

Estruturação de Descrições de Casos de Uso através de Mecanismos de Extensibilidade da UML

Gabriel Silva Borna, BSc.
Roberto Tom Price, Eng., MSc., D. Phil.

Universidade Federal do Rio Grande do Sul – UFRGS
Instituto de Informática
Av. Bento Gonçalves, 9500
Porto Alegre – RS – Brasil
e-mail: {bornia, tomprice}@inf.ufrgs.br

Abstract

This paper presents a structured representation of use case descriptions using stereotyped activity diagrams. The use of the extensibility mechanism of UML is used to configure language elements to be used for the description of system behavior. A way of representing use case descriptions in different levels of abstraction is shown, and ways of associating between descriptive elements and the static model of the system. A CASE tool is presented to demonstrate the proposed use case description method.

Key-words: Use case, use case description, collaboration case, UML, activity diagram, extensibility mechanism, CASE tool.

Resumo

Este trabalho apresenta uma forma de representação estruturada de descrições de casos de uso através do uso de diagramas de atividade estereotipados. O uso da extensibilidade da UML permite configurar elementos da linguagem de tal forma que esta possa também ser utilizada para a descrição do comportamento do sistema. É apresentada uma forma de representação de descrições de casos de uso em vários níveis de abstração, bem como a associação entre elementos da descrição e o modelo estático do sistema. Uma ferramenta CASE é apresentada como prova de conceito para o método de descrição proposto.

Palavras-chave: Caso de uso, descrição de casos de uso, casos de colaboração, UML, diagramas de atividade, mecanismos de extensibilidade, ferramenta CASE.

1 Introdução

Na análise de sistemas orientada a objetos a modelagem de casos de uso possui um papel fundamental. Os casos de uso são o principal mecanismo de levantamento de requisitos [24], caracterizando-se por uma representação narrativa do comportamento do sistema [14][15].

As descrições de casos de uso caracterizam-se por serem descrições do funcionamento do sistema sem, no entanto, entrar em detalhes específicos de como o mesmo será implementado. Diversos autores [6][8][10] defendem que os casos de uso devem ser o mais simples possível, uma vez que devem ser entendidos pelos usuários do sistema – que os utilizam como forma de validação dos requisitos levantados pelo analista – e servir como base aos demais dos envolvidos na construção do sistema.

A necessidade de se criar descrições simples que possam ser entendidas por usuários comuns representa um obstáculo ao uso de mecanismo mais formais de descrição [9] que possam ser utilizados de forma estruturada por projetistas, desenvolvedores e gerentes de projeto.

Existem diversas formas de se representar descrições de casos de uso. A mais comum delas é a forma textual, que embora possua diversos estilos propostos [1][3][17] continua praticamente livre de uma estrutura que permita a modelagem da descrição. Existem diversas propostas de formalizações de casos de uso [12][13] de difícil disseminação no mercado. UML é uma linguagem de modelagem [4] amplamente difundida. Geralmente, os casos de uso são modelados com esta linguagem, mas a interação entre o sistema e o ator (descrição do caso de uso) é realizada de forma narrativa [21]. As descrições dessas interações podem ser realizadas através de diversos elementos da linguagem UML como, por exemplo, diagramas de atividade [3], diagramas de seqüência, diagramas de estado, entre outros. UML possui a facilidade de estender os seus elementos através do conceito de estereótipos, o que poderia ser utilizado para ajudar a enriquecer uma descrição feita com os elementos desta linguagem.

O objetivo deste trabalho é a representação de descrições de casos de uso através de mecanismos de extensibilidade da UML, mais especificamente permitindo a modelagem das atuais representações narrativas por meio de diagramas de atividade estereotipados, garantindo uma representação mais estruturada que possa ser utilizada em outros passos do processo de desenvolvimento de sistemas sem, no entanto, perder a capacidade de se comunicar com o usuário através de mecanismos que permitam gerar uma linguagem que ele compreenda.

2 Descrição de Casos de Uso

Usualmente a descrição dos casos de uso é realizada de forma textual sem nenhum compromisso com uma estrutura de representação mais formal ou estruturada [25]. Este tipo de descrição não pode ser utilizado de forma adequada por mecanismos automatizados que auxiliem o desenvolvimento de sistemas em outras etapas do processo de construção, por não possuir uma estrutura ou modelo de representação adequado.

O formato de uma descrição de casos de uso é dividido em diversas seções, que variam para cada organização, mas as seções mais comuns de serem encontrados em templates de descrições de casos de uso são [5]:

1. Atores: seção com a descrição do atores envolvidos no caso de uso;
2. Pré-condições: regras de estado que definem como o sistema deve se encontrar antes de ocorrer o caso de uso;
3. Fluxo de eventos: interações que ocorrem entre os atores e o sistema à medida que interagem para atingir um determinado objetivo;
4. Pós-condições: regras de estado que definem como o sistema deve se encontrar depois de executado o caso de uso.

A seção mais importante de uma descrição de caso de uso é o fluxo de eventos entre o ator e o sistema. O fluxo de eventos é um conjunto seqüência de eventos do tipo ação-reação que descrevem os caminhos percorridos pelo ator do caso de uso para atingir o seu objetivo.

Um exemplo simples de descrição é mostrado na Tabela 1, na qual o caso de uso “User Login” é descrito através de uma tabela que separa o fluxo entre ator e sistema [23].

Ator	Sistema
1. Preenche o usuário e a senha.	
	2. Valida o usuário e a senha.
	3. Realiza o login do usuário.
Alternativa 1: O usuário e senha estão inválidos. Informa o usuário sobre o erro e requisita novamente as informações de login.	

Tabela 1 – Descrição do caso de uso “User Login”

As descrições de casos de uso podem possuir uma visão externa ou interna do sistema. Em uma visão externa (ou caixa-preta) somente as atividades que são visíveis aos atores externos são descritas sem indicação de como o sistema faz para obter o resultado de uma ação. Na visão interna (ou caixa-branca) o comportamento interno do sistema é descrito. A visão externa permite a validação dos requisitos com o

usuário mas pode acabar perdendo requisitos importantes que poderiam ser obtidos através de uma descrição mais detalhada. A visão interna é de grande utilidade nas próximas fases do desenvolvimento do sistema, mas difícil de ser validada com o usuário [3] devido ao detalhamento utilizado nesta visão.

