

Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação
Departamento de Engenharia de Computação e Automação Industrial



Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade

Luiz Eduardo Galvão Martins

Tese apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas, como parte dos requisitos exigidos para obtenção do título de Doutor em Engenharia Elétrica.

Banca:

Dra. Beatriz Mascia Daltrini (orientadora)

Dra. Heloisa Rocha

Dr. Jaelson F. B. Castro

Dr. José Carlos Maldonado

Dr. Mario Jino

Dr. Ricardo Ribeiro Gudwin

Campinas, SP, agosto de 2001

Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

M366m	<p>Martins, Luiz Eduardo Galvão</p> <p>Uma metodologia de elicitação de requisitos de software baseada na teoria da atividade / Luiz Eduardo Galvão Martins. --Campinas, SP: [s.n.], 2001.</p> <p>Orientadora: Beatriz Mascia Daltrini.</p> <p>Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.</p> <p>1. Engenharia de software - Metodologia. 2. Software – Desenvolvimento. 3. Análise de sistemas. 4. Computação. I. Daltrini, Beatriz Mascia. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.</p>
-------	---

RESUMO

Resumo

Este trabalho apresenta uma metodologia de elicitação de requisitos de software baseada no conceito de atividade. Este conceito é oriundo da Teoria da Atividade, que é uma corrente filosófica importante dentro do campo da Psicologia. Nesta teoria, a atividade é concebida como uma unidade que define um contexto de análise das ações realizadas por um sujeito. A metodologia proposta busca nos elementos constitutivos da atividade referenciais que auxiliem a organização do processo de elicitação de requisitos, que passa a se organizar em torno do conceito de atividade. A propositura desta metodologia parte do pressuposto de que o entendimento das atividades realizadas pelos atores envolvidos em um sistema constitui-se em aspecto chave para o sucesso do software a ser implementado. O conceito de atividade adotado embute elementos como comunidade, regras de conduta, divisão do trabalho, ferramentas de mediação, objetos de transformação, ações e operações, motivos, metas e condições de realização. O referencial teórico deste trabalho está centrado na Engenharia de Requisitos e na Teoria da Atividade. Como exemplo de aplicabilidade da metodologia proposta desenvolvemos um estudo de caso para a elicitação de requisitos em um sistema de protocolos de documentos de uma faculdade. No final do trabalho, utilizando os resultados obtidos no estudo de caso, demonstramos a viabilidade de ligação da metodologia de elicitação proposta com metodologias de análise e especificação de requisitos, através do mapeamento da atividade para outras unidades conceituais que embutem funcionalidade do sistema (por exemplo, casos de uso).

Palavras-Chave

Atividade, Elicitação de Requisitos, Engenharia de Requisitos, Requisitos, Teoria da Atividade.

Abstract

This work presents a methodology for software requirements elicitation based on the activity concept. This concept comes from the Activity Theory, an important philosophical stream in psychology. A activity is conceived as a unit which defines an analysis context for actions performed by a subject. The proposed methodology uses the constitutive elements of activity to build a requirement elicitation process. The proposed methodology claims that understanding activities performed by system actors is a key factor for software implementation success. The activity concept adopted embeds several elements such as community, conduct rules, labor division, mediation tools, object transformation, actions and operations, motives, goals and performing conditions. The theoretical reference of this work is centered on Requirements Engineering and Activity Theory. An example of the methodology applicability is presented in a case study for requirements elicitation of a document protocol system in a faculty. The results of the case study are then used to show how the proposed elicitation methodology can be linked to analysis and specification requirements methodologies, mapping activity to conceptual units which embed system functionality (e.g. use cases).

Keywords

Activity, Requirements Elicitation, Requirements Engineering, Requirements, Activity Theory

AGRADECIMENTOS

Agradeço inicialmente ao colega Luciano Krob Meneghetti, pois foi por intermédio dele que tive os primeiros contatos com a pessoa que se tornaria minha orientadora no doutorado.

Especiais agradecimentos à minha orientadora, Professora Beatriz Mascia Daltrini, que acreditou no meu potencial e ofereceu o suporte necessário para a realização deste trabalho.

Agradeço aos professores do Programa de Pós-Graduação da FEEC com os quais tive contato, que das mais variadas formas me motivaram a realizar meu doutoramento.

Agradeço à comissão de programa do *V Workshop de Teses em Engenharia de Software (WTES – SBC, 2000)*, pelos comentários e críticas construtivas, que se revelaram importantes no processo de amadurecimento do meu trabalho.

Agradeço às secretárias de diversos setores da UNIMEP, que me ajudaram na coleta de informações para a realização do estudo de caso apresentado neste trabalho.

Agradeço à Ana Estela pela ajuda, incentivo e apoio dedicados a mim ao longo do meu doutorado.

Agradecimentos finais vão para as lideranças acadêmicas da Universidade Metodista de Piracicaba (UNIMEP), que quando solicitadas me liberaram de atividades de trabalho para a realização de meu doutorado.

DEDICATÓRIA

Dedico este trabalho a meu filho Caio.

Penso que fazemos todos parte de uma coisa só, que é o que chamamos de Vida. Somos manifestações da Vida, em constante mutação. Para mim a Vida é um único, maravilhoso, corpo místico. Temos apenas que aproveitar o curto tempo desta manifestação da Vida, sobre a qual não temos nenhum controle.

Sumário

PARTE I

1. Introdução.....	13
2. Engenharia de Requisitos	19
2.1. Requisitos de Sistemas <i>versus</i> Requisitos de Software	21
2.2. A Engenharia de Requisitos no Contexto do Desenvolvimento de Software	23
2.3. Atividades da Engenharia de Requisitos.....	25
2.3.1. Elicitação.....	26
2.3.2. Análise.....	27
2.3.3. Especificação.....	29
2.3.4. Validação.....	30
2.3.5. Gerenciamento	32
2.4. Dificuldades Essenciais na Elicitação de Requisitos	33
3. Técnicas Empregadas na Elicitação de Requisitos	39
3.1. Introspecção	40
3.2. Observação.....	41
3.3. Questionário.....	42
3.4. Entrevista	43
3.5. <i>JAD</i>	44
3.6. Análise de Protocolo	46
3.7. Análise de Discurso	47
3.8. Prototipação	47
3.9. Casos de Uso.....	48
3.10. Cenários.....	51

4. Teoria da Atividade.....	55
4.1. Princípios Básicos	56
4.2. O Conceito de Atividade.....	59
4.3. Níveis da Atividade.....	62

PARTE II

5. Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade.....	67
5.1. Visão Geral da Metodologia	68
5.2. Definições dos Principais Conceitos Relativos à Atividade	71
5.2.1. Conceitos Relativos à Estrutura Hierárquica da Atividade.....	71
5.2.2. Conceitos Relativos aos Elementos que Formam o Contexto da Atividade.....	75
5.2.3. Relacionamento Entre os Conceitos Envolvidos na Atividade.....	79
5.3. Descrição das Etapas.....	81
5.3.1. Dividir o Problema em Atividades Realizadas no Contexto do Sistema	82
5.3.1.1. Levantar Atividades Candidatas.....	82
5.3.1.2. Selecionar Atividades.....	83
5.3.1.3. Descrever Histórico das Atividades Selecionadas	83
5.3.2. Delinear o Contexto da Atividade	83
5.3.2.1. Identificar os Motivos e Resultados da Atividade	84
5.3.2.2. Identificar os Elementos no Nível Individual	84
5.3.2.3. Identificar os Elementos no Nível Social.....	85
5.3.2.4. Modelar a Atividade através do Diagrama de Engeström.....	86
5.3.3. Descrever a Estrutura Hierárquica da Atividade.....	87
5.3.3.1. Identificar as Ações e Operações da Atividade.....	88
5.3.3.2. Descrever as Metas das Ações	89
5.3.3.3. Descrever as Condições de Realização das Operações.....	89

6. Estudo de Caso	93
6.1. Elicitação de Requisitos para um Software de Controle de Protocolos da Secretaria de uma Faculdade	94
6.1.1. Descrição Inicial do Problema	94
6.1.2. Abordagem Adotada para a Elicitação dos Requisitos	95
7. Ligação da Metodologia de Elicitação de Requisitos Baseada em Atividades com Algumas Técnicas de Análise e Especificação de Requisitos.....	113
7.1. <i>UML</i>	115
7.1.1. Da Elicitação Baseada em Atividades ao Diagrama de Casos de Uso	115
7.1.2. Da Elicitação Baseada em Atividades ao Diagrama de Seqüência...	122
7.1.3. Elicitação Baseada em Atividades ao Diagrama de Atividades.....	123
Conclusão	127
Referências Bibliográficas	131
Apêndice A - Artigo: “An Approach to Software Requirements Elicitation Using Precepts from Activity Theory”	141
Apêndice B – Artigo: “Estudo Comparativo sobre as Técnicas de Elicitação de Requisitos do Software”	153
Apêndice C - Artigo: “An Approach of Software Requirements Elicitation Based on Activity Theory”	163
Apêndice D – Artigo: “Contribuições da Teoria da Atividade para Processos de Engenharia de Requisitos Baseado em Casos de Uso”	171

Índice de Figuras

Figura 1.1 – Os requisitos como base do processo de desenvolvimento de software	14
Figura 1.2 – Organização da tese	17
Figura 2.1 – Principal influência sobre os requisitos do sistema	22
Figura 2.2 – Principais influências sobre os requisitos do software	22
Figura 2.3 – A Engenharia de Requisitos no contexto do desenvolvimento de software	23
Figura 2.4 – Modelo espiral do processo de Engenharia de Requisitos.....	25
Figura 2.5 – Espiral representando o processo de interação entre elicitação e análise de requisitos	28
Figura 2.6 - O processo de validação dos requisitos	31
Figura 3.1 – Diagrama de casos de uso	49
Figura 3.2 – Exemplo de descrição do caso de uso “Processar Consulta”	50
Figura 3.3 – Meta-esquema para modelagem de requisitos baseada em casos de uso	51
Figura 4.1 - Relacionamento mediado entre sujeito e objeto no nível individual.....	60
Figura 4.2 - Estrutura da atividade “codificar um programa”	61
Figura 4.3 - Modelo sistêmico da atividade (também chamado de diagrama de Engeström)	62
Figura 4.4 - Níveis hierárquicos de uma atividade.....	63
Figura 4.5 – Relacionamento entre os níveis da atividade	64
Figura 5.1 – Etapas da metodologia de elicitação de requisitos proposta.....	68
Figura 5.2 – Decomposição da etapa “Divisão do problema em atividades”	69
Figura 5.3 – Decomposição da etapa “Delineamento do contexto da atividade”	70
Figura 5.4 – Decomposição da etapa “Descrição da estrutura hierárquica da atividade”	70

Figura 5.5 – Relacionamento entre os elementos que formam a estrutura hierárquica da atividade	79
Figura 5.6 - Relacionamento entre os elementos que formam o contexto da atividade.....	80
Figura 5.7 – Tipos de relacionamento representados no diagrama de Engeström	87
Figura 6.1 - Quadro de registro do protocolo.....	94
Figura 6.2 - Modelo sistêmico para a atividade “Criar Protocolo”	103
Figura 6.3 - Modelo sistêmico para a atividade “Atualizar Quadro de Registro do Protocolo”.....	103
Figura 6.4 - Modelo sistêmico para a atividade “Consultar Protocolo por Data”	104
Figura 6.5 - Modelo sistêmico para a atividade “Consultar Protocolo por Assunto”	104
Figura 7.1 – Modelo de casos de uso para o sistema de protocolos.....	116
Figura 7.2 – Quadro descritivo do caso de uso “Criar Protocolo”	118
Figura 7.3 – Quadro descritivo do caso de uso “Atualizar Quadro de Registro do Protocolo”	118
Figura 7.4 – Quadro descritivo do caso de uso “Consultar Protocolo por Data”	119
Figura 7.5 – Quadro descritivo do caso de uso “Consultar Protocolo por Assunto”	119
Figura 7.6 – Mapeamento dos elementos da atividade para os componentes do quadro descritivo do caso de uso.....	120
Figura 7.7 – Quadro descritivo do caso de uso “Criar Protocolo” com elementos complementares extraídos da atividade “Criar Protocolo”	121
Figura 7.8 – Diagrama de seqüência extraído a partir da atividade “Criar Protocolo por Data”	122
Figura 7.9 – Diagrama de seqüência extraído a partir da atividade “Criar Protocolo por Assunto”	123
Figura 7.10 – Diagrama de atividades para “Atualizar Quadro de Registro do Protocolo”	124

Índice de Tabelas

Tabela 2.1 – Custos relativos de reparo do software nas diferentes etapas do ciclo de vida	24
Tabela 4.1 - Decomposição da atividade “Faturar uma Venda”	64
Tabela 5.1 – Definições e princípios da Teoria da Atividade utilizados em cada etapa da metodologia.....	90
Tabela 6.1 – Descrição dos históricos das atividades	98
Tabela 6.2 – Descrição dos motivos e resultados das atividades	99
Tabela 6.3 – Descrição dos elementos das atividades no nível individual	100
Tabela 6.4 – Descrição dos elementos das atividades no nível social	101
Tabela 6.5 - Decomposição da atividade “Criar Protocolo”	105
Tabela 6.6 - Decomposição da atividade “Atualizar Quadro de Registro do Protocolo”	106
Tabela 6.7 - Decomposição da atividade “Consultar Protocolo por Data”	106
Tabela 6.8 - Decomposição da atividade “Consultar Protocolo por Assunto”	106
Tabela 6.9 – Descrição das metas das ações que compõem a atividade “Criar Protocolo”	107
Tabela 6.10 - Descrição das metas das ações que compõem a atividade “Atualizar Quadro de Registro do Protocolo”	107
Tabela 6.11 - Descrição das metas das ações que compõem a atividade “Consultar Protocolo por Data”	107
Tabela 6.12 - Descrição das metas das ações que compõem a atividade “Consultar Protocolo por Assunto”	107
Tabela 6.13 – Descrição das condições de realização das operações da atividade “Criar Protocolo”	108
Tabela 6.14 - Descrição das condições de realização das operações da atividade “Atualizar Quadro de Registro do Protocolo”	109

Tabela 6.15 - Descrição das condições de realização das operações da atividade “Consultar Protocolo por Data”	110
Tabela 6.16 - Descrição das condições de realização das operações da atividade “Consultar Protocolo por Assunto”	110

Parte I

Capítulo 1

Introdução

1. Introdução

Os requisitos do software definem o escopo de um projeto de software, na medida que acrescentamos requisitos para o software o escopo do projeto aumenta, se retiramos alguns requisitos o escopo diminui [Kul00]. Além da definição do escopo, a definição dos requisitos também é necessária para o planejamento e estimativas de variáveis do projeto de software [Amb00].

A partir das afirmações acima, fica clara a importância da correta definição e documentação dos requisitos do software, que irão estabelecer a base para todo o processo de desenvolvimento do sistema de software. Os demais artefatos de software, como arquitetura, interface com o usuário, estrutura de dados, programas e estratégias de teste, que são produzidos ao longo do processo de desenvolvimento, terão como “fundação” os requisitos do software identificados e documentados, resultantes do processo de Engenharia de Requisitos (*Figura 1.1*).



Figura 1.1 – Os requisitos como base do processo de desenvolvimento de software

A elicitação de requisitos é a primeira atividade no processo de Engenharia de Requisitos, onde se busca entender quais são as necessidades do usuário que devem ser atendidas pelo software que será desenvolvido [Tha97]. Várias técnicas de elicitação de requisitos de software têm sido

utilizadas pelos desenvolvedores, de tal forma que as dificuldades inerentes a esta atividade possam ser enfrentadas de maneira mais adequada. No entanto, apesar dessas técnicas oferecerem auxílio aos desenvolvedores, as dificuldades essenciais da elicitação de requisitos, tais como: dificuldade do usuário em saber o que ele realmente precisa, dificuldade no processo de comunicação entre usuários e desenvolvedores, e conseqüentemente, dificuldade em organizar os requisitos do software para o desenvolvimento futuro, ainda constituem um desafio a ser superado.

Argumentamos neste trabalho que para entender quais são as reais necessidades do usuário é preciso identificar e compreender as atividades realizadas por ele no seu contexto de trabalho. Vários pesquisadores têm destacado a importância de se levar em conta o contexto social em que os atores do sistema atuam para uma elicitação de requisitos efetiva [Gog97] [Luf94]. Conforme destacado em [Bre99], grandes sistemas envolvem um número significativo de objetos sociais complexos, que são restritos ao seu contexto, infra-estrutura e história.

Portanto, entendemos que uma análise pormenorizada das ações realizadas pelo usuário, das metas que ele pretende atingir com suas ações, da comunidade em que ele está inserido, das regras e ferramentas de mediação que ele utiliza para a execução de suas ações, do desenvolvimento histórico das atividades que ele realiza, entre outras coisas, se torna fundamental para uma adequada extração dos requisitos do software que irá suportar o trabalho do usuário.

A partir disto, defendemos a idéia de que entender as atividades realizadas pelo usuário, por meio de seus principais aspectos, senão todos os passíveis de reconhecimento, passa a ser central na elicitação de requisitos para uma grande parte de sistemas de software (principalmente em sistemas de informação organizacionais).

Na Psicologia, há algumas décadas vem se desenvolvendo uma teoria que enfoca a atividade humana como o ponto central para o entendimento da formação dos processos mentais humanos, e conseqüentemente, para uma compreensão mais esclarecedora sobre nosso comportamento. Esta teoria é denominada Teoria da Atividade.

Na área da Computação, o interesse sobre a Teoria da Atividade surgiu inicialmente na comunidade de IHC¹, que vem discutindo o potencial da Teoria da Atividade como uma estrutura de fundamentação para as pesquisas e projetos em IHC [Kuu96].

Mais recentemente, na comunidade de Engenharia de Requisitos, começam a surgir alguns trabalhos explorando possibilidades de uso dos preceitos da Teoria da Atividade para a captura, modelagem e definição de requisitos de software [Eli98][Mar99a][Mar99b][Mar99c][McG00].

Neste trabalho, apresentamos uma metodologia de elicitação de requisitos centrada na análise das atividades realizadas pelos usuários do sistema, onde o conceito de atividade que utilizamos está baseado na Teoria da Atividade.

Em [Cas00], Castro destaca que estudos recentes têm feito distinção entre captura de requisitos na fase inicial e captura de requisitos na fase final da Engenharia de Requisitos. A captura de requisitos na fase inicial tem ênfase no entendimento das motivações e razões que fundamentam os requisitos do sistema. Dentro desta visão, a metodologia de elicitação de requisitos proposta neste trabalho oferece suporte para a captura de requisitos na fase inicial da Engenharia de Requisitos.

O trabalho está dividido em duas partes. A primeira parte, composta pelos Capítulos 2, 3 e 4, traz uma problematização e fundamentação teórica sobre conceitos da Engenharia de Requisitos (Capítulo 2), uma coletânea das principais técnicas de elicitação de requisitos utilizadas pelos desenvolvedores de software (Capítulo 3) e uma fundamentação teórica sobre a Teoria da Atividade (Capítulo 4).

A segunda parte é composta pelos Capítulos 5, 6 e 7, que apresentam a metodologia de elicitação de requisitos de software proposta (Capítulo 5), a aplicação da metodologia em um estudo de

¹ Interação Homem-Computador

caso (Capítulo 6), e a ligação que se pode fazer da metodologia de elicitação proposta com algumas técnicas de análise e especificação de requisitos (Capítulo 7).

A *Figura 1.2* apresenta uma visão pictórica da organização e articulação dos assuntos tratados na tese, dividida em capítulos. No apêndice do trabalho, apresentamos alguns artigos publicados ao longo do processo de elaboração da tese.

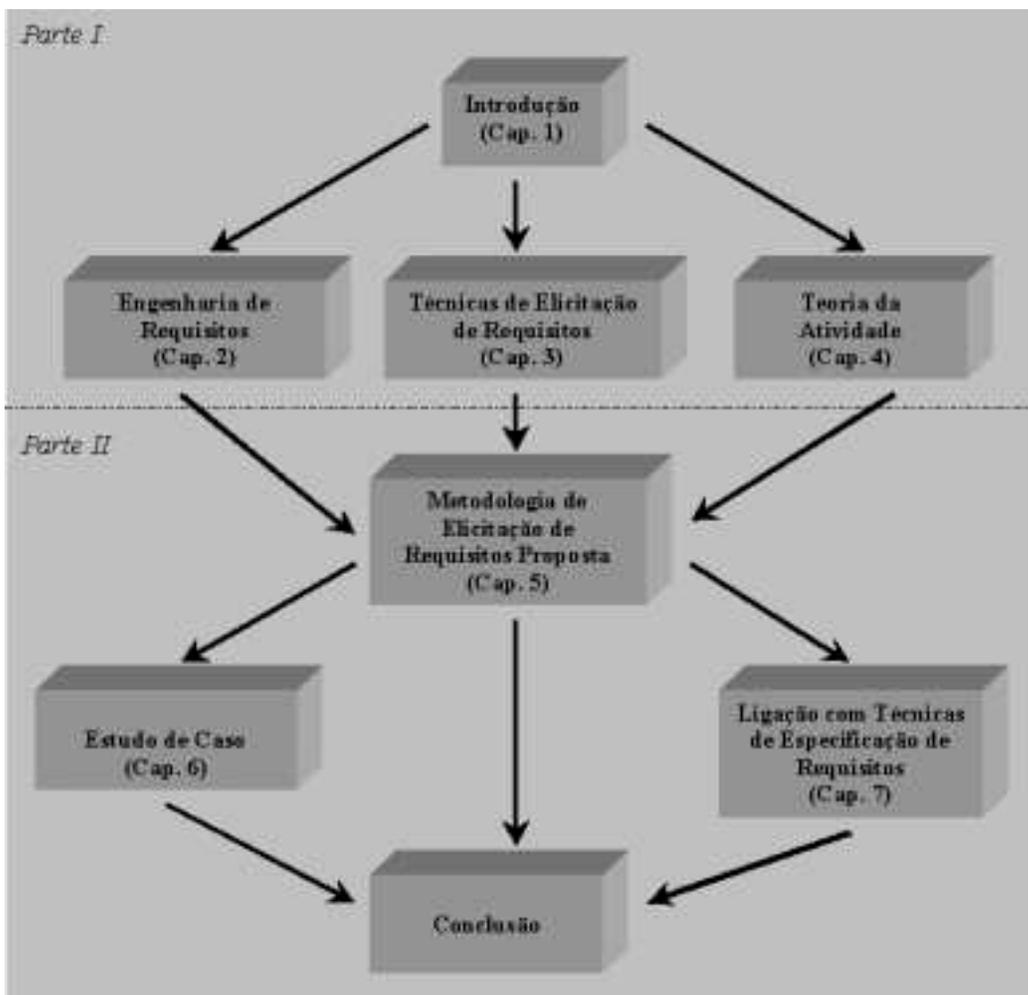


Figura 1.2 – Organização da tese

Capítulo 2

Engenharia de Requisitos

2. Engenharia de Requisitos

A Engenharia de Requisitos é uma disciplina que vem crescendo em importância dentro do contexto de desenvolvimento de software, esse crescimento é decorrente do valor estratégico que vem sendo atribuído aos requisitos como elemento fundamental para o sucesso de projetos de software.

