclo de vida do desenvolvimento de software que operam em ambientes dinâmicos, para isso é baseado no *framework* de engenharia de requisitos chamado i^* [6]. Os modelos i^* são utilizados para promover o melhor entendimento do ambiente organizacional e seus objetivos. Tais modelos incluem conceitos como atores (que podem ser agentes, posições, papéis), dependências entre eles, as quais incluem objetivos (metas), tarefas, recursos e objetivos soft (requisitos não-funcionais).

Por outro lado os mecanismos providos pelos paradigmas de desenvolvimento de software (incluindo a orientação a agentes) não são suficientes para lidar com a natureza transversal (*crosscutting concern*)[19] de alguns interesses. Esse fato torna mais difícil o reuso e manutenção de artefatos de software. Esses interesses transversais são responsáveis pela produção de representações (especificações e implementações) espalhadas e entrelaçadas, por conseguinte mais difíceis de serem entendidas e mantidas. Surge o paradigma de orientação a aspectos [8] com o propósito de prover mecanismos de modularização e composição de interesses transversais em módulos separados, os chamados aspectos. Em nível de código os aspectos podem ser entendidos como sendo trechos de código implementados separadamente das classes da orientação a objetos. Esses trechos de códigos são combinados nas classes em tempo de compilação ou de execução. Portanto os interesses transversais podem ficar modularizados nos aspectos e os demais nas classes da orientação a objetos.

Até pouco tempo atrás a maioria dos trabalhos relacionados com aspectos são limitados à fase de implementação do ciclo de desenvolvimento de software. Porém pesquisas recentes tentam generalizar o conceito de aspectos e aplicá-lo a outras fases do ciclo de vida. Em particular, *Early Aspect* [20] objetiva que interesses transversais sejam identificados em fases iniciais do desenvolvimento de software, como na análise de domínio e análise de requisitos. Sendo assim, visando aproveitar as vantagens desse paradigma de orientação a aspectos desde fases iniciais do processo de

desenvolvimento, várias abordagens de engenharia de requisitos têm sido integradas com aspectos [1,3,13,14,15]. Em particular há uma abordagem que objetiva lidar com interesses transversais em fases iniciais da metodologia Tropos [4,5,7]. A mesma propõe um conjunto de regras (guidelines) para identificar sistematicamente interesses que são candidatos a aspectos em modelos i^* . Essa abordagem servirá de base ao trabalho de graduação proposto.

2. INTEGRAÇÃO DE ASPECTOS NOS MODELOS i^*

Os modelos i^* são bastante interessantes para entender os objetivos presentes em uma organização e suas razões. Porém, os mesmos ficam bastante grandes e complexos à medida que vão sendo detalhados e em muitos casos várias dependências entre os atores se repetem, deixando o diagrama disperso e difícil de manter, portanto perdendo em modularidade.

Um exemplo de diagrama i^* pode ser visto a seguir:

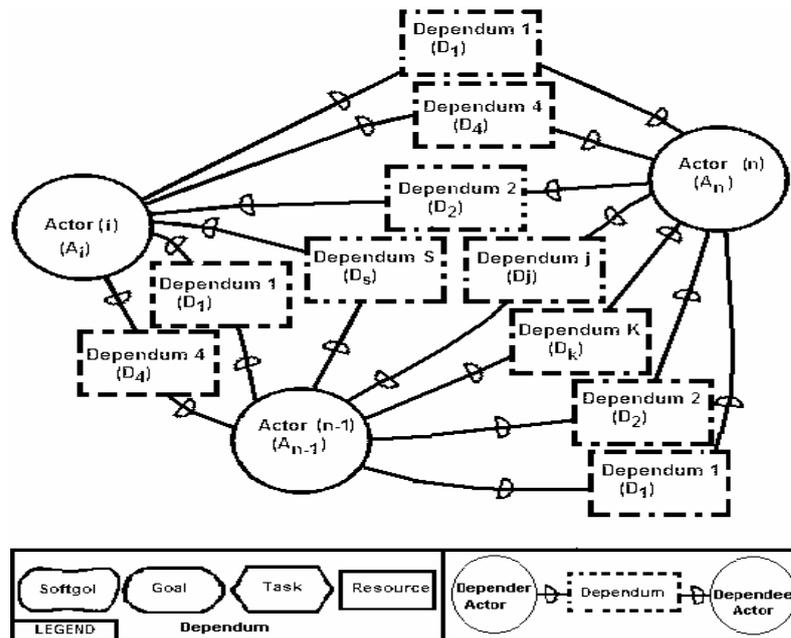


Figura 1 - Exemplo de Modelo i^*

Então, para evitar os problemas citados anteriormente, há uma abordagem que tem como objetivo integrar os princípios da orientação a aspectos aos modelos i^* identificando candidatos a aspectos nos mesmos. A identificação é feita a partir de guidelines que são aplicadas sistematicamente ao diagrama i^* .