Le Roy Mattingly e Harsha Rao [18] propuseram os casos de colaboração, que são casos de uso estereotipados para a descrição interna do sistema. A vantagem é a manutenção e refinamento interno dos casos de uso (casos de colaboração/visão interna) uma vez que a interface com o usuário esteja bem definida (casos de uso/visão externa).

3 Descrição através de Diagramas de Atividade

Diversas notações podem ser utilizadas para a representação de casos de uso como, por exemplo, redes de Petri, linguagens formais, pseudo-código, entre outros [25]. Em UML, diagramas de atividades, diagramas de seqüência e diagramas de estado podem ser utilizados para descrever casos de uso.

Os diagramas de atividades se apresentam como uma notação mais clara para a representação de paralelismos, elementos de lógica condicional e sincronização do que os diagramas de seqüência e diagramas de estado [3]. A descrição através de diagramas de atividade torna a descrição um artefato mais formal e estruturado, porém pode ser de difícil entendimento para os usuários e sua validação junto aos mesmos pode ficar prejudicada.

Entretanto, uma ferramenta CASE a partir de uma representação estruturada como essa poderia extrair uma descrição textual fácil de ser compreendida por usuários. A mesma ferramenta pode utilizar essa descrição estruturada para extrair outros artefatos úteis durante o processo de desenvolvimento, como por exemplo, casos de teste, casos de colaboração, métricas de complexidade, diagramas de navegação, entre outros.

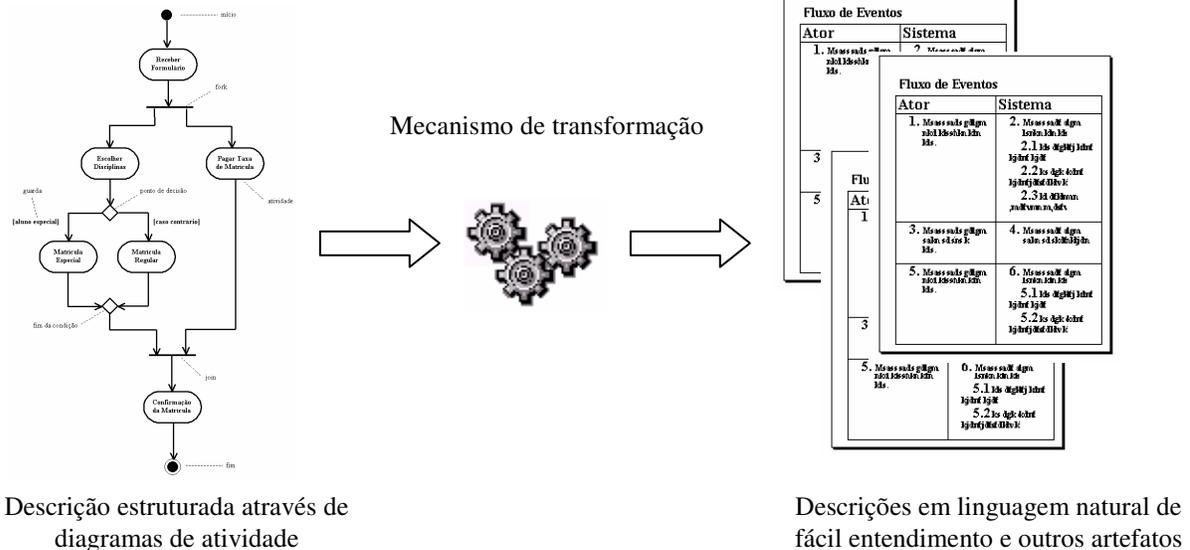


Figura 1 - Mecanismo de transformação

A utilização de diagramas de atividade para a representação de descrições de casos de uso precisa ser adaptada para suportar algumas características próprias das descrições dos casos de uso. As adaptações necessárias – como, por exemplo, a caracterização de atividades que representam passos realizados pelos atores ou pelo sistema dentro de um fluxo de eventos – podem ser realizadas através do mecanismo de extensibilidade da UML [20].

3.1 Representação do fluxo de eventos

O fluxo de eventos de uma descrição de caso de uso representa a interação entre o ator e o sistema. Essa interação se dá através de eventos numerados representados textualmente através de um fluxo contínuo ou de um fluxo separado por colunas [3][23].

Cada evento é realizado pelo ator ou pelo sistema e representa uma interação do tipo ação-reação [20]. A representação através de atividades dentro de um diagrama de atividades pode ser realizada através da utilização dos estereótipos <<ator>> e <<sistema>> para identificar o responsável pela atividade. A descrição dos casos de uso permite a introdução de seqüências alternativas, que representam desvios no fluxo principal. Essas seqüências alternativas podem ser utilizadas para realizar o tratamento de exceções ou indicar um outro caminho possível. A Figura 2 mostra como o exemplo mostrado na Tabela 1 pode ser representado através de um diagrama de atividade que utilize estereótipos.

A representação de seqüências alternativas ou fluxos condicionais é representada através de transições com guardas [4].

Os diagramas de atividades possuem apenas um estado inicial e podem possuir um ou mais estados finais. Os estados finais são utilizados para indicar os possíveis terminos a que a execução do caso de uso pode levar. As pré-condições e pós-condições do sistema são representadas como atributos dos elementos que representam respectivamente os estados iniciais e finais do diagrama de atividade.

Os diagramas de atividade suportam outros elementos como barra de paralelismo e de sincronização, utilizadas para representar fluxos que podem ocorrer em paralelo. As descrições textuais não possuem uma forma simples de representar comportamentos paralelos.