O termo Engenharia de Requisitos é relativamente novo², cunhado com o propósito de abranger todas as atividades envolvidas no descobrimento, documentação e manutenção de um conjunto de requisitos para um sistema baseado em computador [Kot98]. Pode-se dizer que a Engenharia de Requisitos vem substituir a denominação clássica “análise de sistemas”, que não reflete de forma adequada todas as atividades envolvidas no tratamento dos requisitos de um sistema de software.

Embora existam visões um pouco variadas sobre as atividades que compõem a Engenharia de Requisitos, podemos dividi-la nas seguintes atividades [Tha97] [Kot98]: elicitação, análise, especificação, validação e gerenciamento, (abordadas com detalhes na Seção 2.3). Cada uma dessas atividades constitui-se em um processo próprio, com objetivos bem definidos.

Iniciamos nossa discussão sobre Engenharia de Requisitos abordando o objeto principal dessa engenharia, ou seja, o requisito. Para efeitos de esclarecimentos faremos uma breve explanação sobre o que sejam requisitos de sistema e requisitos de software, muito embora estes termos sejam utilizados quase que de forma sinônima.

² Considera-se que a área surgiu em 1993, com a realização do *I International Symposium on Requirements Engineering*.

2.1 Requisitos de Sistema *versus* Requisitos de Software

De uma forma geral podemos definir um requisito como sendo uma declaração de um serviço ou uma restrição que deve ser implementada em um sistema [Kot98]. Em [Kul00] um requisito é definido como qualquer coisa que uma aplicação de computador possa fazer para seus usuários.

É comum encontrarmos os termos requisitos de sistema e requisitos de software sendo utilizados quase que de forma sinônima. Embora a sobreposição destes termos não acarrete maiores problemas, pois normalmente são empregados dentro de um mesmo contexto, ou seja, o de definir a funcionalidade e restrições do sistema de software a ser desenvolvido, parece-nos importante fazer a distinção real que existe entre eles.

Os requisitos de sistema envolvem um contexto mais amplo e menos técnico que os requisitos de software, pois devem descrever o comportamento do sistema visto do “lado de fora”, ou seja, vindos da ótica do usuário do sistema [Dor97]. A representação dos requisitos do sistema deve ser feita de tal forma que possa servir de veículo de comunicação entre os usuários e os desenvolvedores do software.

Provavelmente a maior parte dos requisitos do sistema apontados pelos usuários corresponde a funcionalidades e restrições relativas aos processos realizados por eles em seu contexto de trabalho, que devem ser apoiadas pelo software a ser implementado, e portanto passarão também a ser vistos como requisitos do software. Esses requisitos são chamados de requisitos funcionais

No entanto, além dos requisitos funcionais, os requisitos do software também envolvem restrições e exigências quanto ao desempenho, segurança de acesso, interface com o usuário, portabilidade, modularidade, manutenção e confiabilidade do software. Essas restrições e exigências são atributos de qualidade do software, também chamados de requisitos não-funcionais [Kot98] [Net00]

De uma forma geral, podemos dizer que os requisitos do sistema nos remetem a questões próprias dos processos e informações utilizados pelos usuários, e portanto são requisitos menos técnicos no que se refere aos aspectos computacionais. Os requisitos do software, além de incorporar boa parte dos requisitos do sistema, também abrangem os aspectos computacionais necessários a atender as exigências e restrições citadas acima (requisitos não-funcionais).

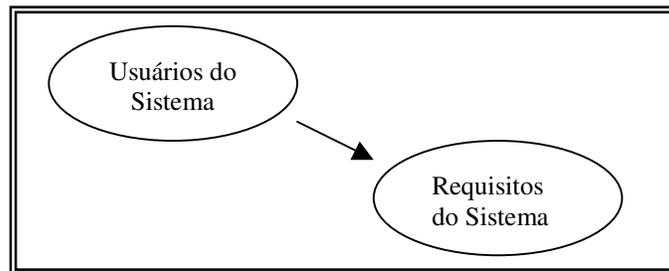


Figura 2.1 – Principal influência sobre os requisitos do sistema

Na *Figura 2.1* procuramos destacar que os usuários são a principal fonte de influência sobre os requisitos do sistema, embora outros fatores também influenciem os requisitos, como políticas organizacionais, estratégias de negócios, decisões governamentais etc. Os requisitos sofrem grande influência da forma como os usuários entendem os processos e informações pertinentes ao sistema em questão no seu contexto de trabalho.

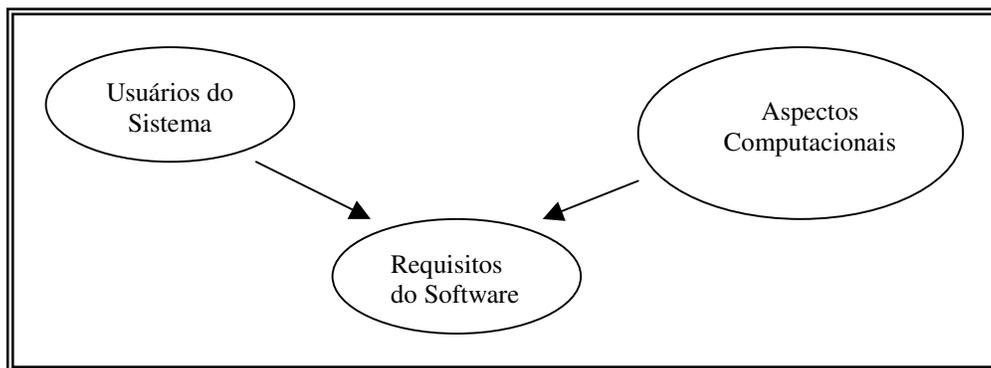


Figura 2.2 – Principais influências sobre os Requisitos do Software

Na *Figura 2.2* destacamos que os requisitos do software, além da influência dos usuários, também sofrem influências dos aspectos computacionais, tais como velocidade de

processamento, capacidade de memória, possibilidades de interface com os usuários através de periféricos, formas de acesso etc.

2.2 A Engenharia de Requisitos no Contexto do Desenvolvimento de Software

A Engenharia de Requisitos deve ser entendida dentro de um contexto mais amplo, ou seja, o contexto do desenvolvimento de software. Embora existam muitos modelos de representação das etapas de desenvolvimento de software (ciclo de vida de software) utilizamos o modelo clássico *Waterfall* [Pre95] para representar o escopo de atuação da Engenharia de Requisitos durante o desenvolvimento de software.

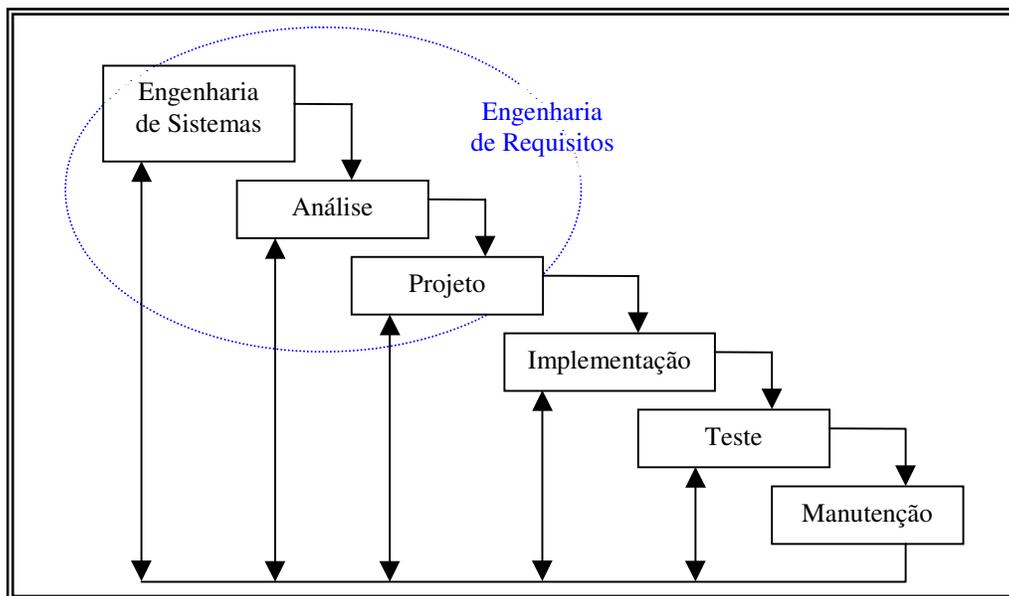


Figura 2.3 – A Engenharia de Requisitos no contexto do desenvolvimento de software

O modelo *Waterfall* foi escolhido por apresentar a seqüência das etapas do ciclo de vida de software de forma mais didática do que outros modelos. No entanto, ele não representa adequadamente o processo de iteração das etapas, conforme preconizado por metodologias de desenvolvimento de software mais recentes.

A Engenharia de Requisitos ocorre de forma intensiva nas primeiras etapas do ciclo de vida, abrangendo a engenharia de sistemas, análise e projeto, podendo se estender para as demais etapas dependendo do paradigma de engenharia de software adotado.

A importância da Engenharia de Requisitos no contexto de desenvolvimento de software advém do fato de que a correta identificação e documentação dos requisitos é fundamental para o sucesso do software. Pesquisas têm comprovado que muitos projetos de implementação de software têm falhado por problemas de requisitos de software [Boe81] [GAO92], ou seja, os requisitos obtidos muitas vezes são incompletos, mal entendidos ou ambíguos.

Além de ser um fator crítico de sucesso para implementação de softwares, no que se refere ao atendimento das reais necessidades dos usuários, os requisitos têm um forte impacto no custo total de projetos de software. Erros na identificação de requisitos, comuns nas etapas iniciais do desenvolvimento de software, quando detectados tardiamente apresentam um alto custo para sua correção, conforme apresentado na *Tabela 2.1*[Fau97].

Tabela 2.1 – Custos relativos de reparo do software nas diferentes etapas do ciclo de vida

Etapas	Custo Relativo de Reparo
Análise	1-2
Projeto	5
Implementação	10
Teste unitário	20
Teste do sistema	50
Manutenção	200

Podemos perceber pelos dados da *Tabela 2.1* que o custo de reparo de um erro nas fases iniciais tem um valor muito menor do que quando reparado nas etapas finais do ciclo de vida, obviamente isto tem um forte impacto no custo total do projeto de software.

2.3 As Atividades da Engenharia de Requisitos

Como citamos no início deste capítulo, consideramos a Engenharia de Requisitos composta por cinco principais atividades: elicitação, análise, especificação, validação e gerenciamento. Elicitação, análise, especificação e validação são atividades que ocorrem nessa seqüência. No entanto, são atividades que se relacionam entre si, e podem em determinados momentos assumir seqüências diferentes. A atividade de gerenciamento ocorre em paralelo a essas atividades.

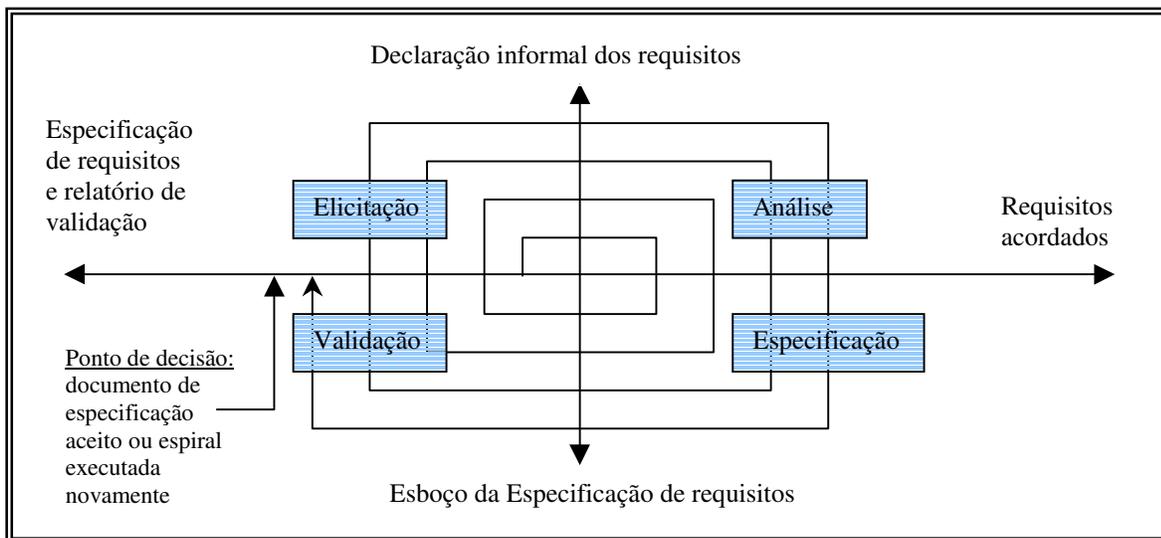


Figura 2.4 – Modelo espiral do processo de Engenharia de Requisitos [Kot98]

A Engenharia de Requisitos enquanto um processo de “construção” dos requisitos do software é um processo iterativo, a seqüência de execução de suas atividades pode ocorrer diversas vezes.

A Figura 2.4 apresenta um modelo espiral do processo de Engenharia de Requisitos fundamentado nas atividades de elicitação, análise, especificação e validação dos requisitos. A elicitação dos requisitos expande a declaração informal dos requisitos, oferecendo subsídios para análise dos requisitos que resultará nos requisitos acordados entre usuários e desenvolvedores, os requisitos acordados devem ser especificados na etapa de especificação dos requisitos, gerando os esboços da documentação dos requisitos. Os esboços de especificação devem ser validados gerando a especificação de requisitos final, juntamente com o relatório de validação.

Durante a validação é decidido se o documento de especificação é aceito ou não. Se não for aceito a espiral deve ser executada novamente, refinando os requisitos até se chegar a um documento de especificação aceito tanto por usuários como por desenvolvedores.

Para que possamos aprofundar nossa abordagem sobre a Engenharia de Requisitos apresentamos a seguir um detalhamento de cada uma das atividades mencionadas nesta seção.

2.3.1 Elicitação³

A elicitação de requisitos é a primeira atividade a ser desenvolvida na Engenharia de Requisitos. Na elicitação busca-se descobrir os requisitos do sistema, normalmente obscuros, vagos e confusos no início do desenvolvimento de um sistema de software [Joh96] .

Nessa etapa, usuários e desenvolvedores trabalham em conjunto para definir o problema a ser solucionado, enfocando principalmente os serviços que o sistema deve oferecer. No entanto, é comum os usuários não saberem exatamente o que eles desejam que seja implementado no sistema de software, o que pode fazer com que os requisitos definidos inicialmente não reflitam as reais necessidades dos usuários [Cas00].

Assim, esta atividade não envolve apenas perguntar ao usuário o que ele deseja. Ela requer uma análise cuidadosa da organização onde o software será implantado, uma análise do domínio da aplicação e uma análise dos processos de negócios onde o software será utilizado [Kot98].

A elicitação de requisitos é uma atividade complexa, principalmente devido ao alto grau de incerteza inerente a esta atividade. Na Seção 2.4 abordamos as principais dificuldades envolvidas na elicitação de requisitos.

³ O termo elicitação vem da palavra inglesa *elicitation*, que significa descobrir, desvendar algo obscuro. Embora o termo elicitação não exista no vocabulário da língua portuguesa ele vem sendo aceito e utilizado na comunidade de Engenharia de Requisitos como uma “tradução” do termo *elicitation*.

Segundo Kotonya e Sommerville [Kot98] a elicitação de requisitos, realizada de forma efetiva, deve abordar quatro dimensões:

- Entendimento do domínio da aplicação: significa conhecer a área onde o sistema é aplicado de uma forma geral. Este entendimento exige conhecimento gerais sobre a aplicação em questão. Por exemplo, se o sistema será aplicado numa área de seguros de automóveis, então devemos obter conhecimentos gerais sobre sinistros, apólices de seguro, mercado de automóveis etc.
- Entendimento do problema: significa conhecer os detalhes específicos do problema de um cliente em particular. Por exemplo, para um sistema de seguros de automóveis que será desenvolvido especificamente para um cliente, o entendimento do problema envolve conhecer como é o processo de atendimento ao cliente quando ocorre um sinistro, como ocorrerá o pagamento do conserto do automóvel do cliente, com quais oficinas a seguradora trabalha etc.
- Entendimento do negócio: normalmente sistemas contribuem de alguma forma com os objetivos e missão da organização onde eles estão inseridos. O entendimento do negócio significa conhecer como o sistema de software a ser desenvolvido interage e afeta os negócios da organização, e que tipo de contribuição ele irá proporcionar.
- Entendimento das necessidades e restrições das pessoas envolvidas no sistema: para obtermos este tipo de entendimento, é necessário conhecermos os processos que o sistema deverá suportar, pois boa parte destes processos são realizados pelas pessoas envolvidas no sistema.

2.3.2 Análise

A atividade de análise de requisitos está muito vinculada à elicitação, uma vez que na medida em que os requisitos vão se desvendando algum grau de análise sobre os mesmos é realizada. O objetivo da análise de requisitos é encontrar possíveis problemas na declaração informal dos requisitos obtida pela elicitação de requisitos [Kot98].

Podemos considerar que existe um processo de forte interação entre a elicitação e análise de requisitos, conforme demonstrado na *Figura 2.5*.

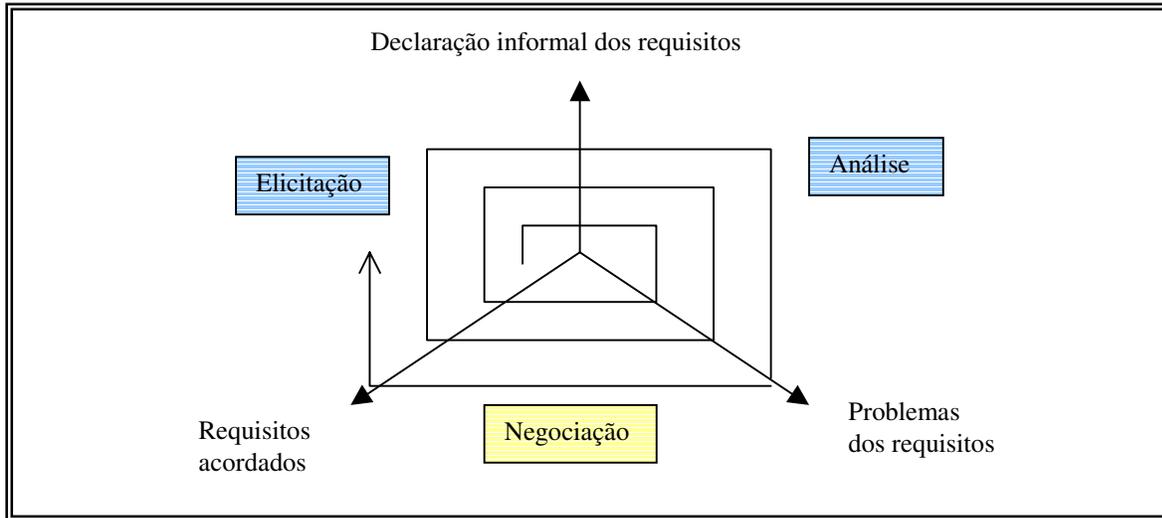


Figura 2.5 – Espiral representando o processo de interação entre elicitação e análise de requisitos [Kot98]

A *Figura 2.5* procura demonstrar a interação entre as atividades de elicitação e análise de requisitos, adicionando uma tarefa importante dentro da atividade de análise, a negociação.

Enquanto os requisitos vão sendo descobertos, alguma análise acaba ocorrendo. Durante a análise é comum encontrarmos problemas nos requisitos descobertos, tipicamente conflitos e ambigüidades, que devem ser negociados entre desenvolvedores e usuários, de forma a se obter um "acordo" sobre os requisitos obtidos.

Existem três elementos fundamentais sobre os requisitos⁴, que devem ser verificados na análise, que auxiliam na identificação de problemas:

⁴ Davis [Dav97] expande para 24 o número de elementos, ou atributos, relativos aos requisitos de software que são passíveis de avaliação.