As guidelines propostas são:

Guideline 1 (no modo SD): Um dependum é candidato a aspecto quando é requerido por pelo menos dois dependees (atores), assumindo que sua operacionalização é a mesma para todos os atores e ele não é provido por um único ator.

Guideline 2 (no modelo SR): Uma tarefa é candidata a aspecto se ela está envolvida em uma relação de decomposição, direta ou indiretamente relacionada a uma dependência externa, e é requerida por duas ou mais tarefas que são também relacionadas a outra dependência externa. Note que uma tarefa é

indiretamente relacionada a uma dependência externa se na hierarquia que ela está presente pelo menos um dos seus antecessores está conectado a uma dependência externa.

Guideline 3 (remover redundâncias): Os candidatos a aspectos identificados pela guideline 2 que são operacionalizações de candidatos a aspectos já identificados pela guideline 1 devem ser agrupados em apenas um candidato a aspecto.

A figura a seguir mostra o estereótipo utilizado para representar um candidato a aspecto nos modelos i^* . Note que a estrela pode ter sua fronteira expandida e seu interior contém o elemento de dependência que gerou o aspecto e a regra de composição do aspecto com os atores envolvidos.

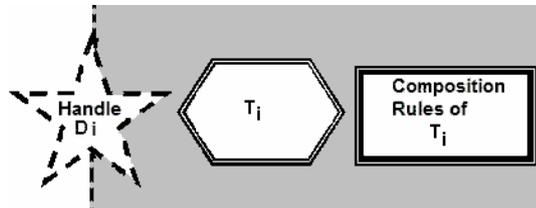


Figura 2 – Representação Expandida de um Candidato a Aspecto em i^*

Após a aplicação das guidelines ao modelo mostrado na figura 1, é obtido o diagrama a seguir:

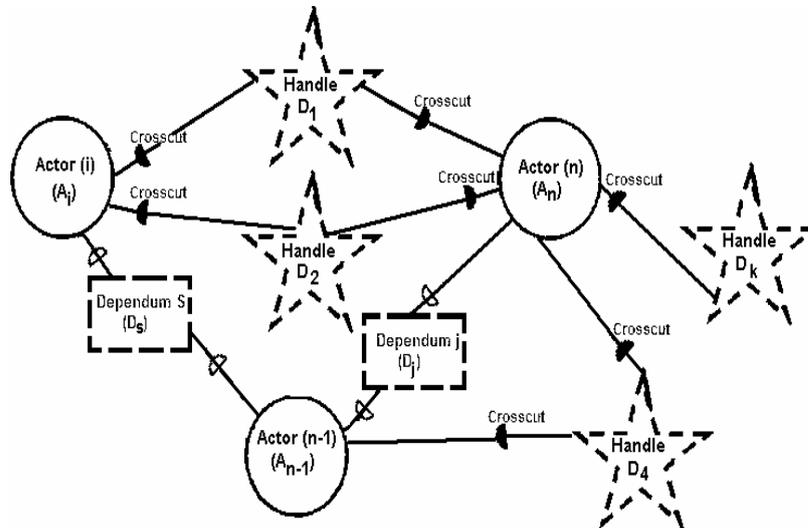


Figura 3 - Diagrama Resultante da Aplicação das Guidelines

3. OBJETIVO DO TRABALHO PROPOSTO

O objetivo do trabalho proposto é estender a ferramenta de modelagem i* chamada OME [9] para que seja possível identificar automaticamente candidatos a aspectos nos modelos i*. Para isso será feita a extensão de um *framework* utilizado pela OME assim como o desenvolvimento de um *plugin* para a mesma. A extensão do *framework* será necessária pelo fato de que será preciso adicionar um novo elemento aos modelos i*, o aspecto. Por sua vez, o *plugin* será implementado para automatizar a identificação de candidatos a aspectos nos modelos seguindo as guidelines propostas em [4,5].