3.2 Representação de casos de colaboração

Os casos de colaboração são casos de uso estereotipados que descrevem o funcionamento interno do sistema [18]. A especificação de UML 1.5 possui o conceito de sub-diagramas de atividade. Essa característica permite que uma atividade possua um outro diagrama de atividade associado dentro dela. Utilizando esse conceito é possível representar casos de colaboração como sub-diagramas das atividades estereotipadas como <<sistema>>.

O caso de uso mais abstrato ou de maior nível pode definir apenas o comportamento geral do caso de uso. Durante o processo de desenvolvimento do sistema, durante a especificação mais específica dos requisitos, o caso de uso pode ir sendo refinado através de um maior detalhamento das atividades do sistema com o uso de sub-diagramas que representem casos de colaboração.

A navegação de forma hierárquica permite que novos casos de colaboração sejam incorporados à descrição inicial do caso de uso. A validação dos requisitos com o usuário pode ser feita através da descrição

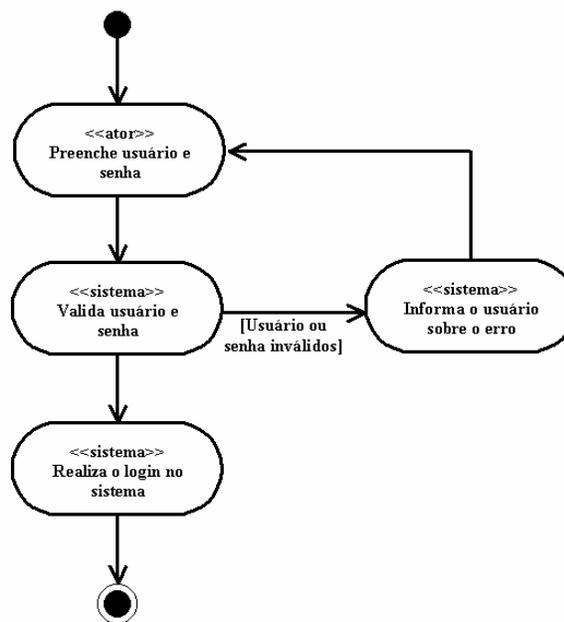


Figura 2 – Descrição através de diagramas de atividades do caso de uso “Login User”

mais alto-nível (primeiro diagrama na hierarquia), enquanto que os outros níveis serviriam de apoio às outras etapas envolvidas no ciclo de desenvolvimento do sistema.

O nível de detalhamento por ir aumentando à medida que novos níveis de descrição são incorporados. Um exemplo de como os casos de colaboração podem ser utilizados como diagramas de atividades estereotipados é mostrado na Figura 3.

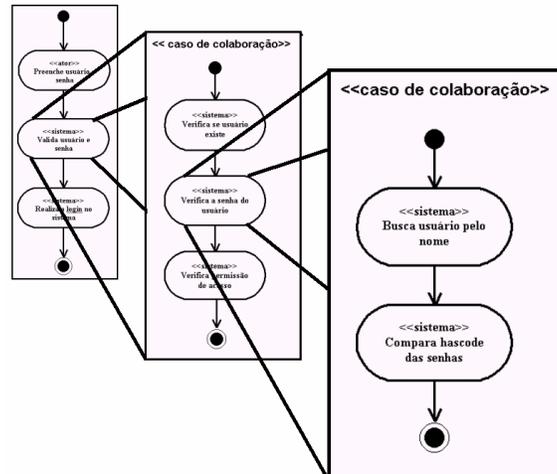


Figura 3 - Representação de casos de colaboração

Uma ferramenta CASE facilitaria o processo de descrição através de mecanismos de navegabilidade drill-down e drill-up. A descrição mais abstrata (alto-nível) poderia ser utilizada para a validação da interação do usuário com o sistema enquanto que a descrição mais específica poderia detalhar o funcionamento interno do projeto.

3.3 Representação de relacionamentos entre os casos de uso

Durante a análise são identificados os casos de uso que compõem o sistema e estes são organizados dentro de um modelo de casos de uso, onde se representam as relações dos atores com os casos de uso, e as relações entre os casos de uso. Os relacionamentos que podem ocorrer entre os casos de uso são os de inclusão, extensão e generalização/especialização [10][11][20].

O relacionamento de inclusão representa uma relação na qual o fluxo de eventos do caso de uso incluído é executado durante o fluxo de eventos de um caso de uso base. Essa chamada deve ser indicada dentro da descrição [3]. Essa indicação é representada através de uma atividade estereotipada como <<ponto de inclusão>>, e existe uma referência entre ela e o caso de uso que está sendo incluído.

Já o relacionamento de extensão apresenta dentro do fluxo de eventos indicações dos lugares onde o comportamento pode ser estendido por outros casos de uso. Essas indicações são chamadas de pontos de extensão [3] e são representadas por atividades estereotipadas como <<ponto de extensão>>. Cada ponto de extensão pode ser referenciado externamente por outro caso de uso.

O relacionamento de generalização/especialização representa a herança de comportamento entre os casos de uso. Nesse caso o fluxo de eventos é herdado e o seu comportamento pode ser alterado ou incrementado [3][11]. Este tipo de relacionamento é pouco utilizado e seu controle é mais complexo, embora

possa simplificar as descrições de casos de uso que possuam pequenas alterações de comportamento. Uma ferramenta CASE poderia auxiliar na manutenção de descrições para casos de uso especializados¹.

A descrição de forma estruturada permite validar o modelo de casos de uso garantindo a consistência entre o que é modelado (relacionamentos de inclusão e extensão entre os casos de uso) e as descrições dos mesmos, uma vez que esses relacionamentos são referenciados na descrição. A relação entre a descrição estruturada e o modelo de casos de uso pode ser vista na Figura 4.