- Verificação de Necessidade: a necessidade de um requisito obtido deve ser analisada. Alguns requisitos podem ser propostos, mas não contribuem com as metas de negócio da organização ou para o problema específico a ser endereçado pelo sistema.
- Verificação de consistência e completitude: consistência significa que nenhum requisito deve ser contraditório, e completitude significa que nenhum serviço ou restrição que sejam necessários ao sistema tenha sido omitido.
- Verificação de Possibilidade: os requisitos devem ser verificados no sentido de assegurar que eles são factíveis de serem implementados no sistema a ser desenvolvido, tanto em termos de orçamento como de tempo.

2.3.3 Especificação

A etapa de especificação é onde os resultados da elicitação e análise de requisitos serão transformados em documentos que organizam os requisitos do sistema. As metodologias de desenvolvimento de sistemas [Wil90][Rum91][Jac92][Boo94][Bri95][Jac98], em geral, oferecem boas técnicas de representação dos requisitos, que auxiliam os desenvolvedores na comunicação com os usuários [DeC93], principalmente na etapa de validação dos requisitos.

Muitas técnicas de modelagem e representação de requisitos de sistemas, que em síntese são técnicas de especificação de requisitos, têm sido propostas nos últimos 20 anos, culminando num esforço de padronização no final dos anos 90, como por exemplo as padronizações definidas em *Fusion* [Col94][Ama94] e UML⁵ [UML97a][UML97b][UML97c][UML97d], que procuraram unificar várias técnicas de modelagem.

⁵ *Unified Modeling Language*

Através da especificação dos requisitos é possível se estabelecer um veículo efetivo de comunicação com os usuários do sistema, de forma que o entendimento do problema obtido pelos desenvolvedores possam ser discutidos e elaborados em conjunto com os usuários.

A especificação de requisitos resulta em documentos que organizam os requisitos obtidos, estes documentos, também chamados de artefatos de software [Pre95][UML97a], assumem um papel fundamental no desenvolvimento de sistemas, principalmente em sistemas que possuem várias versões e que envolvem grandes equipes [Par86]. Dentre os benefícios obtidos pelos documentos gerados podemos citar [Fau97]:

1. O documento de especificação é o veículo básico de comunicação entre desenvolvedores e usuários sobre o que deve ser construído;
2. O documento de especificação registra os resultados da análise do problema (obtido através da elicitação e análise dos requisitos);
3. O documento de especificação define quais propriedades o sistema deve ter e quais são as restrições impostas em seu projeto e implementação;
4. O documento de especificação é a base para estimativas de custo e cronograma;
5. O documento de especificação é a base para o desenvolvimento do plano de teste do sistema;
6. O documento de especificação oferece uma definição padrão de comportamento esperado pelos profissionais envolvidos na manutenção do sistema;
7. O documento de especificação é utilizado para registrar mudanças na engenharia do sistema.

2.3.4 Validação

A validação é a etapa final da Engenharia de Requisitos, onde o objetivo é verificar e validar os requisitos especificados. A etapa de validação procura certificar que os requisitos acordados e especificados representam uma descrição aceitável do sistema a ser construído [Kot98].

A validação dos requisitos envolve tanto usuários como desenvolvedores, que analisam os requisitos buscando identificar possíveis problemas, omissões e ambigüidades. Os principais problemas descobertos durante a validação dos requisitos são:

- Não atendimento a padrões de qualidade;
- Requisitos descritos de forma pobre, os quais levam a ambigüidade;
- Erros na modelagem do problema ou sistema;
- Requisitos conflitantes não identificados durante a etapa de análise.

Estes problemas devem ser sanados antes que o documento de especificação seja aprovado e utilizado para o desenvolvimento do sistema. Alguns problemas são resolvidos com a correção do documento de especificação, no entanto, outros problemas podem levar a uma nova execução da espiral da Engenharia de Requisitos, levando a uma nova elicitação, análise e especificação.

O principal problema enfrentado na validação dos requisitos é que não existe um documento que possa ser usado como base para a validação, uma vez que o próprio documento de especificação, que será validado, é o primeiro parâmetro formal de referência sobre o sistema a ser construído. A *Figura 2.6* representa as entradas e saídas envolvidas no processo de validação dos requisitos.

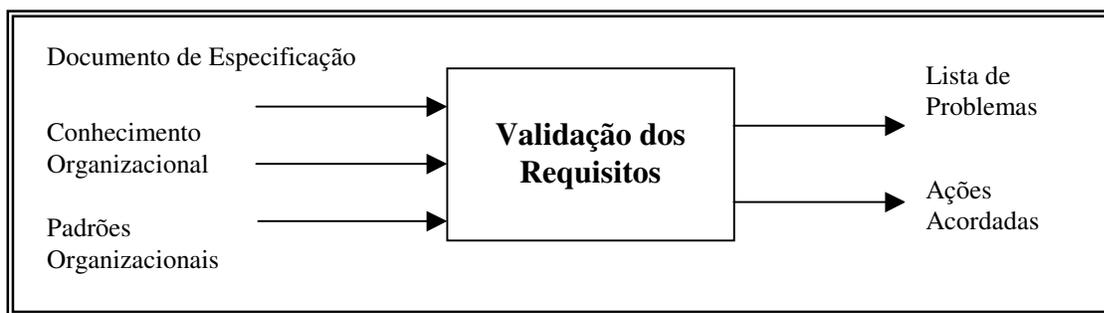


Figura 2.6 - O processo de Validação dos Requisitos [Kot98]

O documento de especificação deve ser um documento que esteja em conformidade com os padrões organizacionais, representando os requisitos negociados e acordados entre usuários e desenvolvedores. O conhecimento organizacional, embora não seja uma "entrada tangível" no

processo de validação, é muito importante, pois está muito ligado à cultura e às estruturas organizacionais onde o sistema será construído.

Como saída o processo de validação deve gerar uma lista de problemas e ações acordadas. A lista de problemas deve relatar todos os problemas identificados no documento de especificação, classificando-os de alguma forma, como por exemplo: ambigüidade, inconsistência etc. As ações acordadas são ações a serem executadas a partir dos problemas relacionados na lista de problemas.

2.3.5 Gerenciamento

O gerenciamento de requisitos é um processo que se desenvolve em paralelo a todas as outras atividades citadas anteriormente. Neste processo o objetivo é gerenciar as mudanças que ocorrem nos requisitos do sistema [Kot98].

Os requisitos do sistema se alteram com o decorrer do tempo, geralmente devido a mudanças no ambiente do sistema e um melhor entendimento do usuário sobre as suas reais necessidades.

As principais questões endereçadas no gerenciamento de requisitos são:

1. Gerenciamento de mudanças em requisitos já acordados;
2. Gerenciamento dos relacionamentos entre os requisitos;
3. Gerenciamento das dependências entre os documentos de requisitos e outros documentos produzidos durante o processo de engenharia de software.

Mudanças nos requisitos ocorrem durante todo o tempo. Assim, mesmo quando os requisitos estão sendo elicitados, analisados, especificados e validados, eles podem estar mudando.

As mudanças nos requisitos são inevitáveis e não significa que o processo de Engenharia de Requisitos adotado tenha sido falho [Kot98]. Essas mudanças resultam de uma combinação de fatores, tais como:

- Inconsistências, conflitos e falhas nos requisitos;
- Evolução do conhecimento do usuário sobre o sistema em desenvolvimento;
- Problemas de custos, cronogramas ou técnicos;
- Mudanças na prioridade do cliente;
- Mudanças ambientais;
- Mudanças organizacionais.

2.4 Dificuldades Essenciais na Elicitação de Requisitos

Identificar corretamente os requisitos do software não é uma tarefa trivial, em parte devido à natureza abstrata do software. Para entendermos melhor as dificuldades enfrentadas na elicitação dos requisitos podemos analisá-las a partir de dois grandes grupos [Bro87][Fau97]: dificuldades acidentais e dificuldades essenciais.

As dificuldades acidentais são aquelas oriundas da falta de controle sobre aquilo que precisa ser construído, dentre as quais podemos destacar: pouco esforço despendido no levantamento de informações junto ao usuário, documentação pobre sobre o requisitos obtidos, pouca revisão dos requisitos obtidos, especificações incorretas dos requisitos e tendência de iniciar logo o processo de desenvolvimento do software.

As dificuldades essenciais são aquelas inerentes à elicitação dos requisitos do software, dentre as quais podemos destacar: dificuldade do usuário em saber efetivamente o que ele quer de um sistema de software, dificuldade de comunicação entre usuário e desenvolvedor e a natureza mutante dos requisitos.

As dificuldades acidentais podem ser consideradas como mais fáceis de serem superadas. A adoção de um processo sistemático que oriente a elicitação, análise, especificação, validação e gerenciamento dos requisitos tende a solucionar, ou pelo menos minimizar significativamente, os problemas dessa categoria.

No entanto, as dificuldades essenciais são mais difíceis de serem superadas, uma vez que fazem parte da natureza do software, que é abstrata, maleável e complexa. A adoção de um processo sistemático para a Engenharia de Requisitos, principalmente no que se refere à especificação, validação e gerenciamento dos requisitos, também poderá ajudar na superação das dificuldades essenciais.

Porém, a problemática que naturalmente existe no processo de comunicação e compreensão humana, que está no cerne da elicitação de requisitos, necessitará de uma abordagem que leve em consideração fatores como o contexto em que as pessoas exercem suas atividades e reconhecem os objetos que lhes são pertinentes, o histórico de evolução dessas atividades e seus instrumentos de mediação, e outros aspectos de relevância social e psicológica que afetam os usuários do software a ser desenvolvido.

Dessa forma, entendemos que as dificuldades essenciais enfrentadas na elicitação de requisitos não poderão ser resolvidas por uma abordagem puramente tecnológica, uma vez que os aspectos sociais assumem grande importância nessa atividade [Gog97]. A maioria dos softwares é desenvolvida sem nenhum auxílio das ciências sociais (como psicologia, sociologia, antropologia etc.), não abordando de forma sistemática as necessidades do usuário, tanto em nível individual como organizacional.

Faulk [Fau97] destaca que as pessoas (usuários do sistema) normalmente não sabem o que elas realmente desejam em termos da funcionalidade a ser embutida no software. Isto não significa que as pessoas não têm uma idéia geral do que elas gostariam que o software fizesse, mas que no início do processo de desenvolvimento elas não têm uma idéia precisa e detalhada de quais funcionalidades o software deverá ter.

Vinculada à dificuldade inicial em se compreender o que o software deve fazer, está a dificuldade de se comunicar os requisitos efetivamente, tanto entre usuários e desenvolvedores do software, como entre os próprios desenvolvedores (analistas, projetistas e programadores).

Brooks [Bro87] destaca que as estruturas conceituais de sistemas de software são complexas, arbitrárias, mutantes e de difícil visualização. O processo de desenvolvimento de software começa com a Engenharia de Requisitos, que tem como primeira atividade a elicitação dos requisitos, neste momento as dificuldades essenciais, apontadas por Brooks, já tornam-se perceptíveis.

Quando Brooks aponta a complexidade como uma propriedade essencial do software ele se refere à quantidade de elementos, ou componentes, que formam o todo, e a necessidade de interação entre esses elementos. Acima do nível de instrução de um programa, os elementos constituintes de um software já tornam-se diferentes entre si, o que nos leva a um número expressivo de elementos distintos, e que ainda precisam interagir entre si. Muitos problemas clássicos de desenvolvimento de software são derivados dessa complexidade e de seu crescimento não-linear, que aumenta na medida em que mais funcionalidade vai sendo requerida dos softwares. Naturalmente, essa complexidade tem grande influência na dificuldade de comunicação entre as pessoas envolvidas na definição dos requisitos do software.

Outro fator que contribui para a complexidade a ser enfrentada no processo de elicitação de requisitos é a necessidade de se construir um produto (software) que precisa estar em conformidade com as visões de diferentes pessoas, ou grupos de pessoas, envolvidas no processo. Os diferentes anseios, motivações e desejos criam uma pluralidade de necessidades, de difícil integração, dificultando ainda mais a compreensão dos requisitos do software.

Brooks ainda destaca que a entidade software está constantemente sendo pressionada para mudanças, o que nos remete, em termos do processo de elicitação de requisitos, à natureza mutante dos requisitos do software. Isto ocorre devido a dois fatores: em parte porque softwares embutem funções de um sistema, e as funções de um sistema são as que mais sofrem pressões de

mudança (em contraste aos dados que são mais estáveis); e em parte porque um software é um produto de alta maleabilidade.

A última dificuldade essencial apontada por Brooks é sobre a invisibilidade da entidade software. Embora muito progresso tenha sido feito no sentido de se conseguir visualizar as estruturas de um software por meio de modelos abstratos (principalmente modelos gráficos, que fazem uso de diagramas), essas estruturas ainda continuam inerentemente invisíveis, uma vez que a “materialização” de um software não pode ser inerentemente embutida no espaço. Essa propriedade “invisível” do software acarreta, obviamente, grandes dificuldades na definição de seus requisitos.

Conclusão do Capítulo

Neste capítulo procuramos oferecer uma visão geral da Engenharia de Requisitos, partindo da definição de requisitos em duas principais vertentes: requisitos funcionais e requisitos não-funcionais. Em seguida, apresentamos um resumo das principais atividades da Engenharia de Requisitos, dividida em elicitação, análise, especificação, verificação e gerenciamento de requisitos.

Ao final do capítulo apresentamos uma discussão sobre as principais dificuldades enfrentadas na elicitação de requisitos, levando em conta a classificação de Brooks para as dificuldades enfrentadas na Engenharia de Software, divididas em dois grupos: dificuldades acidentais e dificuldades essenciais. Destacamos que as dificuldades essenciais são mais difíceis de serem superadas e afetam significativamente a qualidade e confiabilidade dos requisitos elicitados.

A partir do exposto no capítulo, fica clara a importância da elicitação de requisitos como primeira atividade da Engenharia de Requisitos, onde começam a ser desvendadas as reais necessidades dos usuários sobre o sistema de software a ser implementado.

Essa visão geral da Engenharia de Requisitos oferece um contexto mínimo para discutirmos no próximo capítulo algumas técnicas de elicitação de requisitos que vêm sendo utilizadas pela comunidade da área.

Capítulo 3

Técnicas Empregadas na Elicitação de Requisitos

3. Técnicas Empregadas na Elicitação de Requisitos

Esse capítulo apresenta uma coleção de técnicas que vêm sendo utilizadas pelos engenheiros de requisitos durante a elicitação de requisitos do software. Esta coleção não pretende esgotar todas as técnicas de elicitação de requisitos propostas até o momento, mas sim destacar aquelas que vêm oferecendo importantes contribuições para a Engenharia de Requisitos, em geral, e para a elicitação de requisitos, em particular.

3.1 Introspecção

Segundo Goguen [Gog97], a introspecção é provavelmente a primeira e talvez a mais óbvia forma de se tentar entender quais propriedades um sistema deva ter para se alcançar um conjunto de objetivos. A introspecção é o ato de “imaginar” como deveria ser um sistema de software para atender a um determinado problema.

Ela leva em consideração um ponto de vista particular, por exemplo, o ponto de vista do desenvolvedor do sistema, que procura imaginar que tipo de sistema ele desejaria ter para realizar as tarefas que o usuário real irá executar.

Esta abordagem pode ser útil para se obter uma “imagem” inicial do software, mas um sério problema que ela apresenta é que o ponto de vista do desenvolvedor estará permeado de suas próprias experiências e conhecimentos, que provavelmente são distintos do usuário que utilizará o software, podendo levar a uma concepção do software muito distante do que o usuário esteja esperando.

Vantagens: Permite a elaboração de uma concepção de sistema rapidamente.

Desvantagens: Está fortemente vinculada a experiência do desenvolvedor em relação ao problema a ser resolvido.

3.2 Observação

A observação é uma técnica muito utilizada nas ciências sociais, como sociologia, antropologia e psicologia, e vem sendo empregada também para a elicitación de requisitos de software. O princípio básico desta técnica é obter informações a partir da observação de tarefas executadas pelo usuário (ou por um grupo de usuários), com um mínimo de interferência da parte do desenvolvedor.

Uma técnica de observação muito empregada nas ciências sociais e que vem sendo estudada pela comunidade de Engenharia de Requisitos, com potencial promissor para auxiliar no processo de elicitación de requisitos, é denominada etnografia [Ack92] [Har91] [Hug95]. A etnografia envolve um observador inserido dentro de uma comunidade, registrando de forma detalhada os fatos e ações em curso [Kot98] [Gog97].

Em [Hug95] é sugerido que a etnografia possa ser apresentada a partir de três pontos de vista:

1. **Ambiente de trabalho.** Descreve o contexto e a localização física onde o trabalho se realiza, e como as pessoas utilizam objetos para executar suas tarefas.
2. **Perspectivas organizacionais e sociais.** Descrevem a experiência de trabalho do dia a dia, conforme vista pelos diferentes agentes que estão envolvidos no trabalho.
3. **Fluxo de trabalho.** Descreve o trabalho realizado a partir de uma série de atividades com as informações fluindo de uma atividade para a outra.

Vantagens: Os requisitos podem ser obtidos sem interferência de comunicação, uma vez que estarão sendo percebidos pelo observador no ambiente do próprio usuário. A observação ocorre no cotidiano do usuário, onde as situações de trabalho (rotinas) que envolvem as pessoas que vão fazer uso do sistema emergem naturalmente.

Desvantagens: o período de observação para cobrir as principais rotinas de trabalho do usuário pode ser longo, uma vez que muitas situações importantes podem ocorrer apenas ocasionalmente.

3.3 Questionário

O questionário é um instrumento que pode ser útil na elicitação de requisitos quando se deseja obter informações de um grande número de pessoas. Outro aspecto interessante do uso de questionários, é que eles podem ser preparados de tal forma que suas respostas possam ser facilmente tabuladas, possibilitando uma análise estatística dos dados obtidos.

Uma preocupação importante que deve-se ter quando se elabora um questionário é com relação ao perfil de seus respondentes. Em [Fou94] é proposta uma metodologia para a utilização de questionários, dividida nas seguintes etapas:

- 1. Preparação do questionário.** Envolve a identificação do tipo de informações que se deseja obter, identificação do perfil dos respondentes, escolha do formato apropriado do questionário (respostas de múltipla escolha, descritivas ou dissertativas), classificação das questões em grupos coesos etc.
- 2. Identificação dos respondentes.** Esta é uma etapa opcional, aplicada para casos onde se deseja ter a identificação de quem respondeu o questionário.
- 3. Distribuição do questionário.** O questionário deve ser distribuído junto com instruções detalhadas de como preenchê-lo. Prazos de devolução devem ser explicitados.

4. Análise das respostas. As respostas devem ser organizadas e tabuladas, consolidando as informações obtidas no questionário. Sugere-se enviar uma cópia dos resultados obtidos para os respondentes.

Embora o questionário seja uma ferramenta útil quando se quer atingir um grande número de pessoas, ela é inflexível, não permitindo uma análise de questões subjetivas, que podem ser importantes para o entendimento do contexto analisado.

Vantagens: Pode trazer informações pertinentes ao sistema estudado vinda de um grande número de pessoas envolvidas no mesmo. Facilita uma análise estatística das informações obtidas.

Desvantagens: O grau de flexibilidade que o questionário dá para as respostas dos envolvidos muitas vezes é pequeno, tendendo a ser rígido, dificultando a obtenção de informações mais subjetivas. Muitas vezes as pessoas não se sentem à vontade de colocar suas opiniões por escrito.

3.4 Entrevista

A entrevista é uma técnica muito comum utilizada em processos de elicitação de requisitos de software. Existem basicamente dois tipos de entrevista [Kot98] [Gog97]: entrevista fechada e entrevista aberta. Estes dois tipos de entrevista, normalmente, ocorrem de forma individual e privada.

A entrevista fechada consiste num conjunto de perguntas que o entrevistador deve fazer à pessoa entrevistada; trata-se de um tipo rígido de entrevista, havendo pouca margem para discussões de outros assuntos que não aqueles abordados diretamente pelo conjunto de perguntas. As perguntas são definidas previamente.

A entrevista aberta é um tipo mais flexível de entrevista. O entrevistador indica os principais temas a serem discutidos durante a entrevista, que servem como referências iniciais para a

entrevista. No entanto, o entrevistador tem liberdade para falar de assuntos correlatos aos temas iniciais da entrevista. Não se espera resultados conclusivos nesse tipo de entrevista, mas sim gerar discussões sobre os temas, de forma que auxiliem o entrevistador (engenheiro de requisitos) a obter um entendimento mais amplo sobre o sistema em questão.

Vantagens: É flexível na obtenção das informações, sendo adequada para se obter informações de carácter subjetivo. Aproxima o engenheiro de requisitos do usuário do sistema, fazendo com que o usuário se sinta participativo no processo de desenvolvimento do sistema.

Desvantagens: Pode demandar muito tempo se muitas pessoas precisarem ser entrevistadas, uma vez que a entrevista normalmente ocorre de forma individual, com cada usuário do sistema. A tabulação das informações obtidas a partir das entrevistas, principalmente na de tipo aberto, costuma ser trabalhosa.

3.5 JAD⁶

JAD é uma técnica que foi desenvolvida pela IBM⁷ (Canadá) no final da década de 1970. Trata-se de uma técnica⁸ orientada ao trabalho em grupo, que oferece um ambiente para clientes e engenheiros de software trabalharem como um time, compartilhando informações e idéias sobre um determinado tema ou assunto, através de sessões de *brainstorm* mediada por um facilitador [Chi95][Dam97].