A importância da criação dessa extensão da ferramenta OME deve-se principalmente a três fatores:

- A possibilidade de criar diagramas i* com os estereótipos aspectuais. Como já foi dito diversas pesquisas almejam utilizar a orientação a aspectos desde fases iniciais do ciclo de desenvolvimento de software, sendo assim, o suporte dado por uma ferramenta vem a ser um facilitador interessante aos estudos.
- A comprovação da possibilidade de automação das guidelines propostas pela abordagem [4,5].
- Maior facilidade e agilidade na identificação de candidatos a aspectos nos modelos i*.

Também fará parte do escopo do trabalho, a validação da extensão criada. Isso será feito para verificar se a mesma foi capaz de atender satisfatoriamente a todos os requisitos necessários.

4. CRONOGRAMA

A seguir, um cronograma com os marcos das atividades chaves desse trabalho, durante as semanas dos meses de Novembro/2006 até Março/2007, logo em seguida também será apresentada uma breve descrição sobre cada uma destas atividades.

Tabela 1-Cronograma de Atividades

| Atividade | Mês | | | | | | | | | | | | | | |
|--|-------|---|---|-------|---|---|-------|---|---|-------|---|---|-------|---|---|
| | 11/06 | | | 12/06 | | | 01/07 | | | 02/07 | | | 03/07 | | |
| 1. Pesquisa bibliográfica sobre propostas de incorporação de aspectos em modelos i*. | ■ | ■ | ■ | | | | | | | | | | | | |
| 2. Pesquisa sobre ferramentas para construção de modelos i*. | ■ | ■ | ■ | | | | | | | | | | | | |
| 3. Elicitação e especificação dos requisitos da extensão. | | | | ■ | ■ | ■ | ■ | | | | | | | | |
| 4. Análise da arquitetura e definição das classes de projeto da extensão. | | | | | ■ | ■ | ■ | ■ | | | | | | | |
| 5. Codificação e testes da extensão. | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | | | | |
| 6. Validação da extensão. | | | | | | | | | | | ■ | ■ | ■ | | |
| 7. Escrita do trabalho de graduação. | | | | | | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | ■ | |
| 8. Preparação da apresentação oral. | | | | | | | | | | | | | | | ■ |

1. Pesquisa bibliográfica sobre propostas de incorporação de aspectos em modelos i*

Levantamento e análise das propostas acadêmicas de identificação e modelagem de *Early aspects* em modelos i*.

2. Pesquisa sobre ferramentas para construção de modelos i*

Levantamento das ferramentas existentes para criação de modelos i* [2,9,10] e análise dos pontos relevantes de cada uma quanto à possibilidade de extensão.

3. Elicitação e especificação dos requisitos da extensão.

Captação e especificação dos requisitos a serem adicionados às ferramentas existentes para que seja possível a automação das propostas de identificação de aspectos nos modelos i*.

4. Análise da arquitetura e definição das classes de projeto da extensão

Estudo da arquitetura da extensão e posterior definição das classes de projeto do mesmo, com elaboração do diagrama de classes.

| |
|--|
| Proposta de Trabalho de Graduação |
| Automação do Processo de Identificação de Aspectos em Modelos i* |

5. Codificação e testes da extensão

Elaboração do código-fonte da extensão a ser incorporado na ferramenta de modelagem i* e testes do mesmo.

6. Validação da extensão

O objetivo dessa atividade é verificar se a extensão possui e atende satisfatoriamente aos requisitos especificados.

7. Escrita do trabalho de graduação

Elaboração do trabalho escrito que irá conter a definição da proposta de trabalho, os artefatos gerados pelo trabalho e os resultados do mesmo.

8. Preparação da apresentação oral

Elaboração da apresentação oral que irá mostrar todos os pontos relevantes relacionados ao trabalho em questão.