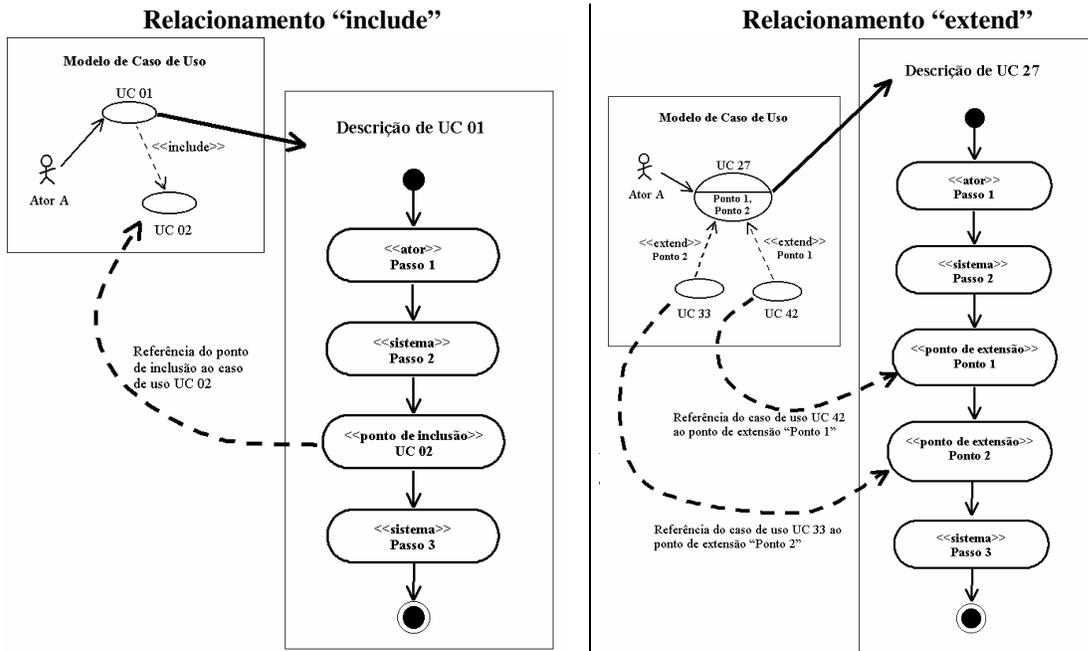


Figura 4 - Representação do relacionamento "include" e "extend"

4 Relação com a arquitetura e o modelo estático do sistema

A arquitetura do sistema pode ser definida como o conjunto de decisões significativas tomadas sobre a organização do sistema, a escolha dos elementos estruturais e interfaces que compõe o sistema [26]. O modelo de visão 4+1 proposto por Kruchten [16] identifica os casos de uso como a interligação entre as diversas visões concorrentes que compõe a arquitetura.

A definição da arquitetura ocorre já nas primeiras fases do ciclo de desenvolvimento. Os casos de uso iniciais ajudam a definir o que seria uma arquitetura adequada para o sistema, o suficiente para poder colocar um protótipo de arquitetura a suportar os casos de uso das primeiras iterações, dentro de um processo de desenvolvimento iterativo [15][19].

À medida que as descrições dos casos de uso e de colaboração vão se detalhando cada vez mais, as descrições entram no universo do projeto de sistemas e acabam se relacionando com a arquitetura definida para a construção do sistema.

A descrição de casos de uso através de diagramas de atividade pode utilizar o conceito de *swim lanes* [10] para identificar estilos arquiteturais (como camadas, sub-sistemas ou componentes do sistema) e identificar os eventos ou atividades associados a cada elemento da arquitetura.

¹ Este trabalho não contempla a descrição de casos de uso especializados, pelo fato de seu uso ser pouco utilizado [3][6][25].

Por exemplo, uma descrição de um caso de colaboração através dos diagramas de atividade poderia possuir as seguintes swim lanes: camada de apresentação, camada de negócio, camada de persistência e camada de banco de dados. Dentro de um mesmo diagrama é possível representar um fluxo de eventos e indicar em que sub-sistemas ou camadas cada evento é realizado.

Na Figura 5 pode ser visto um exemplo no qual o aprofundamento das descrições leva à necessidade de se identificar em que camada cada evento está sendo realizado. Esse tipo de associação pode ser realizado nas etapas de desenvolvimento, através da qual os próprios desenvolvedores ou projetistas podem dar continuidade à descrição criada na análise, refinando-a cada vez mais.

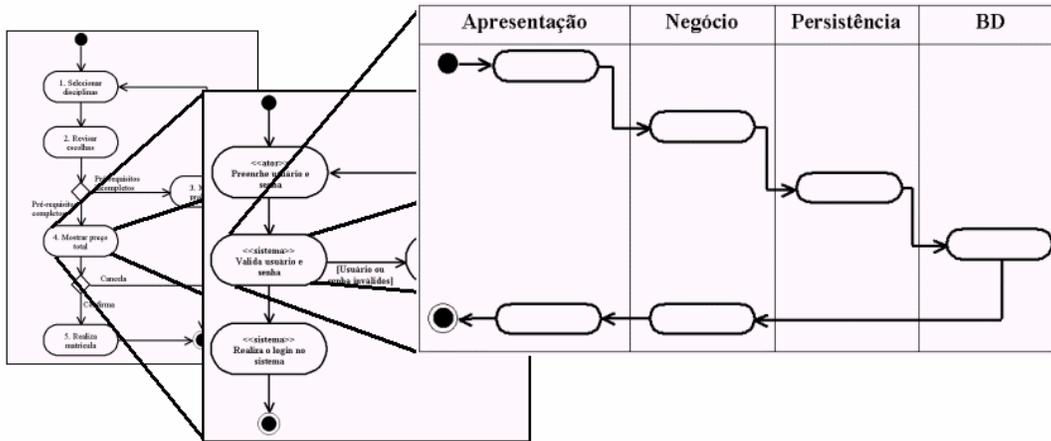


Figura 5 - Exemplo de detalhamento de uma descrição até a arquitetura

Durante o processo de análise de sistemas o modelo de classes vai sendo construído, inicialmente com classes que representam os conceitos de negócio, e posteriormente nas etapas de projeto e desenvolvimento o modelo é enriquecido com classes de projeto mais voltadas à forma como o sistema será implementado, e fortemente ligado à arquitetura definida.