Os defensores desta técnica argumentam que ela auxilia na quebra das barreiras de comunicação existente entre pessoas vindas de diferentes departamentos dentro de uma organização, que estejam envolvidas num mesmo projeto de software.

O processo JAD engloba quatro etapas elementares [Fou94]:

⁶ *Joint Application Development*

⁷ *International Business Machines*

⁸ Uma especialização do JAD é o JRP (*Joint Requirements Planning*), que se baseia na mesma estrutura de trabalho do JAD, mas com um enfoque mais direcionado para *o que* o sistema deve fazer [Kul00].

1. Orientação inicial;
2. Familiarização com a área de negócio e seu sistema de aplicação relacionado;
3. Preparação do material para o *workshop*⁹;
4. Condução do *workshop*.

A etapa de orientação inicial trata de questões de definição de aspectos globais do projeto, especificando itens como: finalidade do projeto, escopo do projeto e áreas funcionais envolvidas, objetivos a serem alcançados pela sessão JAD, suposições técnicas e de negócio que afetam o projeto, fatores críticos para o sucesso etc.

A familiarização com a área de negócio e seu sistema de aplicação relacionado é a etapa em que é realizada uma análise dos processos atuais de negócio e de seus fluxos de informações. As informações obtidas durante esta fase auxiliarão o facilitador da sessão JAD (que é o líder do processo) a entender melhor o contexto de aplicação do projeto.

A terceira etapa, preparação do material para o *workshop*, consiste na elaboração de um modelo simplificado do sistema a ser desenvolvido. Este modelo será desenvolvido a partir da lista de tarefas de negócio identificadas na etapa anterior. Dependendo da natureza do projeto, o modelo construído pode consistir de esboços de telas e relatórios que serão revisados durante o *workshop*.

A condução do *workshop* iniciará com a apresentação dos objetivos do mesmo e da estrutura básica dentro da qual os participantes da equipe deverão operar. O facilitador assumirá um papel chave neste processo, que irá conduzir as discussões e mediar conflitos durante a sessão. Na medida que as discussões progredirem os requisitos que vão sendo levantados deverão ser transcritos pelo facilitador ou um auxiliar.

Vantagens: Confronta os diversos pontos de vista dos participantes do sistema, usuários operativos e gerentes, facilitando o ajuste do entendimento de pontos polêmicos do sistema.

⁹ Também chamado de sessão JAD

Desvantagens: Pode inibir os usuários operativos, cujos pontos de vista podem ser conflitantes com os da gerência. Os conflitos podem ser de difícil solução, podendo estender a sessão. O facilitador deve ser alguém experiente na resolução de conflitos.

3.6 Análise de Protocolo

A análise de protocolo é uma técnica de elicitação de requisitos que pode ser aplicada a partir da narração de uma tarefa realizada pelo usuário. Enquanto o usuário está executando a tarefa ele vai falando o que está fazendo [Gog97]. Os proponentes desta técnica acreditam que este processo pode ser considerado como uma verbalização direta de processos cognitivos específicos [Eri84].

Há três variações possíveis desta técnica [Sil94]:

- Análise de protocolo estilo “Pensando Alto” (forma mais comum): uma tarefa é proposta para o usuário que está sendo entrevistado e, na medida em que realiza a mesma, ele vai verbalizando o que está pensando.
- Análise de Protocolo estilo “Falando Alto”: neste caso o usuário deve falar “em voz alta” qualquer coisa que ele esteja dizendo silenciosamente para si enquanto executa a tarefa.
- Análise de Protocolo estilo “Relato Retrospectivo”: somente após realizar a tarefa o usuário irá relatar o que fez.

Vantagens: Duas formas de comunicar informações sobre o sistema ocorrem simultaneamente, o fazer e o falar o que está fazendo. Requisitos podem ser capturados e ajustados comparando as duas formas de comunicação. O fazer pode completar o falar e vice-versa.

Desvantagens: O usuário pode se sentir inibido de falar enquanto faz, podendo também se desconcentrar no que está fazendo enquanto tenta explicar verbalmente.

3.7 Análise de Discurso

A análise de discurso é uma técnica originária das ciências lingüísticas, que é utilizada no estudo de estruturas lingüísticas, normalmente maiores do que uma sentença. Essas estruturas lingüísticas são chamadas de unidades de discurso [Gog97].

Como uma unidade estrutural, a unidade de discurso deve ter propriedades bem definidas, com limites definidos (onde inicia e onde termina) e estrutura interna passível de descrição.

A técnica é aplicada sobre um texto, que será dividido em unidades de discurso. Para cada unidade, buscam-se os objetos existentes e as relações destes com o domínio do problema [Sil94]. Essas relações podem ajudar na identificação de requisitos importantes para o sistema a ser implementado.

Vantagens: Diversos relacionamentos podem ser verificados entre as unidades de discursos. Do relacionamento entre essas unidades podem emergir novas informações, imperceptíveis no texto original.

Desvantagens: Se o texto a ser analisado estiver desorganizado e confuso o resultado da análise de discurso pode gerar muita ambigüidade. O texto é a única fonte de informações.

3.8 Prototipação

A prototipação consiste na construção de um protótipo do sistema a ser implementado. O protótipo é freqüentemente usado para elicitar e validar requisitos do sistema [Kot98]. Um requisito essencial para um protótipo, é que deve ser possível desenvolvê-lo de forma rápida, tal que ele possa ser utilizado durante o processo de Engenharia de Requisitos. O protótipo não é o sistema real, mas apenas uma simulação do sistema que será construído de fato.

Benefícios que normalmente estão associados ao uso da prototipação são:

- O protótipo ajuda na identificação de falhas e sucessos do sistema, antes que se comece a construção de fato.
- É uma forma efetiva de se desenvolver a interface do usuário do sistema [Dix93][Joh92]. Se um protótipo for desenvolvido como parte da Engenharia de Requisitos, isto pode reduzir custos futuros na implementação do sistema.
- É possível usar o protótipo do sistema para desenvolver testes do sistema, os quais podem ser usados posteriormente no processo de validação do sistema.
- Protótipos ajudam a revelar requisitos incompletos e inconsistentes.

Vantagens: O protótipo oferece uma ajuda efetiva na comunicação entre usuário e engenheiro de requisitos, devido a seu forte apelo visual.

Desvantagens: Existe o risco de se usar a estrutura do protótipo como base de implementação da primeira versão do sistema.

3.9 Casos de Uso

A utilização de casos de uso (*use cases*) para a elicitação de requisitos do sistema vem recebendo interesse crescente da comunidade de Engenharia de Requisitos, e sendo bem aceita por muitos metodologistas [Rum97] [Coc00]. Esta abordagem enfatiza a importância da captura dos requisitos do sistema do ponto de vista dos atores que interagem com ele.

A utilização de casos de uso para a captura de requisitos tem sido disseminada a partir do método *Objectory*, proposto por Ivar Jacobson [Jac92]. Casos de uso também assumem um papel

importante na UML¹⁰ e USDP¹¹ [Jac98] [UML97d], que vêm sendo amplamente divulgadas nos últimos anos.

Casos de uso são utilizados para representar a funcionalidade de um sistema. Em UML um caso de uso é definido como um conjunto de seqüências de ações que um sistema realiza produzindo um resultado de valor observável para um ator particular [Eri98]. Um ator inicia uma interação com o sistema, ou seja, um caso de uso, com o objetivo de atingir alguma meta [Coc00]. Casos de uso são representados graficamente através de diagramas de casos de uso, conforme mostrado na *Figura 3.1*.

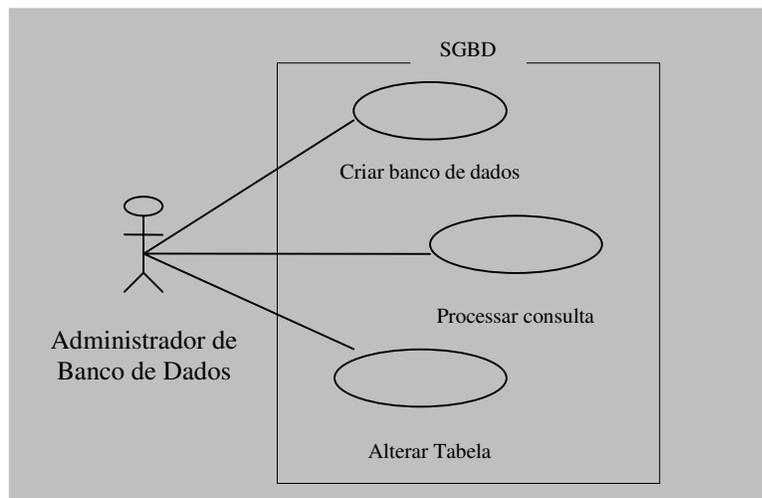


Figura 3.1 – Diagrama de casos de uso

A *Figura 3.1* apresenta um exemplo de digrama de casos de uso, onde o administrador de banco de dados é o ator que interage com o SGBD¹² por meio dos casos de uso “criar banco de dados”, “processar consulta” e “alterar tabela”.

Casos de uso precisam ser descritos, uma vez que através do digrama de casos de uso obtemos apenas uma visão dos casos de uso existentes, com seus relacionamentos, e dos atores envolvidos. Em [Rum97] é apresentada uma sugestão para descrição de casos de uso, conforme

¹⁰ Unified Modeling Language

¹¹ Unified Software Development Process

¹² Sistema Gerenciador de Banco de Dados

apresentado na *Figura 3.2*. A descrição de casos de uso sugerida por Rumbaugh traz detalhes da execução do caso de uso que não podem ser demonstrados no diagrama de casos de uso.

Em [Sut98] é apresentado um meta-esquema para a modelagem de requisitos baseada em casos de uso, onde são destacados elementos importantes para uma melhor compreensão do caso de uso. A *Figura 3.3* apresenta o meta-esquema proposto por Sutcliffe.

Caso de Uso: Processar Consulta.

Resumo: O administrador de banco de dados elabora uma expressão de consulta SQL (*Structured Query Language*) e envia para o SGBD processar a consulta, que irá retornar com os dados solicitados na consulta.

Atores: Administrador de Banco de Dados.

Pré-condições: Todas as Tabelas envolvidas na expressão de consulta estarem disponíveis no momento do processamento da consulta.

Descrição: A expressão de consulta é interpretada pelo processador de consultas do SGBD, que irá verificar a sintaxe da expressão e os parâmetros da consulta (nome das Tabelas envolvidas para eventuais junções, condição de seleção e campos para projeção). A partir da análise sintática da expressão de consulta, o SGBD deverá abrir as Tabelas envolvidas, realizar as junções necessárias e buscar os campos para seleção e projeção. Filtrar as tuplas e as colunas a partir dos parâmetros estabelecidos na expressão de consulta e apresentar o resultado no formato de uma Tabela (relação gerada pela consulta).

Exceções: Problema na abertura das Tabelas envolvidas na consulta. Nome de campos estabelecidos na expressão de consulta não encontrados nas Tabelas envolvidas. Solicitar revisão na expressão de consulta.

Pós-Condições: Todas as Tabelas utilizadas na consulta devem ser fechadas após o processamento da consulta.

Figura 3.2 – Exemplo de descrição do caso de uso “Processar Consulta”

O meta-esquema proposto por Sutcliffe apresenta a idéia de que múltiplos cenários podem ser gerados a partir de um caso de uso. Além do relacionamento direto com cenários, o caso de uso também se relaciona diretamente com vários outros elementos (propriedade, agente, ação e meta), que contribuem para um desenvolvimento mais acurado da modelagem de requisitos baseada em casos de uso. Os elementos que não se relacionam diretamente com o caso de uso (evento, objeto, estrutura do objeto, transição de estado e estado), na medida em que vão sendo obtidos durante o

processo de Engenharia de Requisitos, auxiliam na formação do contexto em que o caso de uso é executado, e portanto contribuem para o refinamento da modelagem dos requisitos.

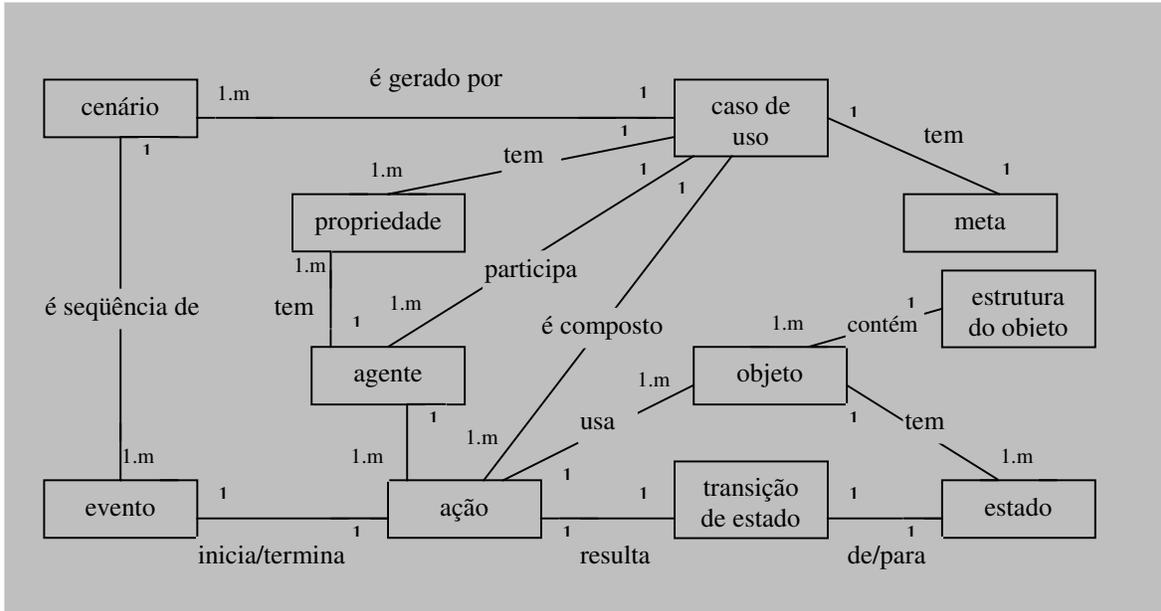


Figura 3.3 – Meta-esquema para modelagem de requisitos baseada em casos de uso [Sut98]

Vantagens: Uma vez identificados adequadamente os casos de uso, os requisitos do sistema estão praticamente definidos. Oferece uma estrutura de organização para o processo de elicitação e análise dos requisitos. Os requisitos podem ser apresentados para o usuário por meio de diagramas de fácil entendimento, o que facilita o processo de comunicação entre usuário e engenheiro de requisitos.

Desvantagens: Cada caso de uso precisa ser descrito detalhadamente em separado, uma vez que sua representação pictórica com o diagrama de casos de uso oferece poucas informações sobre os requisitos.

3.10 Cenários

Muitos usuários preferem descrever a sua rotina de trabalho exemplificando situações reais do seu dia a dia ao invés de produzir descrições abstratas e genéricas de tais processos. Em outras

palavras, eles preferem descrever os processos de trabalho em que estão envolvidos através de cenários. Cenários podem ser vistos como instâncias de casos de uso.

Conforme apresentado em [Lei97], cenários descrevem situações em nível do macrosistema, evoluem durante o desenvolvimento do software e são expressos através de linguagem natural. Breitman [Bre98] descreve a estrutura de um cenário por meio da concepção de elementos como contexto, episódio, objetivo, ator e recurso.

Para sistemas complexos um grande número de cenários normalmente é requerido. Segundo [Kot98], cenários podem ser descritos de várias formas, devendo incluir pelo menos as seguintes informações:

1. Uma descrição do estado do sistema antes do cenário em questão;
2. Um fluxo normal de eventos no cenários;
3. Exceções do fluxo normal de eventos;
4. Informações sobre outras atividades que podem estar ocorrendo ao mesmo tempo;
5. Uma descrição do estado do sistema após a realização do cenário.

Este conjunto de informações evidencia a relação tipo-instância que existe entre caso de uso e cenários. Assim, um conjunto de cenários similares pode ser abstraído em um caso de uso.

Vantagens: Podem ser facilmente descritos pelos usuários do sistema. Auxiliam na identificação de situações de exceções do sistema.

Desvantagens: Grandes sistemas precisam de um número elevado de cenários a serem descritos. O grande volume de informações gerado na descrição dos cenários pode demandar esforço considerável para a extração dos requisitos.

Conclusão do Capítulo

Procuramos destacar neste capítulo técnicas importantes que vêm sendo empregadas para a elicitação de requisitos. Explicamos objetivamente cada uma das técnicas, suas formas de aplicação e enfoque principal. Destacamos também as principais vantagens e desvantagens de cada técnica.

Esse panorama atual das técnicas de elicitação de requisitos nos traz referências para analisarmos a metodologia de elicitação de requisitos proposta no Capítulo 5.

Capítulo 4

Teoria da Atividade

4. Teoria da Atividade

A Teoria da Atividade é uma corrente de pensamento importante dentro da Psicologia contemporânea, que oferece uma estrutura filosófica e interdisciplinar para estudar a formação dos processos mentais humanos, tendo como elemento chave a atividade humana, tanto no nível individual como no nível social.

Ela tem raízes históricas oriundas de três vertentes: a filosofia clássica alemã dos séculos XVIII e XIX (de Kant a Hegel); os escritos de Marx e Engels, que elaboraram inicialmente o conceito de atividade; e a Psicologia Soviética, fundada por Vygotsky, Leont'ev e Lúria. O termo "Teoria da Atividade" surgiu durante as décadas de 1920 e 1930, dentro da escola histórico-cultural soviética de psicologia [Min86][Kuu96][Kap97].

A Teoria da Atividade, tal como é conhecida hoje, tem como seus principais fundadores Vygotsky e Leont'ev. Os conceitos fundamentais desta teoria foram elaborados por Vygotsky, mas a consolidação e integração destes conceitos numa estrutura organizada deve-se a Leont'ev [Wer81][Leo78].

4.1 Princípios Básicos

A Teoria da Atividade é formada por um conjunto de princípios que constituem um sistema conceitual geral. A organização da Teoria da Atividade em princípios pode variar de autor para autor, os quais normalmente a dividem em cinco ou seis princípios. Os princípios básicos da Teoria da Atividade [Kap96][Kap97][Kap98], conforme adotamos neste trabalho, estão descritos a seguir.

(1) Princípio da unidade entre consciência e atividade.

É considerado o princípio fundamental da Teoria da Atividade, onde consciência e atividade são concebidas de forma integrada. A consciência¹³ pode ser entendida como um conjunto de aspectos psicológicos que são utilizados no âmbito racional, e a atividade como a interação humana com sua realidade objetiva. A formação dos processos mentais tem sua gênese na realização das atividades.

Este princípio declara que a mente humana emerge e evolui como um componente especial da interação humana com o seu ambiente (onde a consciência é o repositório dos resultados dessas interações), surgindo no processo de evolução para ajudar a espécie humana a sobreviver. Assim, a mente humana pode ser analisada e entendida somente dentro do contexto da atividade humana. Pode-se dizer que a consciência humana é abastecida pelas atividades que a pessoa realiza.

(2) Princípio da orientação a objetos.

Este princípio enfoca a abordagem da Teoria da Atividade para o ambiente no qual seres humanos interagem. Seres humanos vivem em ambientes que são significativos para eles. Estes ambientes são formados por entidades (objetos) que combinam todos os tipos de características objetivas, incluindo aquelas determinadas culturalmente, e que por sua vez influenciam as formas como as pessoas agem sobre essas entidades.

Em qualquer atividade que realizamos, nos defrontamos com objetos do mundo real, que de alguma forma influenciam nossa maneira de executar atividades.

(3) Princípio da estrutura hierárquica da atividade.

A Teoria da Atividade diferencia os procedimentos humanos em três níveis: atividade, ação e operação, levando em conta os objetivos para os quais estes procedimentos são orientados. A

¹³ Uma definição mais precisa do que seja a consciência humana ainda é tema de pesquisa dentro da própria Psicologia, e portanto procuramos não estabelecer uma definição rígida para a mesma.

atividade é orientada a motivos, a ação orientada a metas e a operação orientada a condições de realização.

Esta diferenciação permite que uma mesma atividade possa ser analisada por diferentes pontos de vista, levando em consideração a base sobre a qual a análise pretende ser realizada: motivos, metas ou condições (estes conceitos são discutidos na Seção 4.3).

(4) Princípio da internalização-externalização.

Este princípio enfoca os mecanismos básicos de origem dos processos mentais. Ele declara que processos mentais são derivados das ações externas¹⁴ através do curso da internalização. A internalização é o termo usado para descrever a conversão de processos e objetos materiais externos para processos executados no plano mental, ou ainda, no plano da consciência [Leo81]. A internalização ocorre a partir do contato com o ambiente em que a pessoa está inserida.

A externalização é o processo inverso da internalização, onde os processos mentais se manifestam através de atos, de tal forma que eles possam ser verificados e corrigidos se necessário.

(5) Princípio da mediação.

A atividade humana é mediada por ferramentas, tanto externas (como um machado ou um computador) quanto internas (como uma heurística ou um conceito). As ferramentas são “veículos” da experiência social e do conhecimento cultural.

Uma questão importante que permeia este conceito não é o fato de que com o uso de uma ferramenta uma atividade possa ser executada de maneira mais facilitada ou menos custosa, mas sim o fato de que na verdade uma nova atividade é criada quando passamos a utilizar um instrumento de mediação [Wer81].

¹⁴ Na Teoria da Atividade há uma diferenciação importante entre elementos externos ao plano mental e internos ao plano mental. Estes elementos podem ser ações, processos, objetos etc.