Referências

- [1] Yu, Yijun., Leite, J., Mylopoulos J.: From Goals to Aspects: Discovering Aspects from Requirements Goal Models, Requirements Engineering conference 2004, Kyoto, Japan.
- [2] Bertolini, D., Novikau A., Susi, A., Perini A.: Taom4e: an Eclipse ready tool for Agent-Oriented Modeling. Issue on the development process, Trento-Povo, Italy.
- [3] Spies, E., Rüger, J., Moreira A.: Using i* to identify candidate aspects. In: Workshop. In UML'04, Lisboa, Portugal, October 11 (2004).
- [4] Alencar, F., Silva, C., Moreira, A., Araújo, J., Castro, J.: "Identifying Candidate Aspects with I-star Approach", In: Work. on Early Aspects:, Bonn, Germany, March 2006.
- [5] Alencar, F., Moreira, A., Araújo, J., Castro, J.: Towards an Approach to Integrate i* with Aspects, 8th Intl. Bi-Conference Work. on Agent-Oriented Information Systems (AOIS-2006), in conjunction with CAiSE'06. Luxembourg, June 2006.
- [6] Yu, E.: Modeling Strategic Relationships for Process Reengineering. Ph.D. thesis, Department of Computer Science, University of Toronto, Canada (1995).
- [7] Alencar, F., Moreira, A., Araújo, J., Castro, J., Silva, C., Mylopoulos, J.: Using Aspects to Simplify i* Models, Requirements Engineering conference 2006, Minnesota, USA.
- [8] Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Lopes, C., Loingtier, J., Irwin, J.: Aspect-Oriented Programming. In: Proceedings of the European Conference on Object-Oriented Programming. ECOOP'97, Finland, Springer-Verlag (1997).
- [9] Organization Modelling Environment (OME). Disponível em: <http://www.cs.toronto.edu/km/ome/> (Acessado em 17.11.2006).
- [10] Tool For Agent Oriented Modelling For Eclipse (TAOM4E). Disponível em: <http://sra.itc.it/tools/taom4e/> (Acessado em 17.11.2006).
- [11] Castro, J., Kolp, M., Mylopoulos, J.: Towards Requirements-Driven Information Systems Engineering: The Tropos Project, In: Information Systems Journal, Vol. 27, Elsevier (2002) 365-89.

[12] Giorgini, P., Kolp, M., Mylopoulos, J., Castro, J.: Tropos: A Requirements-Driven Methodology for Agent-Oriented Software. Book Chapter in Agent-Oriented Methodologies. Ed.: Idea Group (2005) 20 – 45.

[13] Araújo, J., Moreira, A., Brito, I., Rashid, A.: Aspect-Oriented Requirements with UML. In: Workshop on Aspect-oriented Modeling with UML, UML 2002, Dresden, Germany (2002)

[14] Brito, I., Moreira, A.: Integrating the NFRframework in a RE model. Early Aspects 2004. In: Workshop of the AOSD'04, Lancaster, UK, March (2004) 22-26

[15] Rashid, A., Moreira, A., Araújo, J.: Modularization and Composition of Aspectual Requirements. In: Proceedings of the Conference on Aspect Oriented Software Development (AOSD'03), USA (2003)

[16] Zambonelli, F., Jennings, N., Wooldridge, M.: “Developing Multiagent Systems: the Gaia Methodology”, In: ACM Transactions on Software Engineering and Methodology, 12(3), p. 317 – 370 (2003).

[17] DeLoach, S.: “Engineering Organization-based Multiagent Systems”, In: Proceedings of the 4th International Workshop on Software Engineering for Large-Scale Multi-Agent Systems (SELMAS'05), Saint Louis, EUA, ACM Press, p. 1 – 7 (2006).

[18] Iglesias, C. Á. “Definición de una Metodología para el Desarrollo de Sistemas Miltiagente”, Tese de Doutorado, Departamento de Engenharia de Sistemas de Telecomunicações, Universidade Politécnica de Madrid. Madrid, 322p (1998).

[19] Terminologia de DSOA em português. Disponível em: <http://twiki.im.ufba.br/bin/view/AOSDbr/TermosEmPortugues> (Acessado em 24.11.2006).

[20] Rashid, A., Sawyer, P., Moreira, A., Araujo, J.: "Early Aspects: A Model for Aspect-Oriented Requirements Engineering", Proc. RE, 2002, IEEE Computer Society Press, pp. 199-202.

| |
|-----------------------------------|
| Proposta de Trabalho de Graduação |
|-----------------------------------|

| |
|--|
| Automação do Processo de Identificação de Aspectos em Modelos i* |
|--|

DATA E ASSINATURAS

Recife, 30 de novembro de 2006

Cleviton Vinicius Fonseca Monteiro (Proponente)

Jaelson Brelaz de Castro (Orientador)

Fernanda Maria Ribeiro de Alencar (Co-orientadora)