Uma ferramenta CASE pode permitir a associação de elementos da descrição (eventos do fluxo de eventos do caso de uso) a classes do modelo estático ou a métodos de classes. Essa associação permite o rastreamento de conceitos afetados por alterações de requisitos. Se, por exemplo, o usuário quiser alterar alguma funcionalidade é possível identificar os pontos afetados por tal alteração de forma automática.

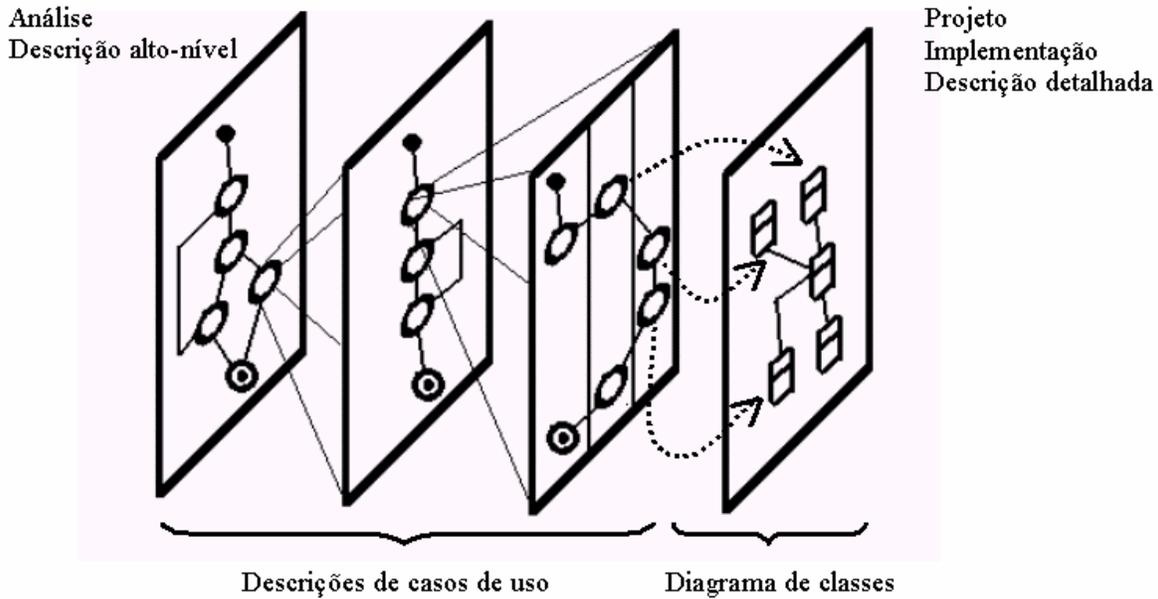


Figura 6 – Refinamento das descrições, relação com a arquitetura e modelo de classes

A Figura 6 mostra como o refinamento das descrições pode evoluir até o ponto de se ter elementos de descrição associados a classes ou métodos.

4 Modelo para a descrição dos casos de uso

A solução proposta de descrição dos casos de uso através de diagramas de atividade estereotipados é mapeada para as classes a seguir. A intenção não é a construção de um modelo que represente as estruturas definidas pela UML. A OMG já apresenta um modelo de mapeamento muito mais genérico para a sintaxe da UML do que o apresentado neste trabalho. O modelo aqui proposto é mostrado na Figura 7, definido para provar o conceito de descrição de casos de uso através de uma estrutura mais formal, baseada em diagramas de atividade. Outra modelagem seria possível, inclusive uma que reaproveitasse a modelagem definida pela OMG.