(6) Princípio do desenvolvimento.

De acordo com a Teoria da Atividade, entender um fenômeno significa conhecer como ele se desenvolveu até sua forma atual, pois ao longo do tempo ele sofre alterações. Compreender essas alterações auxilia no entendimento do seu estado atual. Este conceito nos remete a idéia de que a atividade humana é dinâmica, se alterando e se transformando ao longo da evolução humana.

Esses princípios não são idéias isoladas, eles estão intimamente ligados. A natureza da Teoria da Atividade é manifestada nesse conjunto de princípios.

4.2 O Conceito de Atividade

A palavra atividade tem sido a tradução que o mundo ocidental tem empregado para a palavra russa *deyatel'nost*. No entanto, *deyatel'nost* tem um significado muito mais profundo do que a palavra atividade empregada no nosso cotidiano. Dentro da psicologia Soviética a palavra *deyatel'nost* significa uma “unidade organizacional para a realização de uma função mental”, o que nos traz uma nova dimensão para o significado de atividade. É por este vértice que a Teoria da Atividade elabora o conceito de atividade. A seguir apresentamos uma série de considerações sobre os elementos envolvidos no conceito de atividade.

Segundo a Teoria da Atividade uma atividade embute uma forma de agir de um sujeito direcionada para um objeto. No nível individual, uma atividade possui três elementos: *sujeito*, *objeto e ferramenta de mediação*. O *sujeito* é o agente que atua sobre o objeto da atividade. O *objeto* é o elemento para o qual as ações da atividade estão direcionadas, um objeto pode ser algo material, ou algo menos tangível, como um plano ou um conceito qualquer.

O relacionamento recíproco entre o sujeito e o objeto da atividade é sempre mediado por uma ou mais *ferramentas* (também chamadas de artefatos de mediação), que podem ser instrumentos, sinais, procedimentos, máquinas, métodos, leis, formas de organização de trabalho etc. Ferramentas sempre possuem um papel de mediação, usadas no processo de transformação dos objetos [Kuu96].

A *Figura 4.1* representa a estrutura de relacionamento, no nível individual, entre o sujeito e o objeto no contexto de uma atividade (conforme concepção original de Vygotsky [Vyg94]), onde a ferramenta assume um papel importante na mediação entre eles. Através desta mediação algum resultado é obtido. Transformar um objeto em um resultado motiva a existência de uma atividade.

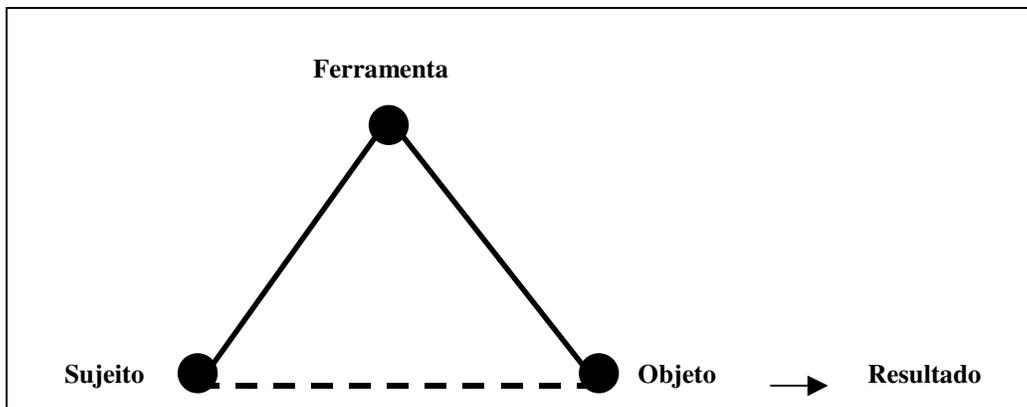


Figura 4.1 - Relacionamento mediado entre sujeito e objeto no nível individual [Kuu96]

Para exemplificar a estrutura acima considere a seguinte atividade: "codificar um programa". Nesse caso, o sujeito da atividade é um programador, uma das ferramentas de mediação é um editor de texto, o objeto a ser transformado é um algoritmo e o resultado é o programa fonte pronto para compilação (vide *Figura 4.2*).

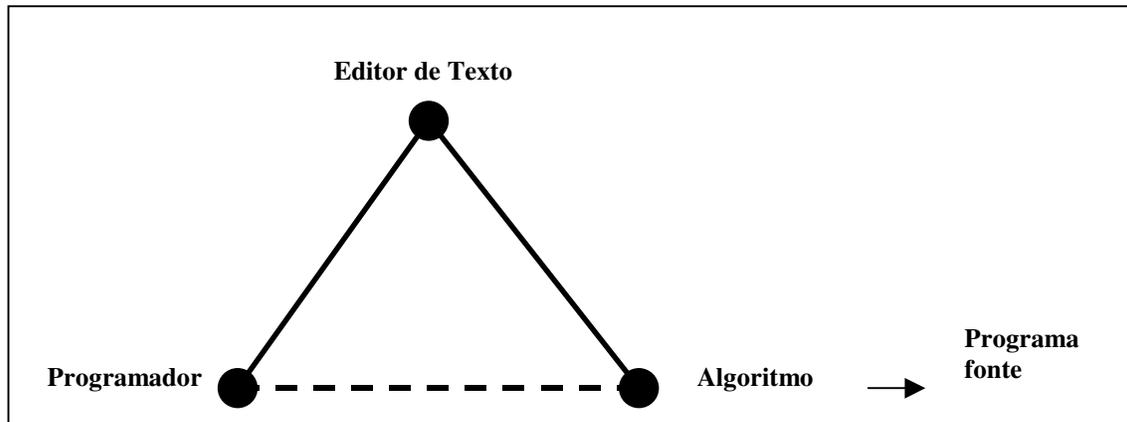


Figura 4.2 - Estrutura da atividade "codificar um programa"

É importante observar que embora neste exemplo estejamos destacando como ferramenta de mediação um editor de texto, outras ferramentas de mediação também são utilizadas no contexto desta atividade, por exemplo, a lógica de programação e o conjunto de instruções (códigos) de uma linguagem de programação que o programador utiliza para transformar o algoritmo em código compilável.

Embora a representação do relacionamento mediado entre sujeito e objeto no nível individual seja útil, essa estrutura é simples demais para representar as considerações de relações sistêmicas existentes entre o sujeito e o seu ambiente, uma vez que essas relações são encontradas em muitas atividades [Mar99a]. Assim, um novo elemento deve ser adicionado: a *comunidade*. Uma comunidade é formada por todos os sujeitos que compartilham um mesmo objeto. Quando se introduz o conceito de *comunidade* novas formas de mediação aparecem (além daquela possibilitada pelas ferramentas), estas novas formas de mediação são denominadas *regras* e *divisão do trabalho* (vide Figura 4.3).

As *regras*, enquanto uma forma de mediação entre o sujeito e a comunidade, são normas implícitas ou explícitas estabelecidas por convenções e relações sociais dentro da comunidade. A *divisão do trabalho*, enquanto forma de mediação entre a comunidade e o objeto, se refere à forma de organização de uma comunidade, relacionada ao processo de transformação de um objeto em um resultado.

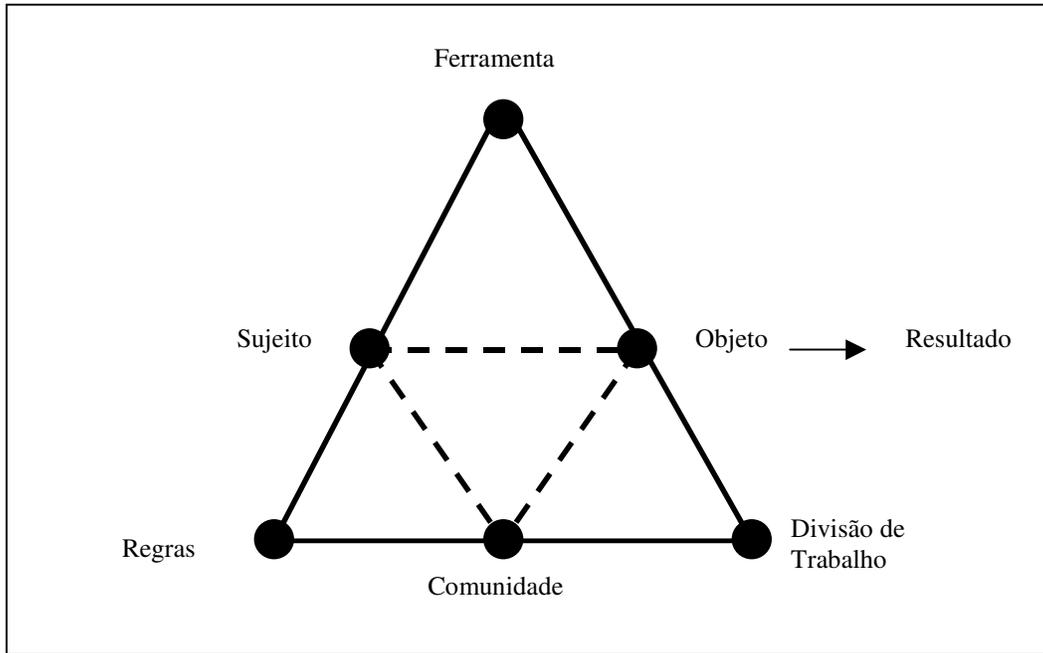


Figura 4.3 - Modelo sistêmico da atividade (também chamado de diagrama de Engeström)
[Eng87] [Col93]

Todas as formas de mediação (ferramentas, regras e divisão do trabalho) possuem um desenvolvimento histórico próprio, com características particulares relacionadas ao contexto em que foram desenvolvidas.

Segundo a Teoria da Atividade, atividades não são estáticas, pois evoluem ao longo do tempo. Cada atividade tem sua própria história, embutindo “fases” passadas. Uma análise histórica do desenvolvimento é frequentemente necessária para o entendimento da situação corrente.

4.3 Níveis da Atividade

Uma atividade é decomposta em *ações*, e cada ação pode ser decomposta em *operações* (vide Figuras 4.4 e 4.5). Atividades são formações de longo prazo, seus objetos são transformados em resultados não apenas uma vez, mas sim de forma iterativa, que tipicamente consiste em várias

fases ou etapas. Assim, uma atividade, concebida como tal num dado momento, passou por um processo de evolução, onde ações e operações podem ter sido criadas, eliminadas e transformadas para que a atividade chegasse ao seu “formato” atual [Kuu96].

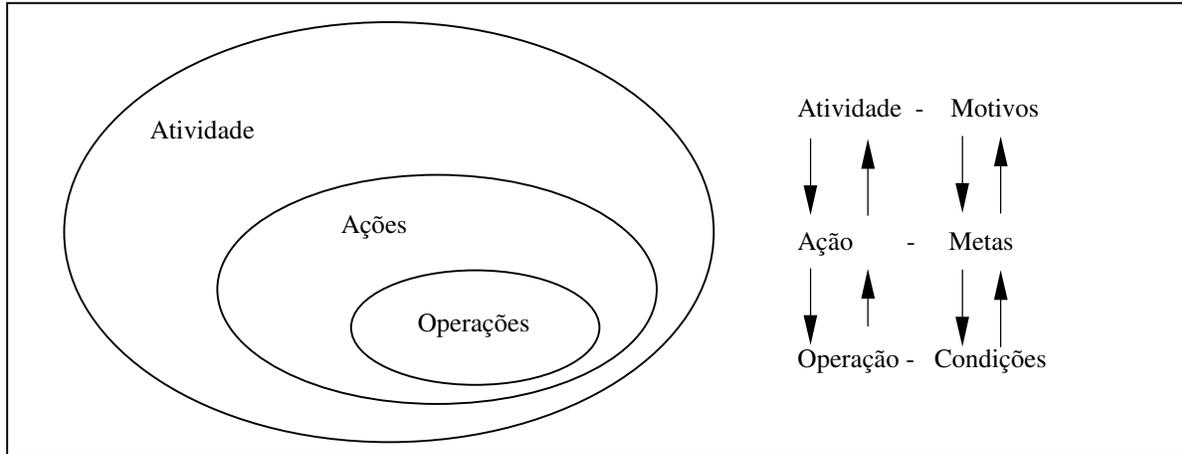


Figura 4.4 - Níveis hierárquicos de uma atividade [Mar99a]

Enquanto uma atividade é orientada a motivos, as ações são orientadas a metas, e as operações orientadas a condições. Uma atividade é realizada por ações cooperativas ou individuais, podendo se estabelecer cadeias ou redes de ações que estão relacionadas umas com as outras pelo motivo da atividade. As ações que constituem uma atividade são energizadas por seu motivo [Leo81]. O motivo é a razão que orienta a atividade, expresso em termos de desejos e necessidades humanas. As metas são objetivos de curto prazo a serem atingidos pelas ações da atividade.

As ações concretizam uma atividade, e possuem metas bem definidas para tal. Assim, existe uma relação importante entre o motivo da atividade e as metas das ações que compõem a atividade. O motivo da atividade pode ser visto como o ponto para o qual as metas das ações devem convergir (o “norte” das ações).

Uma característica importante de uma ação é que ela é planejada antes de sua execução efetiva, diferentemente de uma operação, que é executada de forma automática, sem um planejamento prévio, bastando apenas uma análise das condições atuais para a sua execução. O planejamento de uma ação é feito de forma consciente, usando-se algum modelo mental para isso, quanto

melhor o modelo mais sucesso terá a ação. Este planejamento para a execução de uma ação é chamado de orientação.

Quando uma ação é realizada várias vezes e alcança um nível de maturidade suficiente para que ela possa ser executada automaticamente, ou seja, sem um planejamento prévio, então ela passa para o nível de operação. Dessa forma, uma operação é uma ação que se tornou comum no contexto de uma atividade, pois é executada com um alto grau de repetição dentro deste contexto.

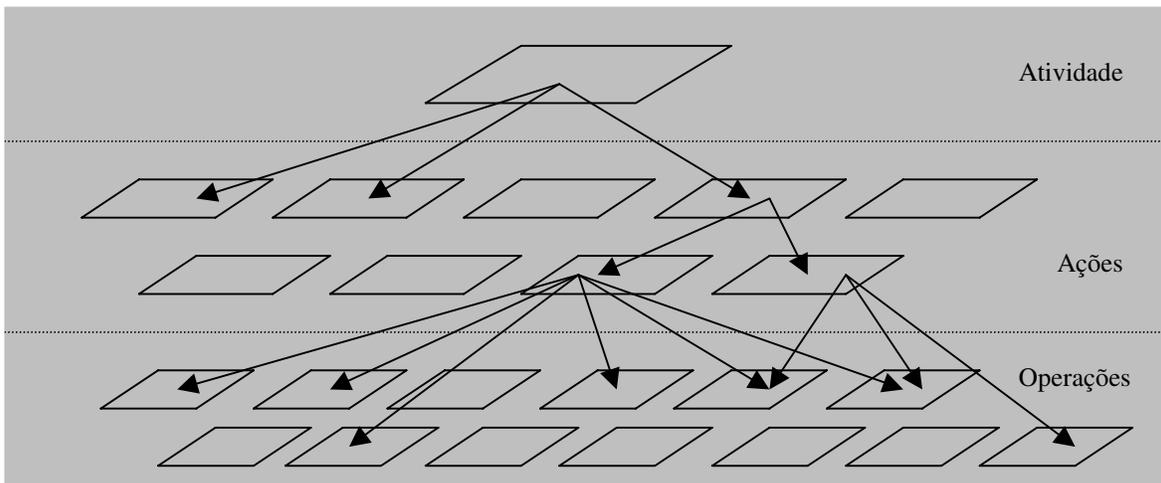


Figura 4.5 – Relacionamento entre os níveis da atividade [Bur99]

Na *Tabela 4.1* damos um exemplo de decomposição de uma atividade. A atividade "Faturar uma Venda" foi decomposta nas ações "Emitir Nota Fiscal" e "Emitir Duplicatas". A atividade "Faturar uma Venda" é motivada pela necessidade de documentar a venda para efeitos fiscais e fazer o parcelamento da mesma. A ação "Emitir Nota Fiscal" tem como meta gerar uma nota fiscal nos padrões da legislação fiscal, para efeitos de arquivo e documentação, e a ação "Emitir Duplicatas" tem como meta gerar a documentação necessária para o pagamento parcelado da venda realizada.

Tabela 4.1- Decomposição da atividade "Faturar uma Venda"

Atividade	Ações	Operações
Faturar uma Venda	Emitir Nota Fiscal	Preencher campos da NF
		Calcular impostos
		Imprimir NF
	Emitir duplicatas	Gerar datas de vencimento
		Imprimir duplicatas

Conclusão do Capítulo

A Teoria da Atividade traz os conceitos sobre os quais a metodologia de elicitação de requisitos proposta neste trabalho irá se sustentar. Conforme visto neste capítulo, o conceito de atividade é o ponto central da Teoria, que declara que a atividade estabelece um contexto mínimo para o entendimento das ações realizadas por pessoas durante a transformação do objeto da atividade, que pode ser um objeto físico ou abstrato.

A atividade é vista como uma estrutura hierárquica, dividida em ações e operações. Atividades são orientadas a motivos, enquanto ações são orientadas a metas (relacionadas entre si por meio dos motivos da atividade) e as operações são orientadas a condições de realização. Além dos elementos que compõem a estrutura hierárquica da atividade, também existem os elementos que formam o contexto da atividade, quais sejam: sujeito, ferramentas de mediação, objeto de transformação, regras, comunidade, divisão do trabalho e resultados alcançados.

As idéias e conceitos apresentados neste capítulo sofrerão um processo de refinamento no próximo capítulo, que apresenta definições objetivas sobre cada conceito ligado a atividade. Essas definições servirão de guias iniciais para a utilização da metodologia de elicitação de requisitos proposta .

Parte II

Capítulo 5

Uma Metodologia de
Elicitação de Requisitos de
Software Baseada na Teoria
da Atividade

5. Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade

5.1 Visão Geral da Metodologia

Nesta seção apresentamos uma visão geral da metodologia de elicitação de requisitos proposta, destacando a seqüência e iteração entre as etapas. A metodologia proposta se divide em três etapas principais:

1. Divisão do problema em atividades (unidades de elicitação de requisitos)
2. Delineamento do contexto da atividade (para cada atividade)
3. Descrição da estrutura hierárquica das atividades (para cada atividade)

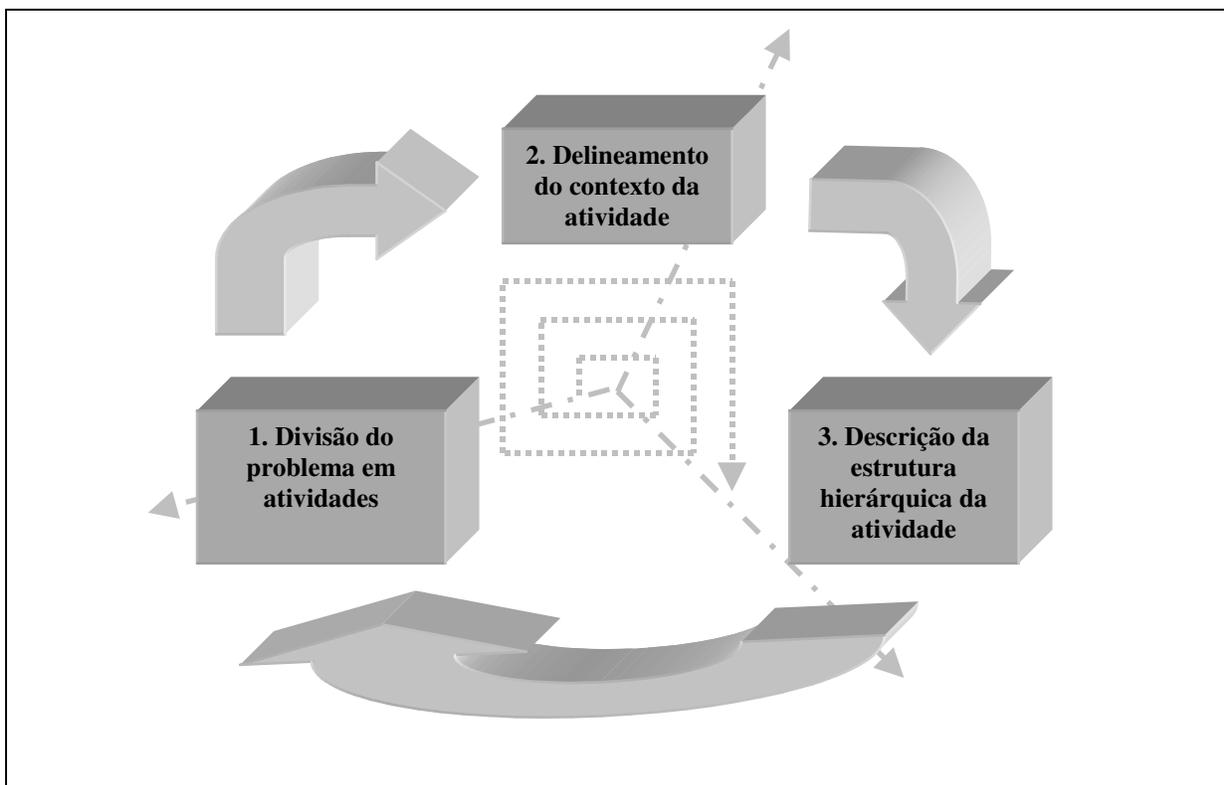


Figura 5.1 – Etapas da metodologia de elicitação de requisitos proposta

A *Figura 5.1* apresenta as três principais etapas da metodologia de elicitação de requisitos proposta neste trabalho, enfatizando que o desdobramento destas etapas deve ocorrer de forma iterativa e incremental, ou seja, na medida em que as atividades vão sendo identificadas e descritas os requisitos vão se tornando mais claros e a elicitação de requisitos vai se desenvolvendo gradativamente.

Cada etapa exibida na *Figura 5.1* divide-se em etapas menores, que possuem procedimentos próprios. Nas *Figuras 5.2 à 5.4* apresentamos uma decomposição de cada uma das etapas apresentadas na *Figura 5.1*.

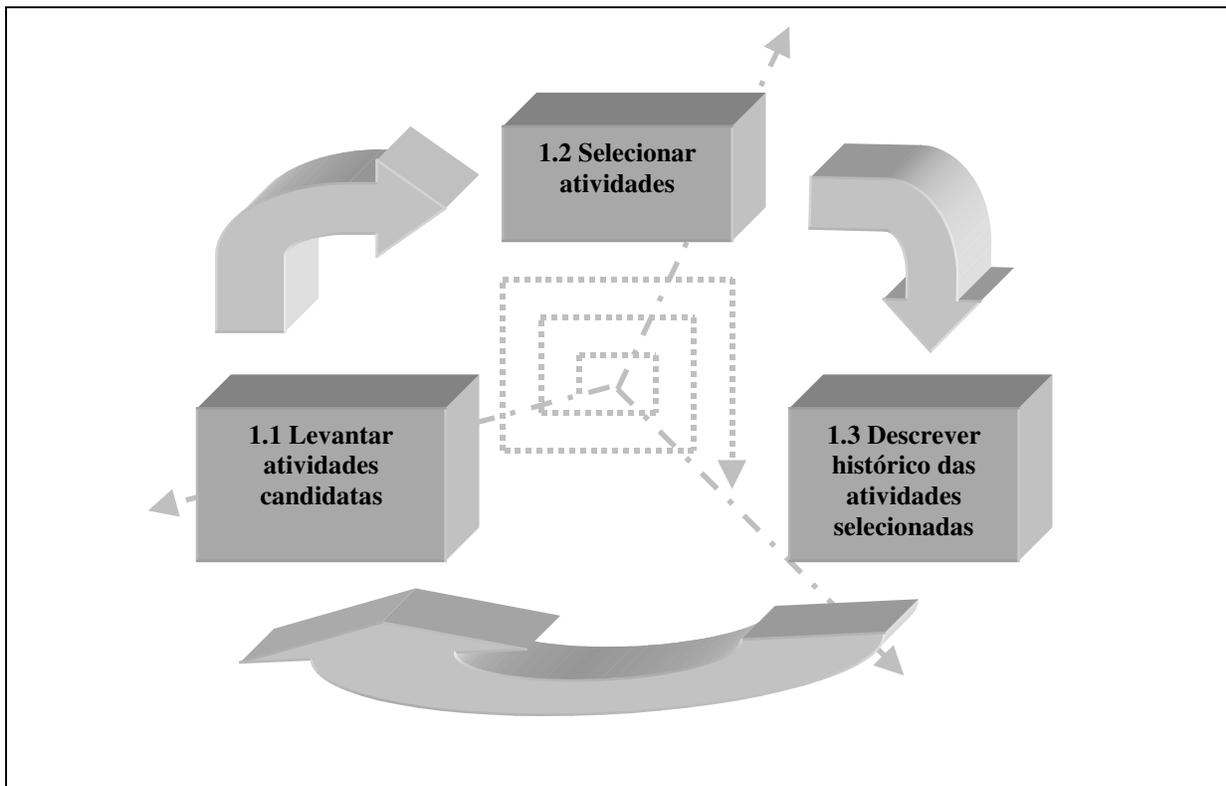


Figura 5.2 – Decomposição da etapa “Divisão do problema em atividades”

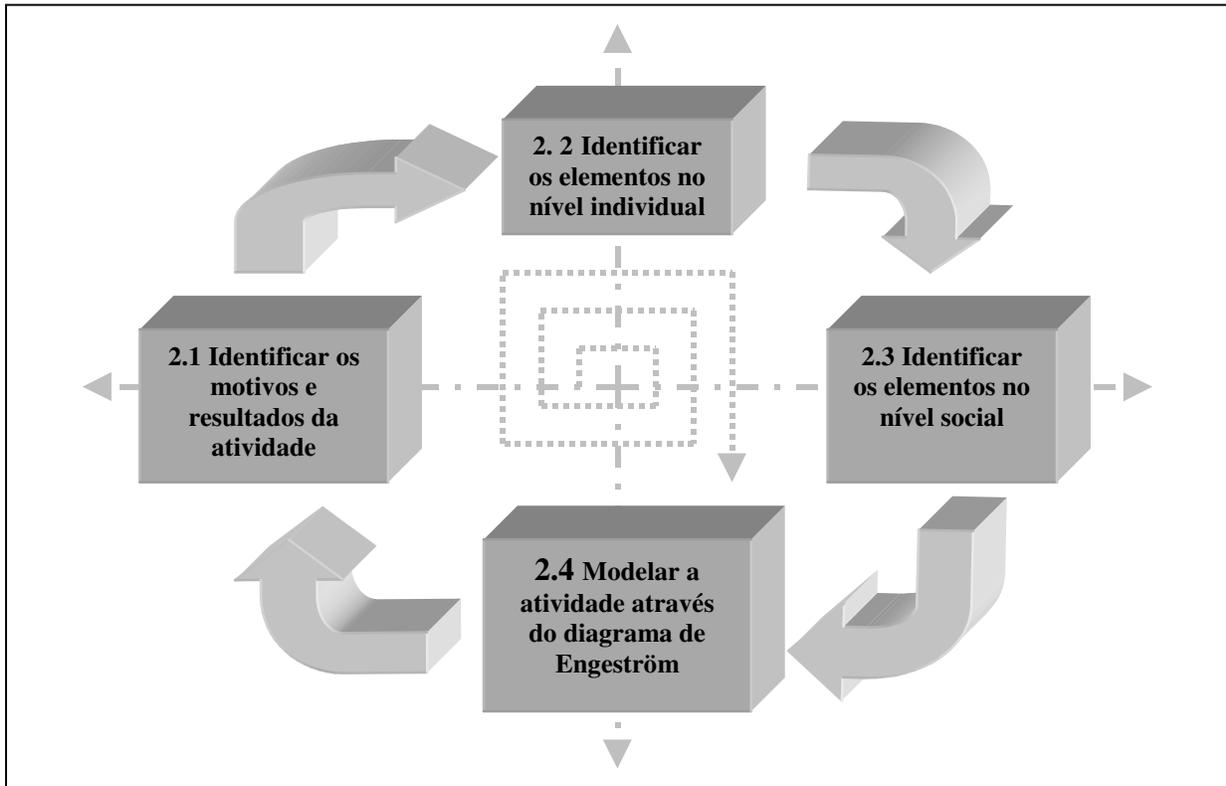


Figura 5.3 – Decomposição da etapa “Delineamento do contexto da atividade”

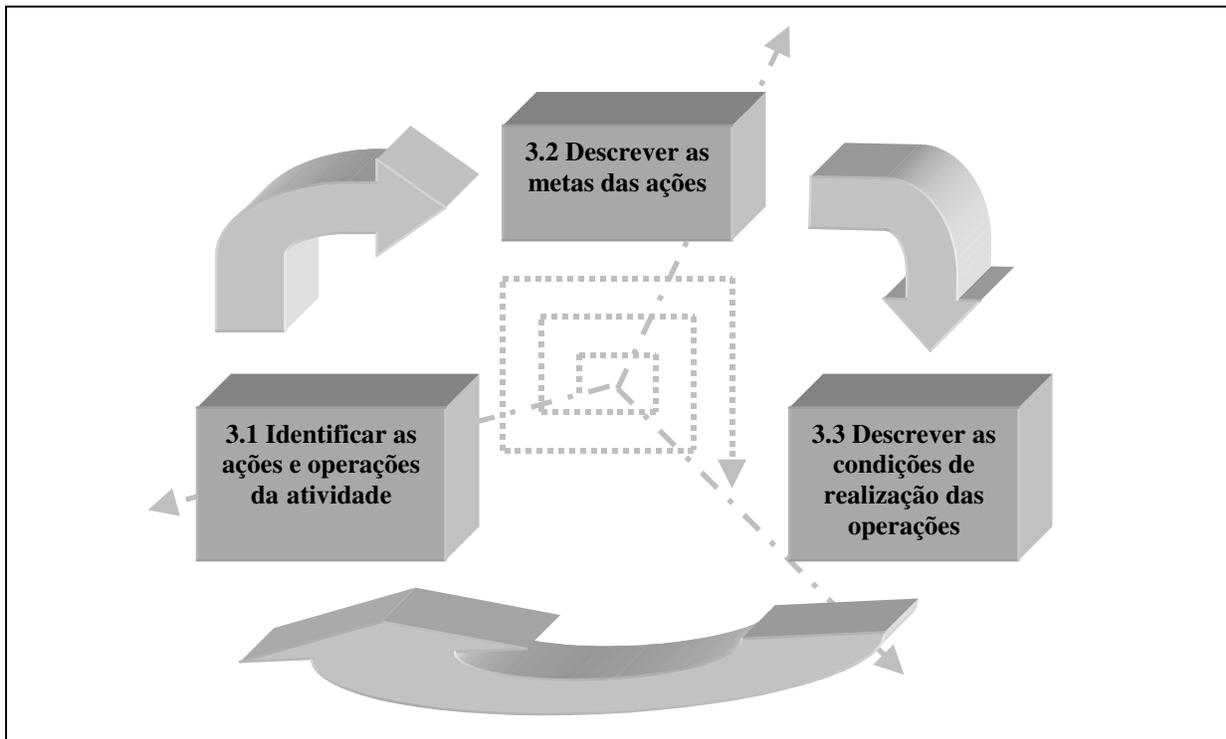


Figura 5.4 – Decomposição da etapa “Descrição da estrutura hierárquica da atividade”

Cada uma das etapas apresentadas aqui estão descritas com detalhes na Seção 5.3. A seguir apresentamos a definição de todos os conceitos relativos à atividade que são utilizados na metodologia proposta.

5.2 Definição dos Principais Conceitos Relativos à Atividade

As Seções 4.1, 4.2 e 4.3 ofereceram uma visão geral da Teoria da Atividade, apresentando uma série de preceitos importantes para a definição da metodologia de elicitação de requisitos que apresentamos neste capítulo. Apresentamos a seguir uma definição objetiva para cada conceito vinculado à atividade, que foram comentados no Capítulo 4, pois os mesmos são interpretados na metodologia proposta conforme aparecem nas definições. Com vistas à organização da metodologia, classificamos os conceitos da Teoria da Atividade em dois grupos:

- Conceitos Relativos à Estrutura Hierárquica da Atividade;
- Conceitos Relativos aos Elementos que Formam o Contexto da Atividade.

Para cada conceito será apresentada uma caracterização geral, enfatizando o seu objetivo e importância no contexto de uma atividade, seguido de uma definição¹⁵.

5.2.1 Conceitos Relativos à Estrutura Hierárquica da Atividade

Os conceitos apresentados neste grupo estão ligados à estrutura hierárquica funcional da atividade, importantes para a compreensão da estrutura de níveis existentes em uma atividade. Os elementos que formam este grupo são:

- Atividade
- Motivo

¹⁵ É importante destacar que a Teoria da Atividade não apresenta definições sobre os elementos da atividade conforme estão apresentados nesta seção. Foi necessário um esforço de interpretação para a elaboração destas definições, de tal forma que estes elementos pudessem ser aplicados na metodologia de elicitação de requisitos proposta.

- Ação
- Meta
- Operação
- Condições

Atividade

Uma atividade oferece um contexto mínimo para a elicitação de requisitos, a partir do entendimento de um conjunto de ações inter-relacionadas. As ações abrangidas na atividade agem sobre um determinado objeto para se atingir algum resultado [Kuu96].

Atividades não são entidades rígidas ou estáticas, elas se transformam ao longo do tempo. A dimensão dinâmica de uma atividade implica que para uma efetiva compreensão da situação atual é necessária uma análise histórica do desenvolvimento da atividade. Uma atividade pode ao longo de seu desenvolvimento embutir novas ações e operações, assim como também poderá vir a se transformar numa ação no contexto de uma outra atividade.

Definição 1 [Atividade]: atividade é uma unidade de elicitação de requisitos que oferece um contexto mínimo para o entendimento de um conjunto de ações cooperantes que agem sobre um ou mais objetos, transformando-os num resultado.

Motivo

Toda atividade possui um ou mais motivos. O motivo orienta a realização da atividade [Bur99]. A Teoria da Atividade trata de questões relativas ao desenvolvimento das atividades humanas, assim, é importante destacar que os motivos que levam à realização das atividades estão intimamente relacionados aos desejos ou necessidades dos seres humanos envolvidos.

Definição 2 {Motivo}: motivo é a razão que orienta a atividade, expresso em termos de desejos ou necessidades humanas.

Ação

Uma atividade deve transformar um objeto num resultado através de passos conscientes. No entanto, esta transformação não ocorre de forma direta entre atividade e objeto. O objeto é transformado através de ações que incidirão sobre ele no contexto da atividade. As ações cooperam entre si durante o processo de transformação do objeto. Uma ação é uma forma consciente de agir, implicando na escolha de um modelo mental adequado para enfrentar determinada situação [Kuu96].

Ações são energizadas pelo motivo da atividade, mas elas possuem metas próprias, bem definidas e imediatas, que contribuem diretamente para a transformação do objeto em resultado. Uma ação pode ser dividida em operações. As ações são realizadas pelos sujeitos envolvidos na atividade.

Definição 3 {Ação}: uma ação é um “passo” consciente realizado com o intenção de se atingir uma meta bem definida no contexto da atividade.

Meta

Toda ação possui pelo menos uma meta. As metas devem ser alcançadas pelas ações para que o resultado da atividade possa ser atingido. As metas são definidas na medida em que vão sendo testadas e descartadas pelas ações [Leo81]. A experiência individual do sujeito, acumulada no desenvolvimento da atividade, é que dará condições para a escolha adequada das metas a serem atingidas [Bur99]. Podem existir várias metas candidatas para uma mesma ação.

Definição 4 {Meta}: uma meta é um objetivo imediato a ser atingido por uma ação.

Operação

Uma operação é uma ação que se tornou extremamente rotineira dentro do contexto de uma atividade, sendo realizada de forma automática¹⁶ sob determinadas condições. Segundo Kuutti [Kuu96] uma ação torna-se automática (não consciente) quando ela é realizada inúmeras vezes pelo sujeito, onde este utiliza-se de um mesmo modelo mental considerado adequado o suficiente para a realização da ação.

Definição 5 {Operação}: uma operação é uma ação que se tornou rotineira no contexto da atividade, de tal forma que ela é realizada de forma automática pelo sujeito.

Condições

Para que uma operação possa ser realizada a contento, ela deve estar sob condições reconhecidas pelo sujeito. Quando as condições de execução da operação mudam, então a operação volta para o nível de ação [Kuu96], pois ela não poderá ser executada de forma automática pelo sujeito.

Definição 6 {Condições}: um conjunto de variáveis que possuindo um determinado estado habilita a execução de uma operação.

¹⁶ Entendemos por automática uma ação que de tão rotineira passa a ser realizada de forma não consciente.

5.2.2 Conceitos Relativos aos Elementos que Formam o Contexto da Atividade

Os conceitos apresentados neste grupo versam sobre os componentes da atividade, que auxiliam no entendimento de seu contexto, tanto no nível individual como social. Os elementos que formam este grupo são:

- Sujeito
- Objeto
- Ferramenta
- Comunidade
- Regras
- Divisão de Trabalho
- Resultado

Sujeito

O sujeito é um agente da atividade, é quem executa ações e operações sobre o objeto da atividade. Na medida em que o sujeito transforma o objeto da atividade em um resultado, ele também é transformado pelas características do objeto [Nar96], principalmente pelas características culturais carregadas pelo objeto. Uma atividade pode ter vários sujeitos engajados.

Definição 7 {Sujeito}: é um agente que transforma o objeto da atividade por meio da execução de ações e operações.

Objeto

Um objeto pode ser qualquer coisa que possa ser acessada, processada ou transformada no decorrer da atividade [Bur99].

Segundo Kuuti [Kuu96], “*um objeto pode ser uma coisa material, mas também pode ser algo menos tangível (tal como um plano) ou totalmente intangível (tal como uma idéia), desde que ele possa ser compartilhado para manipulação e transformação pelos participantes da atividade*”.

Definição 8 {Objeto}: algo material ou abstrato, que pode ser compartilhado pelos participantes da atividade.

Ferramenta

Uma ferramenta é um artefato de mediação entre o sujeito e o objeto da atividade. Vygotsky [Vyg81] oferece uma classificação de dois tipos relacionados de ferramentas: as ferramentas técnicas e as ferramentas psicológicas.

As ferramentas técnicas servem para transformar os objetos [Bur99], por exemplo: machado, caneta, computador etc. As ferramentas psicológicas servem para visualizar, comunicar e representar conceitos [Bur99], por exemplo: escrita, diagramas, sinais convencionais, símbolos matemáticos, estratégias para resolver problemas etc.

Definição 9 {Ferramenta Técnica}: é um artefato físico de mediação utilizado pelo sujeito na transformação de um objeto.

Definição 10 {Ferramenta Psicológica}: é um artefato abstrato de mediação utilizado pelo sujeito para visualizar, comunicar ou representar conceitos.

Comunidade

O conceito de comunidade é fundamental para que possamos compreender uma atividade em toda sua amplitude, pois os seres humanos (sujeitos) sempre realizam suas atividades dentro de uma ou mais comunidades [Col93].

No contexto da atividade uma comunidade é formada pelos sujeitos que de alguma forma influenciam o objeto (ou objetos) da atividade. Com a inserção da comunidade como um componente da atividade, Engeström amplia a noção de mediação existente nas atividades humanas proposta por Vygotsky, destacando dois novos componentes de mediação: regras e divisão do trabalho [Col93].

Definição 11 {Comunidade}: é um conjunto formado por sujeitos que influenciam na transformação do objeto da atividade.

Regras

As regras constituem o instrumento de mediação entre comunidade e sujeito. Segundo Cole e Engeström [Col93] as regras são “*as normas e as sanções que especificam e regulam os procedimentos corretos esperados e as interações aceitáveis entre os participantes*”.

Definição 12 {Regras}: conjunto de normas e procedimentos dentro de uma comunidade, que um sujeito deve atender durante a realização de uma atividade.

Divisão do Trabalho

A divisão do trabalho constitui o instrumento de mediação entre a comunidade e o objeto da atividade. Segundo Cole e Engeström [Col93] uma comunidade implica em uma divisão de trabalho, que é uma “*negociação contínua de distribuição de tarefas, poderes e responsabilidades entre os participantes da atividade*”.

Também podemos entender a divisão do trabalho como sendo os papéis que os participantes da atividade assumem no contexto da mesma, de tal forma que se consiga algum grau de organização das influências dos participantes sobre o objeto da atividade.

Definição 13 {Divisão do Trabalho}: é o conjunto de papéis e responsabilidades que os sujeitos assumem dentro de uma comunidade durante a realização de uma atividade.

Resultado

O resultado é o produto final da atividade que, espera-se, esteja em conformidade com os motivos que levaram à realização da atividade. Para se chegar ao resultado, um processo de transformação ocorreu, envolvendo todos os conceitos apresentados neste capítulo, que em síntese formam uma atividade.

Definição 14 {Resultado}: é o produto final do processo de transformação inerente à atividade.

5.2.3 Relacionamento entre os Conceitos Envolvidos na Atividade

Conforme apresentado nas Seções 5.2.1 e 5.2.2, uma atividade embute vários elementos conceituais, que precisam ser entendidos para termos uma compreensão completa da atividade. No entanto, esses elementos não podem ser vistos de maneira isolada. A seguir apresentamos o relacionamento existente entre os elementos da atividade. Uma visão pictórica destes relacionamentos é oferecida através do diagrama de classe, conforme padronização da UML [Boo99].

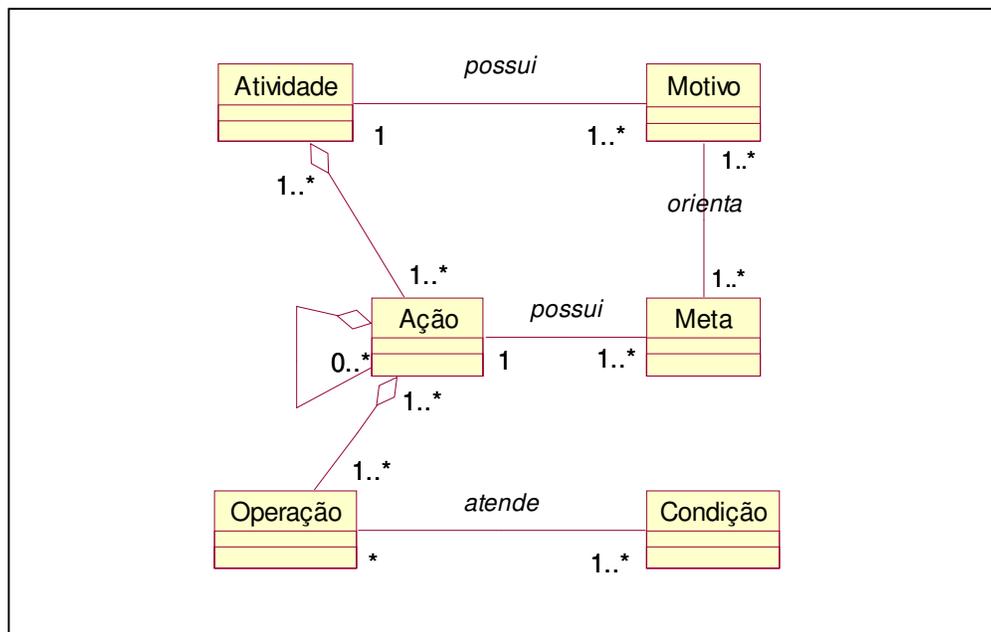


Figura 5.5 – Relacionamento entre os elementos que formam a estrutura hierárquica da atividade

A *Figura 5.5* apresenta o relacionamento existente entre os elementos que formam a estrutura hierárquica da atividade, onde destacamos que a atividade é uma agregação de ações, e que uma ação é uma agregação de operações. Uma ação pode fazer “parte” de várias atividades. Uma ação também pode agregar outras ações. Toda atividade possui pelo menos um motivo, que no contexto do sistema analisado está associado a apenas uma atividade. Cada ação possui pelo menos uma meta, que no contexto do sistema analisado também está associado a apenas uma ação. Os motivos da atividade orientam as metas das ações da atividade. Uma operação pode atender de uma a muitas condições, e uma condição pode definir o estado de execução de várias operações.