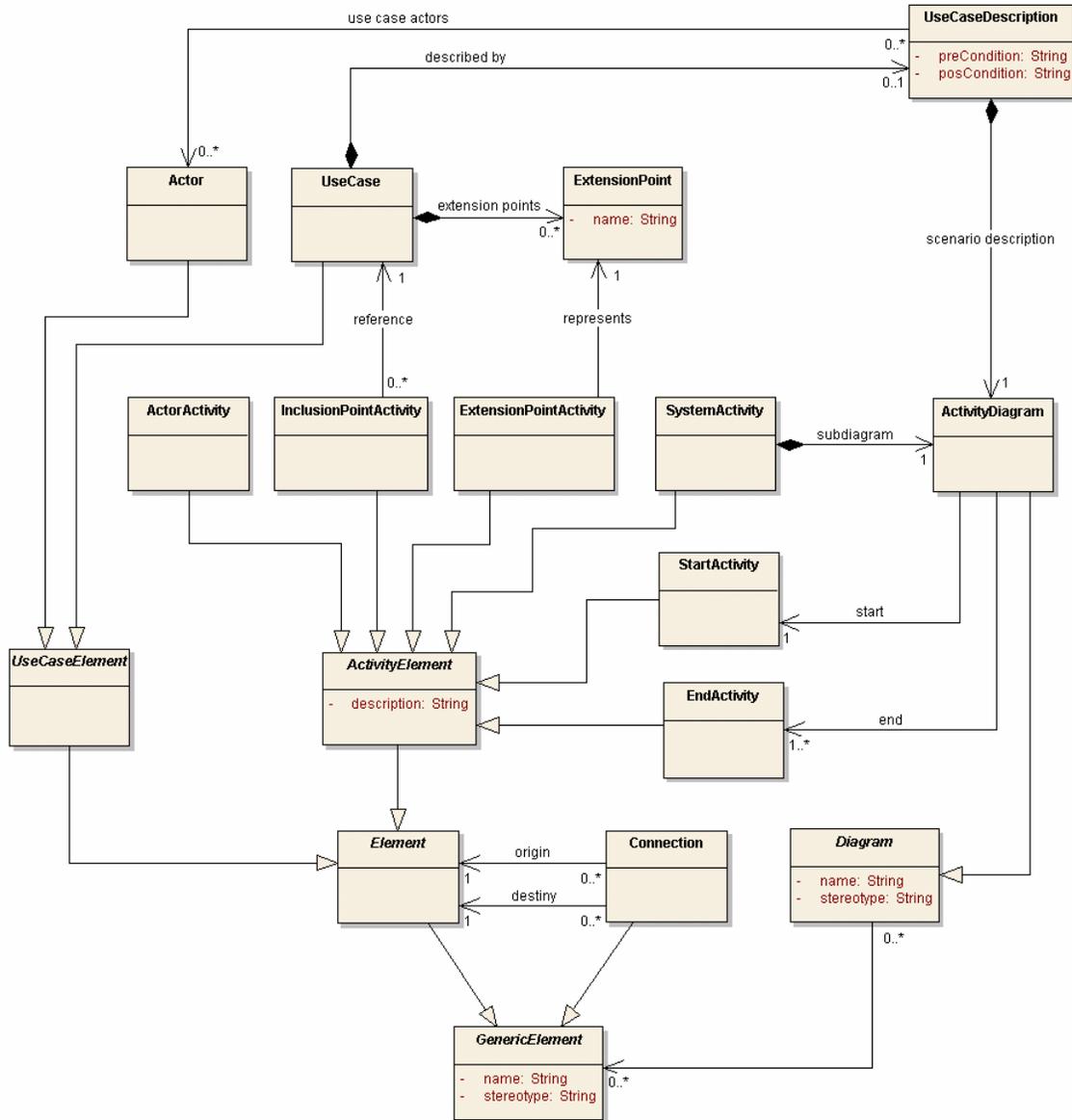


Figura 7 – Modelo geral para representação de descrições de casos de uso através de diagramas de atividade

O relacionamento da descrição com a arquitetura ocorre quando o diagrama de atividade que descreve o caso de uso possui swim lanes associadas às partições ou camadas de uma descrição de arquitetura definida. São identificadas as camadas onde cada atividade da descrição está localizada bem como as classes ou métodos aos quais a atividade está associada. A associação entre a atividade da descrição e classes ou métodos possui uma indicação sobre o tipo de referência realizado (o tipo de referência pode ser “utiliza”, “executa”, entre outros). A Figura 8 mostra o modelo para essas características.

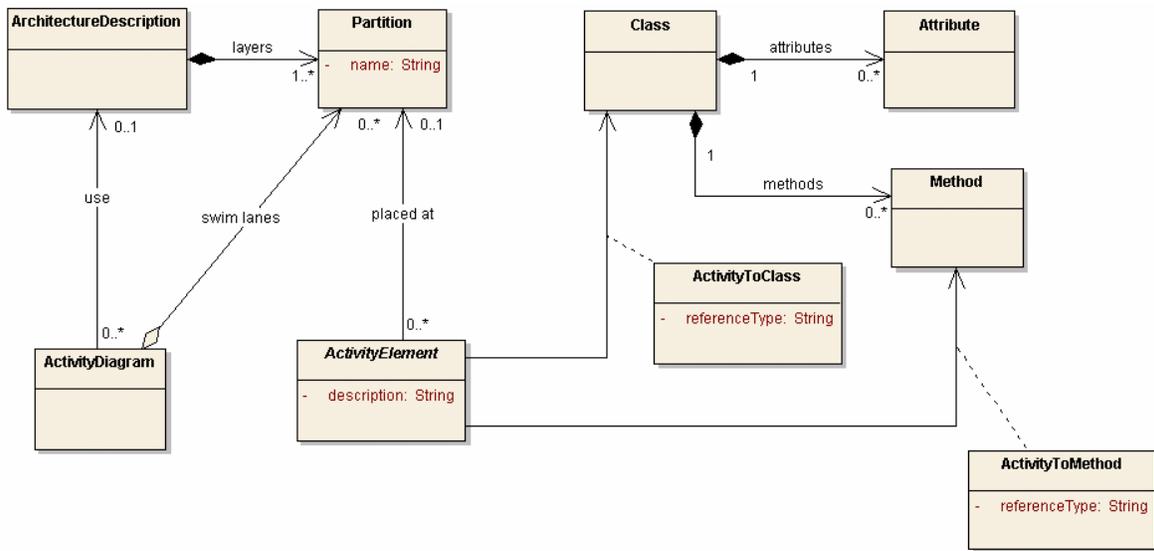


Figura 8 – Relação com a arquitetura e o modelo de classes

A Figura 9 mostra um exemplo de como uma atividade pode estar associada a uma classe e a métodos de classes. A relação com o modelo estático do sistema ocorre nos níveis mais detalhados das descrições de casos de uso.

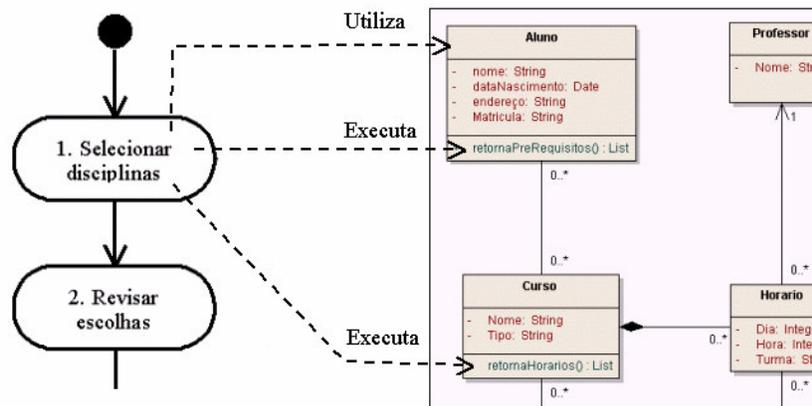


Figura 9 – Associação entre os elementos de descrição e o modelo estático

No exemplo, as referências entre as atividades e o modelo de classes do sistema (classes e métodos) possuem uma classificação cuja semântica poderia ser definida livremente pelos usuários de uma ferramenta CASE com suporte a esse tipo de descrição.