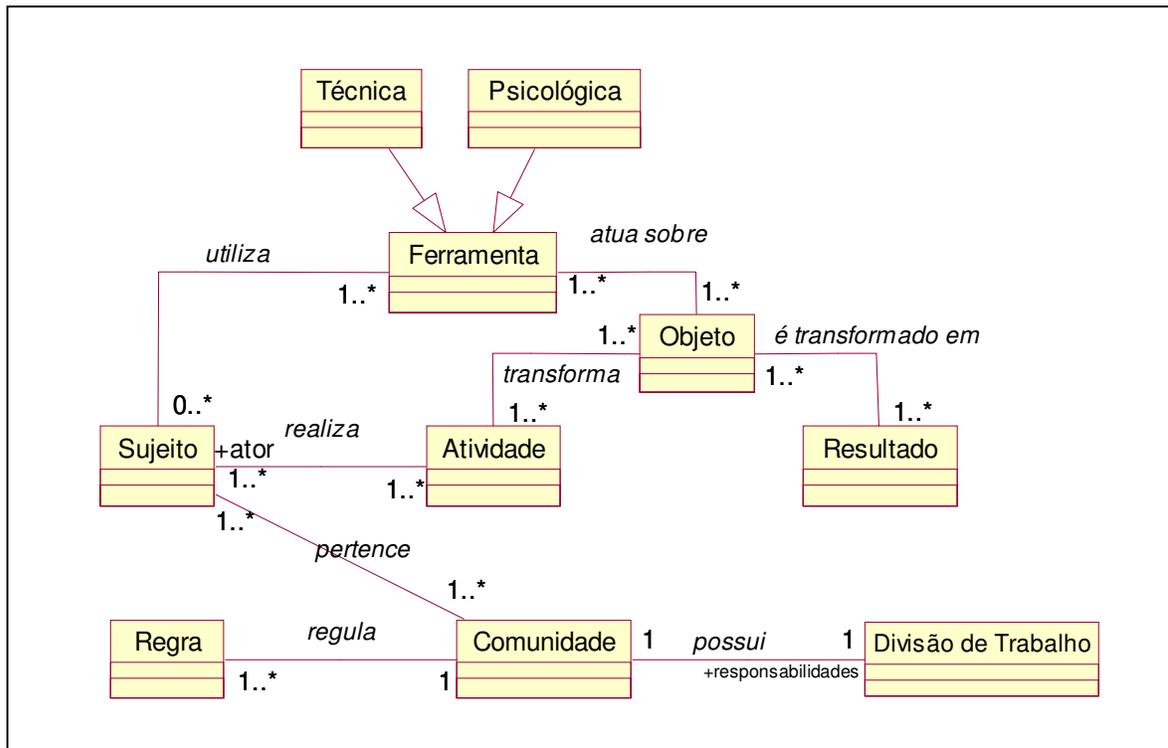


Figura 5.6 – Relacionamento entre os elementos que formam o contexto da atividade

A *Figura 5.6* apresenta o relacionamento entre os elementos que formam o contexto da atividade. Percebemos que a maioria dos elementos se relaciona com multiplicidade de muitos para muitos, o que denota o grau de complexidade nas relações existentes entre estes elementos.

Uma atividade pode ser realizada por vários sujeitos, onde cada sujeito é visto como um ator da atividade. Um mesmo sujeito pode participar de várias atividades. Um sujeito durante a realização da atividade pode utilizar várias ferramentas de mediação (especializadas em ferramentas técnicas e psicológicas). Uma ferramenta pode ser utilizada por vários sujeitos.

O sujeito utiliza as ferramentas para transformar os objetos da atividade, assim, uma ferramenta pode atuar sobre vários objetos, que por sua vez podem receber a atuação de várias ferramentas. Um objeto é transformado em um ou mais resultados, que podem ser resultantes de mais de um objeto transformado.

Um sujeito pode pertencer a mais de uma comunidade, que é formada por vários sujeitos. A comunidade é regulada por várias regras e possui uma divisão do trabalho (a divisão do trabalho assume o papel de um conjunto de responsabilidades dentro da comunidade).

Outros relacionamentos podem ser identificados entre os elementos da atividade, como por exemplo ferramenta-atividade, comunidade-atividade, que consideramos implícitos no modelo apresentado e podem ser lidos de forma indireta por meio de outros relacionamentos (como atividade-sujeito e sujeito-ferramenta).

5.3 Descrição das Etapas

Essa seção apresenta um detalhamento das etapas da metodologia de elicitação de requisitos proposta, destacando os procedimentos que devem ser adotados em cada uma das etapas, as definições e princípios da Teoria da Atividade que são empregados em cada procedimento, e como se dá o encadeamento entre as etapas.

5.3.1 Dividir o Problema em Atividades Realizadas no Contexto do Sistema

A metodologia de elicitação de requisitos proposta neste trabalho está centrada no conceito de atividade, conforme preconizado na Teoria da Atividade. Partimos da premissa de que a atividade deve ser considerada como uma unidade de elicitação de requisitos, uma vez que as pessoas envolvidas num sistema organizam a sua forma de trabalhar em atividades.

De acordo com o *princípio (1)* da Teoria da Atividade “a consciência humana é abastecida pelas atividades realizadas pelas pessoas” e “a formação dos processos mentais tem sua gênese na realização das atividades”. A partir disto, organizar o problema em atividades se torna um passo primordial nesta abordagem de elicitação de requisitos.

No entanto, identificar atividades relevantes no contexto do sistema não é uma tarefa trivial. Este procedimento exige uma aferição da granularidade das diversas tarefas realizadas pelos sujeitos envolvidos no sistema. Baseando-se nos preceitos da Teoria da Atividade, as tarefas podem ter *status* de atividade, ação ou operação, onde a atividade é a tarefa de maior granularidade, seguida por ação e operação.

5.3.1.1 Levantar Atividades Candidatas

Qualquer tarefa realizada por agentes envolvidos no sistema pode inicialmente ser indicada como uma atividade em potencial. Dessa forma, não há num primeiro momento um “filtro” que garanta a seleção imediata de atividades.

No entanto, fazendo uso da *definição (1)*, podemos realizar um primeiro levantamento de possíveis atividades que são realizadas pelos agentes envolvidos (possivelmente usuários do futuro sistema de software), mas ainda sem a preocupação de uma classificação precisa das tarefas como atividade, ação ou operação.

5.3.1.2 Selecionar Atividades

Feito o levantamento inicial das atividades, devemos buscar uma classificação mais precisa sobre as tarefas apontadas inicialmente como atividades. Para este procedimento, devemos fazer uso das *definições (1) (3) e (5)*, pois as mesmas auxiliam na avaliação da granularidade das tarefas em atividades, ações ou operações. A classificação das tarefas como atividades, ações ou operações se baseia no *princípio (3)* da Teoria da Atividade.

Ao término desta etapa, já temos uma organização mínima para a elicitação dos requisitos, onde cada atividade passa a ser vista como uma unidade de elicitação. Devemos, portanto, partir para uma investigação das origens das atividades selecionadas.

5.3.1.3 Descrever Histórico das Atividades Selecionadas

De acordo com o *princípio (6)* da Teoria da Atividade “entender um fenômeno significa conhecer como ele se desenvolveu até sua forma atual”. O fenômeno de nosso interesse é a atividade que o sujeito realiza e, portanto, devemos partir para uma investigação histórica da origem e desenvolvimento da atividade.

Uma descrição do histórico de cada atividade deve ser apresentada nesta etapa, procurando destacar os aspectos originais que levaram ao surgimento da atividade. Entendemos que este procedimento ajuda a contextualizar melhor as atividades no dia a dia de seus atores.

5.3.2 Delinear o Contexto da Atividade

Uma vez definidas as atividades, podemos partir para o delineamento do contexto de cada uma delas. Conforme apresentado na Teoria da Atividade, os seguintes elementos estão presentes em qualquer atividade humana realizada: motivos, resultados, sujeitos, ferramentas de mediação (técnicas e psicológicas), objetos, regras, comunidade e divisão do trabalho. Estes elementos formam o que chamamos de contexto da atividade.

A necessidade de delinear o contexto da atividade através dos elementos citados acima se baseia nos *princípios (2) e (5)* da Teoria da Atividade.

5.3.2.1 Identificar os Motivos e Resultados da Atividade

Inicialmente sugerimos a identificação dos elementos motivos e resultados da atividade. Conforme apresentado na *definição (2)* um motivo é expresso através de desejos e necessidades humanas, os quais consideramos ser o “ponto de partida” da atividade. Por sua vez o resultado da atividade, conforme apresentado na *definição (14)*, é o produto final do processo de transformação embutido na atividade, o qual consideramos como o “ponto de chegada” da atividade.

Através da identificação destes dois elementos, temos uma primeira medida do “tamanho” da atividade. Fazendo uma analogia da atividade com um percurso, a identificação dos motivos e resultados da atividade seriam como as estacas de início e fim do percurso. As curvas e retas, declives e aclives do percurso serão determinados pelos elementos no nível individual e social da atividade.

5.3.2.2 Identificar os Elementos no Nível Individual

Os elementos que formam a atividade no nível individual são sujeito, ferramenta de mediação (técnica e psicológica) e objeto de transformação da atividade (podendo existir uma variedade de cada um destes elementos dentro de uma mesma atividade).

Partimos inicialmente para a identificação do sujeito (ou sujeitos) da atividade. Conforme apresentado na *definição (7)* o sujeito é o responsável pela transformação do objeto da atividade em um resultado, através da execução de ações e operações. O sujeito é o principal agente da atividade, sendo uma fonte importante de informações que auxiliará na identificação de outros elementos da atividade, como por exemplo, objeto da atividade, regras e divisão do trabalho.

O sujeito atua diretamente sobre o objeto de transformação da atividade, sendo assim, após a identificação do sujeito, partimos para a identificação do objeto da atividade. Pela *definição (8)* temos que o objeto da atividade pode ser algo material ou abstrato, e que é compartilhado pelos participantes da atividade (sujeitos e comunidade). A identificação do objeto da atividade denota uma compreensão importante sobre a atividade pois, sendo compartilhado pelos participantes da atividade, ele se torna um elemento central que servirá como um ponto de referência para a identificação dos demais elementos da atividade.

O sujeito atua sobre o objeto da atividade através de ferramentas de mediação. Várias ferramentas de mediação podem ser utilizadas durante uma atividade, na medida em que as ações e operações vão sendo executadas. Assim tanto ferramentas técnicas como psicológicas podem ser utilizadas numa mesma atividade. Conforme apresentado na *definição (9)* uma ferramenta técnica é um artefato físico de mediação utilizado pelo sujeito na transformação do objeto. Necessariamente algum tipo de transformação ocorrerá no objeto através do uso de uma ferramenta técnica. Pela *definição (10)* uma ferramenta psicológica é um artefato abstrato utilizado pelo sujeito para visualizar, comunicar ou representar conceitos. As ferramentas identificadas podem ser utilizadas simultaneamente ou alternadamente durante a realização da atividade.

Ao analisarmos a atividade no nível individual estamos realizando uma abstração da atividade que nos auxilia na sua compreensão inicial, deixando de lado os aspectos da mediação social, que embora sejam inerentes a qualquer atividade humana, num primeiro momento poderiam desviar nossa atenção na identificação dos elementos básico da atividade (sujeito, objeto e ferramenta de mediação).

5.3.2.3 Identificar os Elementos no Nível Social

Após termos identificados os elementos básicos da atividade, e portanto ganharmos uma compreensão sobre a mesma no nível individual, partimos para os elementos do nível social da atividade.

No nível social o elemento chave a ser identificado é a comunidade (ou comunidades) em que o sujeito da atividade está inserido. A comunidade de interesse é aquela que pode influenciar de alguma maneira a realização da atividade. Um mesmo sujeito pode ser membro de mais de uma comunidade.

Para a identificação da comunidade nos baseamos na *definição (11)*, que declara que uma comunidade é formada pelos sujeitos que de alguma forma influenciam o objeto (ou objetos) da atividade. Conforme apresentado na Seção 4.2, o relacionamento entre sujeito-comunidade é mediado por regras, e o relacionamento entre comunidade-objeto é mediado pela divisão do trabalho. Dessa forma, baseados nas *definições (12) e (13)* partimos para a descrição das regras que regulam a comunidade em questão e para a descrição da divisão do trabalho existentes entre os seus componentes.

A definição das regras e divisão do trabalho contribuirão posteriormente para a identificação das ações e operações, pois estas são executadas pelo sujeito da atividade, que tem comportamento regulado pelas regras e atribuições dentro da comunidade (ou comunidades) em que atua.

5.3.2.4 Modelar a Atividade através do Diagrama de Engeström

Tendo todos os elementos da atividade identificados, uma representação pictórica destes elementos se revela útil para efeitos de visualização do contexto formado pela atividade. Assim, adotamos o diagrama de Engeström para a modelagem dos principais elementos da atividade.

Conforme apresentado na Seção 4.2, os elementos da atividade que são visualizados através do diagrama de Engeström são: sujeito, ferramenta de mediação, objeto, regras, comunidade, divisão do trabalho e resultado da atividade.

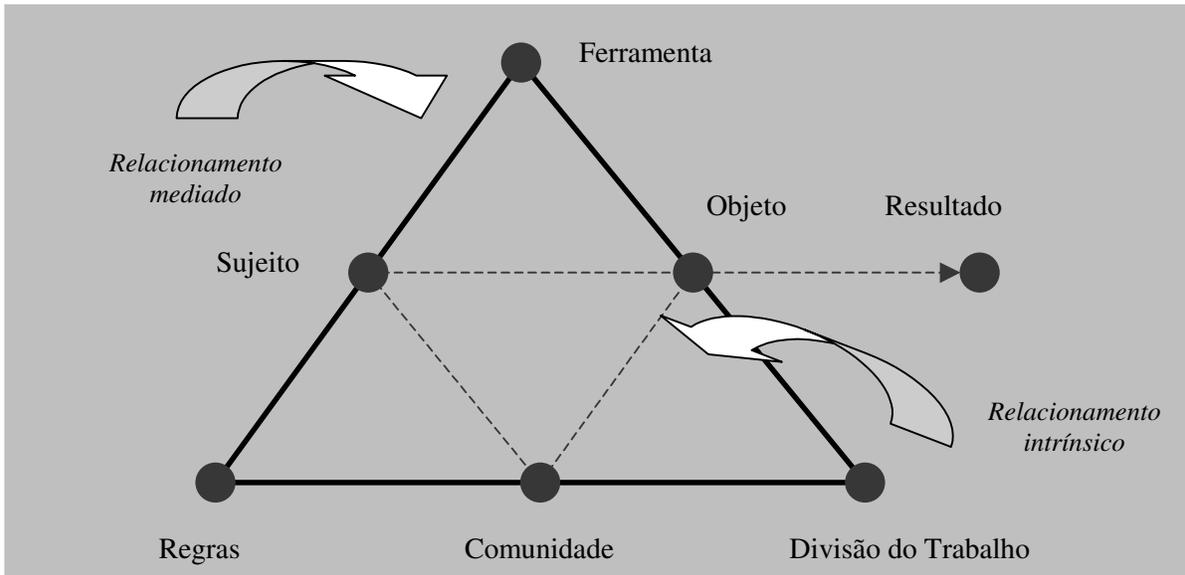


Figura 5.7 – Tipos de relacionamento representados no diagrama de Engeström

A Figura 5.7 destaca dois tipos de relacionamentos que existem entre os elementos da atividade, os quais denominamos de *relacionamento intrínstico* e *relacionamento mediado*. No contexto da atividade, existe um relacionamento intrínstico entre os elementos sujeito, objeto e comunidade, destacado no diagrama de Engeström através das linhas pontilhadas. No entanto, durante a realização da atividade, estes elementos somente podem se relacionar de forma mediada, através dos elementos de mediação ferramenta, regras e divisão do trabalho. O relacionamento mediado é que garante a materialização do relacionamento intrínstico. O relacionamento mediado é representado no diagrama de Engeström através das linhas sólidas.

5.3.3 Descrever a Estrutura Hierárquica da Atividade

Esta etapa da metodologia se baseia no *princípio (3)* da Teoria da Atividade, que divide a atividade em três níveis hierárquicos: atividade, ação e operação. Quando chegamos neste ponto do processo, já temos delineado o contexto formado pela atividade, que deve agora ser estruturada em termos de suas ações e operações.

Segundo a Teoria da Atividade, as ações e operações realizadas por um sujeito somente podem ser entendidas dentro do contexto da atividade, que passa a ser um referencial básico para a compreensão dessas ações e operações.

Do ponto de vista da elicitação de requisitos, esta é uma etapa fundamental, pois as ações e operações podem ter um correspondente direto com os requisitos funcionais do sistema. Essa idéia se baseia no fato de que grande parte dos sistemas de software desenvolvidos automatizam (total ou parcialmente) procedimentos executados inicialmente por pessoas, e portanto embutem ações e operações realizadas por elas. Muitas das ações e operações identificadas poderão ser mapeadas como requisitos funcionais do sistema.

5.3.3.1 Identificar as Ações e Operações da Atividade

De acordo com a *definição (3)* uma ação é um passo consciente realizado (pelo sujeito da atividade) com o intenção de se atingir uma meta bem definida no contexto da atividade, e a *definição (5)* declara que uma operação é uma ação que se tornou rotineira no contexto da atividade, de tal forma que ela é realizada de forma automática pelo sujeito. Portanto, buscamos nesta etapa da metodologia identificar “passos” conscientes, e não conscientes (automáticos), realizados pelos sujeitos das atividades durante a execução das mesmas.

As regras e a divisão do trabalho, identificadas durante o delineamento da atividade, contribuem significativamente para a identificação das ações e operações, pois estas materializam o comportamento (interno e externo) do sujeito, que precisa levar em conta regras de conduta na comunidade (ou comunidades) em que se encontra inserido.

A divisão do trabalho define papéis e responsabilidades no processo de transformação do objeto da atividade, estes papéis orientam a realização de muitas ações do sujeito.

5.3.3.2 Descrever as Metas das Ações

Na medida em que as ações vão sendo reveladas no processo de elicitação, suas metas também vão surgindo, pois, baseados na *definição (4)*, uma meta é um objetivo imediato a ser atingido por uma ação.

A descrição das metas das ações complementa o entendimento sobre as ações que o sujeito realiza, ligando-as aos resultados esperados da atividade. O resultado da atividade será atingido na medida em que as metas vão sendo alcançadas pela execução das ações.

5.3.3.3 Descrever as Condições de Realização das Operações

Esta é a última etapa da metodologia proposta, onde vamos descrever as condições de realização das operações identificadas nas atividades, completando assim a captura de todos os elementos relativos às atividades que foram identificadas no contexto do sistema.

Para que uma operação seja executada ela deve estar sob condições reconhecidas pelo sujeito, caso contrário a operação, que antes era realizada de forma automática, precisará se basear em um novo modelo mental, deixando de ser automática e retornando ao nível de ação.

Segundo a *definição (6)*, as condições para a realização de uma operação são formadas por um conjunto de variáveis que possuindo um determinado estado determina a execução de uma operação. Estas variáveis, com seus respectivos estados, precisam ser identificadas para que possamos ter uma descrição das condições de realização da operação.

A *Tabela 5.1* apresenta um resumo das definições e princípios que foram adotados em cada uma das etapas da metodologia. No próximo capítulo, apresentamos um estudo de caso onde aplicamos na íntegra a metodologia de elicitação de requisitos proposta.

Tabela 5.1 – Definições e princípios da Teoria da Atividade utilizados em cada etapa da metodologia

<i>Etapas</i>	<i>Procedimentos</i>	<i>Definições</i>	<i>Princípios</i>
1. Divisão do problema em atividades	1.1 Levantar atividades candidatas	(1)	(1)
	1.2 Selecionar atividades	(1,3,5)	(3)
	1.3 Descrever histórico das atividades selecionadas	-	(6)
2. Delineamento do contexto das atividades	2.1 Identificar os motivos e resultados da Atividade	(2,14)	(2)
	2.2 Identificar os elementos no nível individual	(7,8,9,10)	(2,5)
	2.3 Identificar os elementos no nível social	(11,12,13)	(2,5)
	2.4 Modelar a atividade através do diagrama de Engeström	(7,8,9,10, 11,12,13)	(5)
3 Descrição da estrutura hierárquica das atividades	3.1 Identificar as ações e operações da atividade	(3,5)	(3)
	3.2 Descrever as metas das ações	(4)	(3)
	3.3 Descrever as condições de realização das operações	(5,6)	(3)

Conclusão do Capítulo

Neste capítulo apresentamos de forma detalhada uma nova metodologia de elicitação de requisitos, que tem como princípio básico a organização do processo de elicitação de requisitos em torno do conceito de atividade.