A maioria das ferramentas CASE de modelagem UML encontradas no mercado possuem funcionalidades que permitem associações simples entre casos de uso e artefatos de documentação externos (arquivos), geralmente utilizados para descrever os primeiros de forma narrativa sem nenhum comprometimento com uma estrutura de representação. Algumas ferramentas como, por exemplo, Visual Paradigm [27], possuem meios um pouco mais elaborados que permitem a descrição dos casos de uso dentro da própria ferramenta. Outros projetos como, por exemplo, o desenvolvido pela PUC-Rio chamado C&L (Cenário e Léxico) [28], auxiliam a descrição colaborativa de cenários introduzindo uma organização da informação de forma estruturada. Entretanto, essas ferramentas não permitem a associação de elementos internos das descrições dos casos de uso (passos / atividades) com outros elementos que compõe o modelo do sistema, nem possuem mecanismos para a geração de outros artefatos.

Neste trabalho é apresentada uma ferramenta CASE desenvolvida com o intuito de oferecer um mecanismo de descrição baseado no método apresentado, utilizando diagramas de atividade estereotipados para estruturar as descrições de casos de uso, permitindo o processamento para a geração de artefatos e a associação entre outros elementos do modelo do sistema.

5 Ferramenta CASE de descrição

Para a prova de conceito do método de descrição de casos de uso através de diagramas de atividade com o mecanismo de extensibilidade da UML, foi desenvolvida uma ferramenta para a construção de descrições desse tipo baseado no modelo proposto. A ferramenta se chama UC Papyrus, tem valor apenas acadêmico e possui licença GPL (GNU Public License). A ferramenta e os códigos fontes estão disponíveis em <http://sourceforge.net/projects/ucpapyrus>.

A ferramenta é um editor diagramático de descrições de caso de uso baseadas em diagramas de atividade. As atividades estereotipadas foram transformadas em ícones para facilitar a leitura do diagrama. A Figura 10 mostra a aparência da ferramenta.

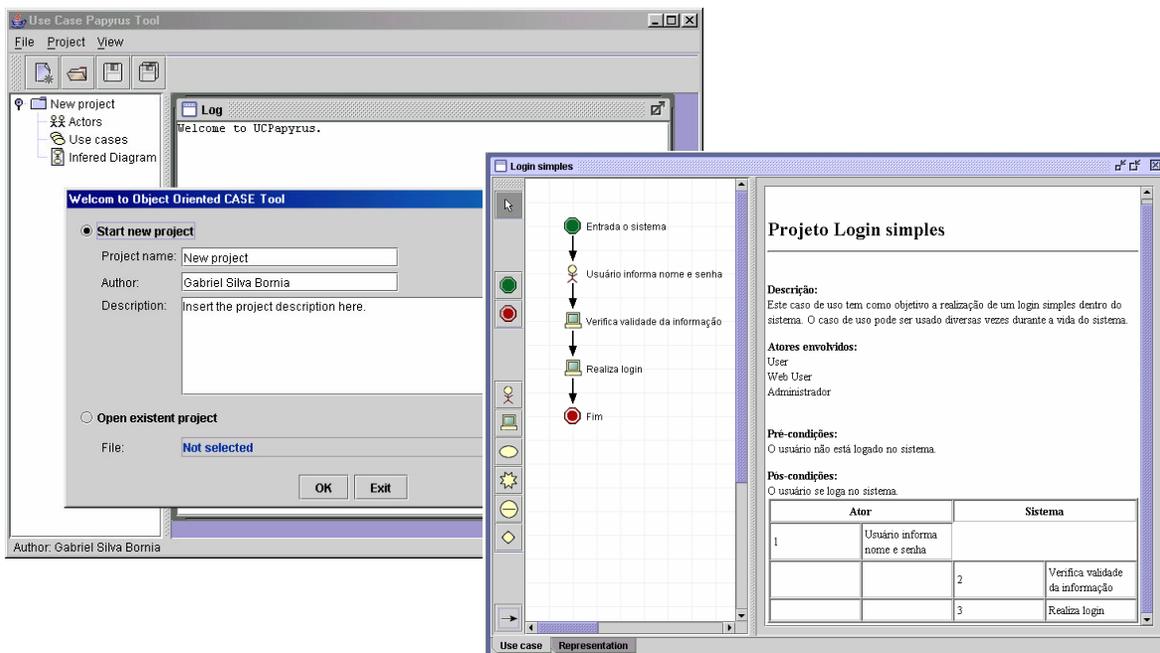


Figura 10 – Ferramenta CASE implementada

Todas as descrições de casos de uso são internamente representadas através de documentos semi-estruturados (XML). Através de regras de transformação XSL é possível processar o documento de representação e gerar diversos artefatos como, por exemplo, uma descrição amigável ao usuário semelhante às descrições textuais normalmente utilizadas no processo de análise. A ferramenta permite a adição de arquivos das regras XSL para processar as descrições.

6 Considerações finais e trabalhos futuros

A importância dos casos de uso vai muito além do escopo da representação de requisitos. Os casos de uso guiam todo o processo de desenvolvimento, a concepção da arquitetura, o projeto do sistema, a

validação dos componentes implementados, a documentação do sistema, entre outras tarefas que compõe o ciclo de vida do sistema.

O fato de se utilizar uma linguagem natural para a descrição interna dos casos de uso pode ser importante para a validação com os usuários, mas é uma informação que pouco pode ser sistematizada em processos automatizados que possam reaproveitar essas informações em outras etapas do desenvolvimento de software.

Este trabalho propõe um mecanismo que estruture as descrições dos casos de uso, de forma que a própria representação dos requisitos sirva como mecanismo de validação junto aos usuários e ao mesmo tempo sirva de entrada para mecanismos computacionais capazes de utilizar essas informações para a geração de outros artefatos úteis nas demais etapas no desenvolvimento do sistema.

O uso da extensibilidade da UML é indicado como meio de utilizar a própria linguagem UML como forma de representação estruturada para descrições de casos de uso. O objetivo é mostrar que as descrições de casos de uso podem ser modeladas através da própria UML e – se devidamente assistidas por uma ferramenta CASE – servir de base para a geração de outros artefatos que auxiliem o processo de desenvolvimento de software.

Atualmente a ferramenta CASE está sendo aprimorada para suportar a geração de informações que possam auxiliar o gerenciamento de projetos. Essas informações são métricas extraídas das próprias descrições como, por exemplo, número de transações, número de pontos de inclusão, número de pontos de extensão. Esses dados serão utilizados como entrada para cálculos de estimativa de esforço para estimar tempos de implementação [2].

A estruturação de casos de uso pode permitir a realização de trabalhos como, por exemplo:

- Geração de scripts em LOTOS para realizar a simulação de passos que possam ser executados dentro de um caso de uso, de forma que esse script possa servir como base para possíveis estudos para testes automatizados;
- A existência de uma descrição diagramática para os casos de uso pode permitir a identificação de padrões de descrições de casos de uso, que eventualmente podem vir a ser catalogados;
- As descrições podem ser enriquecidas com informações que permitam auxiliar a construção de interfaces homem-máquina;
- Implementação da rastreabilidade de alterações de requisitos, indicando os artefatos e classes que devem ser re-visitados [7].

Referências

- [1] Ambler, Scott W. *Documenting a use case: what to include, and why*. Ronin International. EUA, 2000.
- [2] Anda, Bente; Dreiem, Hege; Sjøberg, Drag I.K.; Jørgensen, Magne. *Estimating software development effort based on use cases: experiences from industry*. 4th International Conference on the Unified Modeling Language: UML 2001. Canada, 2001.
- [3] Armour, Frank; Miller, Granville. *Advanced use case modeling: software systems*. Addison-Wesley. EUA, 2000.
- [4] Booch, Grady; Rumbaugh, James; Jacobson, Ivar. *The Unified modeling language user guide*. Addison-Wesley. EUA, 1999.
- [5] Cockburn, Alistair. *Structuring use cases with goal*. Humans and Technology. EUA, 2000.
- [6] Cockburn, Alistair. *Use cases, ten years later – from their evolution, learn what to expect and how to better work with them*. STQE Magazine. EUA, 2002.
- [7] Ecklund, Earl F.; Delcambre, Lois; Freiling, Michael. *Change cases: use cases that identify future requirements*. 11th ACM Conference on Object-Oriented Programming Systems, Languages and

- Applications: OOPSLA'96. EUA, 1996.
- [8] Firesmith, Donald G. *Use cases: the pros and cons. Report on Object Analysis and Design 2*. EUA, 1995.
 - [9] Fowler, Martin; Cockburn, Alistair; Jacobson, Ivar; Anderson, Bruce; Graham, Anderson. "Question time! About use cases". 13th ACM Conference on Object-Oriented Programming Systems, Languages and Applications: OOPSLA'98. Canada, 1998.
 - [10] Fowler, Martin; Scott, Kendall. *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley. EUA, 1999.
 - [11] Génova, Gonzalo; Llorens, Juan; Quintana, Victor. *Digging into use case relationships*. 5th International Conference on the Unified Modeling Language: UML 2002. Alemanha, 2002.
 - [12] Hurlbut, Russel R. *The Three R's of Use Case Formalisms: Realization, Refinement, and Reification*. Expertech, Ltda. EUA, 1997.
 - [13] Hurlbut, Russel R. *A Survey of Approaches for Describing and Formalizing Use Cases*. Expertech, Ltda. EUA, 1997.
 - [14] Jacobson, Ivar; Christerson, Magnus; Patrik Jonsson; Övergaard, Gunnar. *Object-oriented software engineering: a use case driven approach*. Addison-Wesley. EUA, 1992.
 - [15] Jacobson, Ivar; Booch, Grady; Rumbaugh, James. *The unified software development process*. Addison-Wesley. EUA, 1998.
 - [16] Kruchten, Philippe. *Architectural blueprints – the "4+1" view model of software architecture*. IEEE Software. EUA, 1995.
 - [17] Larman, Craig. *Applying UML and patterns: an introduction to object-oriented analysis and design*. Prentice Hall. EUA, 1998.
 - [18] Mattingly, Le Roy; Rao, Harsha. *Writing effective use cases and introducing collaboration cases. Journal of object oriented programming 11*. EUA, 1998.
 - [19] Rosenberg, Doug; Scott, Kendall. *Use Case Driven Object Modeling with UML: A practical approach*. Addison-Wesley. EUA, 1999.
 - [20] Rumbaugh, James; Jacobson, Ivar; Booch, Grady. *The unified modeling language reference manual*. Addison-Wesley. EUA, 1999.
 - [21] Schneider, Geri; Winters, Jason. *Applying Use Cases: A practical guide*. Addison-Wesley. EUA, 1998.
 - [22] Sendall, Shane; Strohmeier, Alfred. *From use caso to system operation specifications*. 3th International Conference on the Unified Modeling Language: UML 2000. Reino Unido, 2000.
 - [23] Wirfs-Brock, Rebecca. *Designing scenarios: making the case for the use case framework. Smalltalk report*. EUA, 1993.
 - [24] Leffingwell, Dean; Widrig, Don. *Managing Software Requirements: A Unified Approach*. Addison-Wesley. EUA, 2000.
 - [25] Cockburn, Alistair. *Writing Effective Use Cases*. Addison-Wesley. EUA, 2000.
 - [26] Shaw, Mary; Garlan, David. *Software Architecture: perspectives on an emerging discipline*. Prentice Hall. EUA, 1996.
 - [27] Visual Paradigm for UML. <http://www.visual-paradigm.com>. Acesso em 10/02/2004.
 - [28] Cristoph, Roberto; Felicíssimo, Carolina; Leite, Julio. *C&L: Uma ferramenta de edição e visualização de cenários e léxicos*. PUC-RJ. <http://sl.les.inf.puc-rio.br/cel/>. Acesso em 10/02/2004.