Oferecemos definições para os conceitos da Teoria da Atividade que possam ser aplicados diretamente no processo de elicitação de requisitos.

Dividimos a metodologia em 3 fases, onde destacamos os procedimentos necessários para a execução de cada fase, os princípios da Teoria da Atividade e as definições dos conceitos da Teoria da Atividade aplicáveis em cada procedimento, e os resultados esperados em cada procedimento.

Capítulo 6

Estudo de Caso

6. Estudo de Caso

6.1 Elicitação de Requisitos para um Software de Controle de Protocolos da Secretaria de uma Faculdade

O estudo de caso apresentado neste capítulo é sobre a elicitação de requisitos para um software que deve controlar protocolos da secretaria de uma faculdade. A declaração do problema a seguir foi obtida a partir de uma situação real, onde foram realizadas entrevistas do tipo aberta [Gog97] com as secretárias que trabalhavam na faculdade¹⁷.

6.1.1 Descrição Inicial do Problema

O sistema de protocolos consiste em controlar os documentos que entram e saem da secretaria da faculdade. Para qualquer documento que entra ou sai da faculdade deve ser criado um protocolo que registra a “passagem” deste documento pela secretaria da faculdade. A secretária que recebe o documento gera um número de protocolo para o documento e preenche os campos do quadro de registro do protocolo (conforme *Figura 6.1*).

(1) Protocolo n ° ____	(4)Origem	(5)Data	(6)Rec	(7)Destino	(8)Data	(9)Rec
(2) Nome: _____						
(3)Assunto: _____						

Figura 6.1 - Quadro de registro do protocolo

Os campos apresentados no quadro de registro do protocolo têm o seguinte significado:

1. Número do protocolo

¹⁷ Ao todo foram realizadas quatro entrevistas, com três secretárias. As entrevistas foram feitas individualmente, e gravadas com a permissão das entrevistadas.

2. Nome de quem encaminhou o documento (que pode ser o nome de uma pessoa ou de um local/setor da Universidade)
3. Assunto
4. Origem (de onde veio/quem assinou)
5. Data do recebimento
6. Quem recebeu (nome do responsável pela criação do protocolo)
7. Destino (para quem veio/vai o documento)
8. Data que o documento foi encaminhado para o destinatário
9. Assinatura do destinatário (ou alguma informação de como chegou para o destinatário)

Essas informações são anotadas de forma a se manter registros sobre os documentos que circulam pela secretaria da faculdade. Esses registros são importantes para comprovar o envio e o recebimento dos mesmos, bem como para consultas posteriores. Num mesmo protocolo podem ser registradas até cinco "movimentações" do documento em questão (entrada ou saída do mesmo documento/processo), que correspondem ao número de linhas que podem ser utilizadas para cada "movimentação" no quadro de registro do protocolo.

As secretárias frequentemente necessitam consultar protocolos já realizados. Esta situação normalmente é necessária quando alguém da faculdade precisa localizar um documento que foi enviado ou recebido há algum tempo.

No momento em que foi feita a descrição do problema o controle dos protocolos era feito de forma manual. Os protocolos ficam registrados num livro, onde cada folha do livro comporta até 10 registros de protocolo. Todas as folhas são numeradas. Os livros de protocolo são editados e impressos pela gráfica da Universidade, e são enviados para as faculdades conforme a solicitação das mesmas.

6.1.2 Abordagem Adotada para a Elicitação dos Requisitos

A abordagem que adotamos para a elicitação dos requisitos do estudo de caso se baseou na metodologia proposta no Capítulo 5, consistindo das seguintes etapas:

1. Divisão do problema em atividades realizadas no contexto do sistema, que consistiu nos seguintes procedimentos:
 - 1.1. Levantamento das atividades candidatas
 - 1.2. Seleção das atividades
 - 1.3. Descrição do histórico de cada atividade selecionada
2. Delineamento do contexto de cada atividade, que consistiu nos seguintes procedimentos :
 - 2.1. Identificação dos motivos e resultados da atividade
 - 2.2. Identificação dos elementos das atividades no nível individual
 - 2.3. Identificação dos elementos das atividades no nível social
 - 2.4. Modelagem das atividades através do diagrama de Engeström
3. Descrição da estrutura hierárquica de cada atividade, que consistiu nos seguintes procedimentos:
 - 3.1. Identificação das ações e operações da atividade
 - 3.2. Descrição das metas das ações
 - 3.3. Descrição das condições de realização das operações

A seguir comentaremos cada etapa da metodologia aplicada ao estudo de caso, apresentando os resultados obtidos nas mesmas.

1. Divisão do problema em atividades realizadas no contexto do sistema

Conforme preconiza a metodologia de elicitação de requisitos proposta neste trabalho, a primeira etapa do processo de elicitação é a divisão do problema em unidades de elicitação de requisitos. Estas unidades de elicitação correspondem às atividades identificadas no contexto do problema, conforme declarado na *definição (1)* do Capítulo 5.

1.1 Levantamento das Atividades Candidatas

A partir das entrevistas realizadas com as secretárias da faculdade e baseando-se na *definição (1)* da metodologia, levantamos inicialmente as seguintes tarefas como atividades candidatas:

- Criar protocolo
- Gerar número de protocolo
- Preencher quadro de registro do protocolo
- Atualizar quadro de registro do protocolo
- Pesquisar número de protocolo
- Consultar protocolo por data
- Consultar protocolo por assunto

1.2 Seleção das Atividades

As atividades levantadas inicialmente passaram por uma análise mais detalhada, pois as mesmas foram confrontadas com as *definições (1) (3) e (5)* onde buscamos identificar se as atividades abrangiam ações e/ou operações relevantes do problema descrito. Como resultado deste procedimento selecionamos quatro atividades:

- Criar protocolo
- Atualizar quadro de registro do protocolo
- Consultar protocolo por data
- Consultar protocolo por assunto

As tarefas *gerar número de protocolo, preencher quadro de registro de protocolo e pesquisar número de protocolo* foram desconsideradas como atividades pois são realizadas dentro do contexto das atividades selecionadas, e portanto deverão ser tratadas como ações ou operações (são analisadas na etapa 3).

1.3 Descrição do histórico de cada atividade selecionada

Antes de descrevermos as atividades selecionadas nos seus “estados atuais”, iniciamos com um levantamento sobre a evolução histórica das atividades. Esta evolução histórica proporciona uma

Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade

visão dinâmica da atividade, auxiliando no entendimento do porque a atividade está sendo realizada dessa forma no momento atual.

Para compreendermos melhor as atividades selecionadas no estudo de caso levantamos dados históricos do desenvolvimento de cada atividade, conforme apresentado na *Tabela 6.1*.

Tabela 6.1 – Descrição dos históricos das atividades

<i>Criar Protocolo</i>	<i>Atualizar Quadro de Registro do Protocolo</i>	<i>Consultar Protocolo por Data</i>	<i>Consultar Protocolo por Assunto</i>
<ul style="list-style-type: none"> • Os setores da universidade que iniciaram com o controle de protocolos foram Reitoria e Vice-Reitorias, difundindo-se posteriormente para os demais setores da universidade • No início não havia controle formal dos protocolos, os setores criavam controles próprios • Alguns setores possuíam um livro onde registravam um resumo do assunto que tratava o documento enviado/recebido, a data do envio/recebimento e para quem foi/veio o documento 	<ul style="list-style-type: none"> • Atividade inexistente, pois no início não havia o quadro de registro do protocolo • Toda vez que o mesmo documento retornava para o setor, um novo registro de protocolo era feito no livro (isto acarretava em duplicação desnecessária de informações) 	<ul style="list-style-type: none"> • Quando havia registro do protocolo em algum livro, era feita de forma seqüencial no mesmo • Dificilmente esta consulta tinha sucesso, pois a informalidade fazia com que muitos documentos não fossem protocolados 	<ul style="list-style-type: none"> • Quando havia registro do protocolo em algum livro, era feita de forma seqüencial no mesmo • Dificilmente esta consulta tinha sucesso, pois a informalidade fazia com que muitos documentos não fossem protocolados

2. Delineamento do contexto de cada atividade

Após a divisão do problema em unidades de elicitação, que são as atividades, delineamos o contexto de cada atividade selecionada. Nesta etapa fizemos uso das *definições* (2) (7) (8) (9) (10) (11) (12) (13) e (14), conforme apresentadas na metodologia proposta.

2.1 Identificação dos motivos e resultados da atividade

Para cada atividade identificamos seus motivos e resultados esperados, conforme descrito na *Tabela 6.2*. Para a identificação desses elementos da atividade nos baseamos nas *definições* (2) e (14).

Tabela 6.2 – Descrição dos motivos e resultados das atividades

	<i>Criar Protocolo</i>	<i>Atualizar Quadro de Registro do Protocolo</i>	<i>Consultar Protocolo por Data</i>	<i>Consultar Protocolo por Assunto</i>
Motivos	<ul style="list-style-type: none"> • Necessidade de controle dos documentos que entram e saem da Faculdade • Necessidade de saber o destino final de um documento que entrou ou saiu da Faculdade • Necessidade de comprovação do envio/recebimento do documento 	<ul style="list-style-type: none"> • Necessidade de rastreamento de um documento já protocolado inicialmente • Necessidade de comprovação do envio/recebimento do documento 	<ul style="list-style-type: none"> • Necessidade de confirmar se um documento foi realmente enviado/recebido pela Faculdade • Necessidade de resgatar um documento tendo como base algum intervalo de tempo 	<ul style="list-style-type: none"> • Necessidade de confirmar se um documento foi realmente enviado/recebido pela Faculdade • Necessidade de resgatar um documento tendo como base o assunto tratado no mesmo
Resultados	<ul style="list-style-type: none"> • Protocolo realizado 	<ul style="list-style-type: none"> • Protocolo atualizado 	<ul style="list-style-type: none"> • Localização do documento pesquisado 	Localização do documento pesquisado

2.2 Identificação dos elementos das atividades no nível individual

Após a descrição dos motivos e resultados de cada atividade, nos baseamos nas *definições (8) (9) (10) e (11)* para identificar os elementos sujeito, ferramenta (técnicas e psicológicas) e objeto das atividades, conforme descrito na *Tabela 6.3*.

Tabela 6.3 – Descrição dos elementos das atividades no nível individual

	<i>Criar Protocolo</i>	<i>Atualizar Quadro de Registro do Protocolo</i>	<i>Consultar Protocolo por Data</i>	<i>Consultar Protocolo por Assunto</i>
Sujeitos	Secretária da Faculdade	Secretária da Faculdade	Secretária da Faculdade	Secretária da Faculdade
Ferramentas Técnicas	Caneta Carimbo	Caneta	Nenhuma	Nenhuma
Ferramentas Psicológicas	Capacidade de Escrita	Capacidade de Escrita	Capacidade de Leitura	Capacidade de Leitura
Objetos	<ul style="list-style-type: none"> • Quadro de Registro do Protocolo • Documento Encaminhado 	<ul style="list-style-type: none"> • Quadro de Registro do Protocolo • Linha de “Movimentação” • Documento Encaminhado 	<ul style="list-style-type: none"> • Livro de Protocolos • Quadro de Registro do Protocolo • Linha de “Movimentação” 	<ul style="list-style-type: none"> • Livro de Protocolos • Quadro de Registro do Protocolo • Linha de “Movimentação”

2.3 Identificação dos elementos da atividade no nível social

Após a identificação dos elementos das atividades no nível individual, nos baseamos nas *definições (12) (13) e (14)* para identificar os elementos comunidade, regras e divisão do trabalho das atividades, ou seja, elementos do nível social, descritos na *Tabela 6.4*.

**Uma Metodologia de Elicitação de Requisitos de Software
Baseada na Teoria da Atividade**

Tabela 6.4 – Descrição dos elementos das atividades no nível social

	<i>Criar Protocolo</i>	<i>Atualizar Quadro de Registro do Protocolo</i>	<i>Consultar Protocolo por Data</i>	<i>Consultar Protocolo por Assunto</i>
Comunidade	<ul style="list-style-type: none"> • Secretárias • Professores • Bolsistas • Alunos 	<ul style="list-style-type: none"> • Secretárias • Professores • Bolsistas • Alunos 	<ul style="list-style-type: none"> • Secretárias • Professores • Bolsistas • Alunos 	<ul style="list-style-type: none"> • Secretárias • Professores • Bolsistas • Alunos
Regras	<ul style="list-style-type: none"> • O campo <u>número do protocolo</u> deve ser gerado seqüencialmente. O novo número gerado deve ser igual ao número do último protocolo mais um • Os campos <u>número do protocolo e nome</u> devem ser destacados, utilizando-se cor vermelha para o preenchimento destes campos • Todos os campos do quadro de registro do protocolo devem ser preenchidos, com exceção do campo <u>origem</u>, que quando tem o mesmo conteúdo do campo <u>assunto</u>, pode ser omitido • Todos os documentos protocolados no livro de protocolos devem receber um carimbo constando a data em que o documento chegou na secretaria da faculdade e o número do 	<ul style="list-style-type: none"> • Todos os campos da linha de atualização do protocolo devem ser preenchidos • Se não houver mais linha de atualização disponível no quadro, deve ser criado um novo protocolo 	<ul style="list-style-type: none"> • Deve ser informado a data de recebimento do documento, que será utilizada para consultar o livro de protocolos 	<ul style="list-style-type: none"> • Deve ser informado o assunto do documento, que será utilizado para consultar o livro de protocolos

**Uma Metodologia de Elicitação de Requisitos de Software
Baseada na Teoria da Atividade**

	protocolo gerado para esse documento			
Divisão do Trabalho	<ul style="list-style-type: none"> • As secretárias são responsáveis pelo envio/recebimento dos documentos que entram e saem da Faculdade • Os professores, bolsistas e alunos recebem e solicitam o envio dos documentos 	<ul style="list-style-type: none"> • As secretárias são responsáveis pelo envio e recebimento dos documentos que entram e saem da Faculdade • Os professores, bolsistas e alunos recebem e solicitam o envio dos documentos 	<ul style="list-style-type: none"> • As secretárias realizam as consultas solicitadas • Os professores, bolsistas e alunos solicitam a localização de documentos 	<ul style="list-style-type: none"> • As secretárias realizam as consultas solicitadas • Os professores, bolsistas e alunos solicitam a localização de documentos

(continuação da Tabela 6.4)

2.4 Modelagem das atividades através do diagrama de Engeström

Após termos identificado e descrito todos os elementos da atividade, tanto no nível individual como no nível social, temos condições de oferecer uma representação pictórica da mesma através do diagrama de Engeström, que traz uma visão dos relacionamentos existentes entre os elementos que definem o contexto da atividade.

Nas *Figuras 6.2 a 6.5* apresentamos a modelagem das atividades descritas nas seções anteriores. A modelagem das atividades se baseia no diagrama de Engeström (também chamado de modelo sistêmico), conforme sugerido na metodologia proposta.

Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade

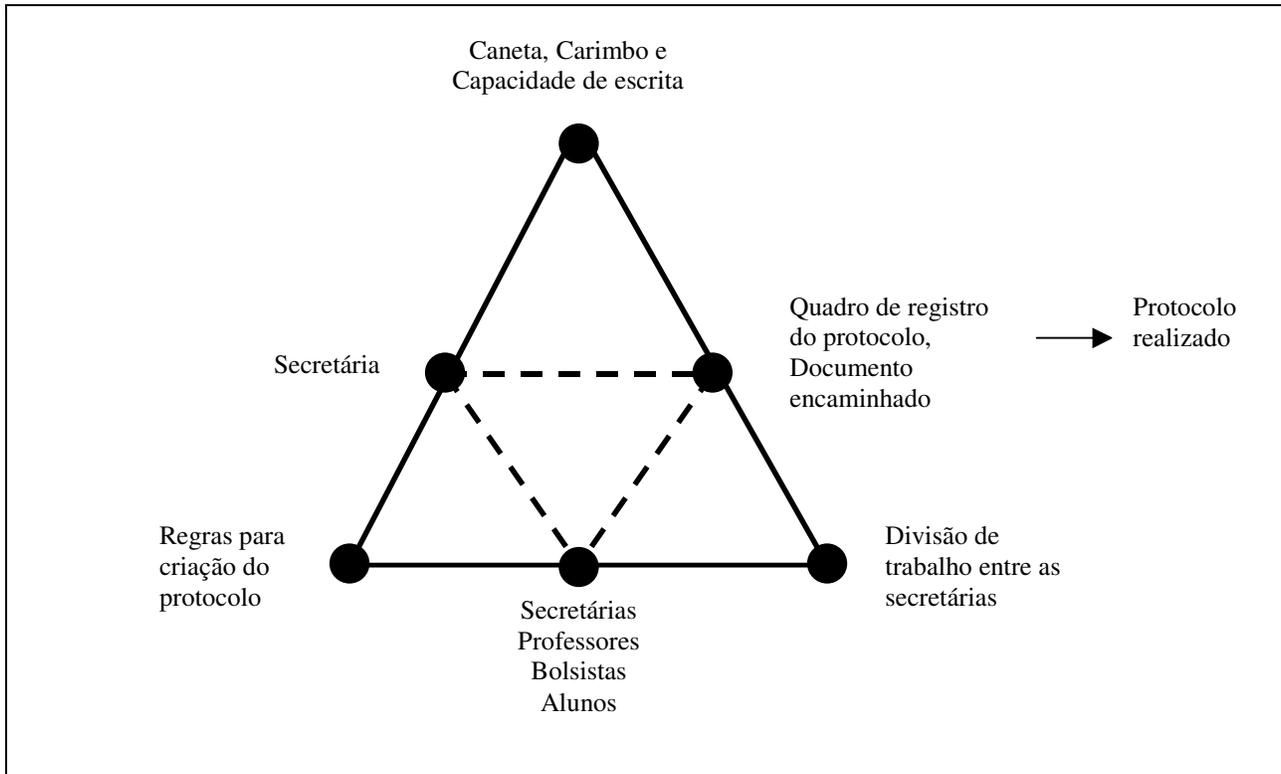


Figura 6.2 - Modelo sistêmico para a atividade "Criar Protocolo"

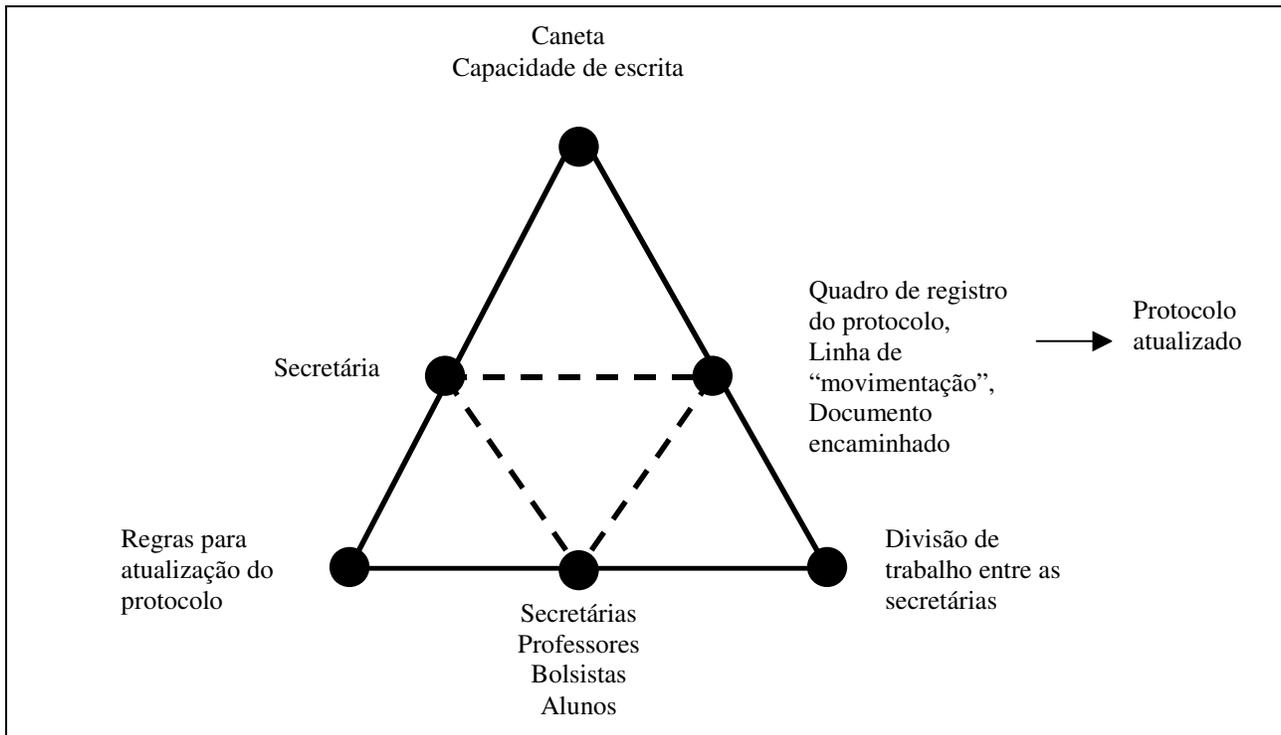


Figura 6.3 - Modelo sistêmico para a atividade "Atualizar Quadro de Registro do Protocolo"

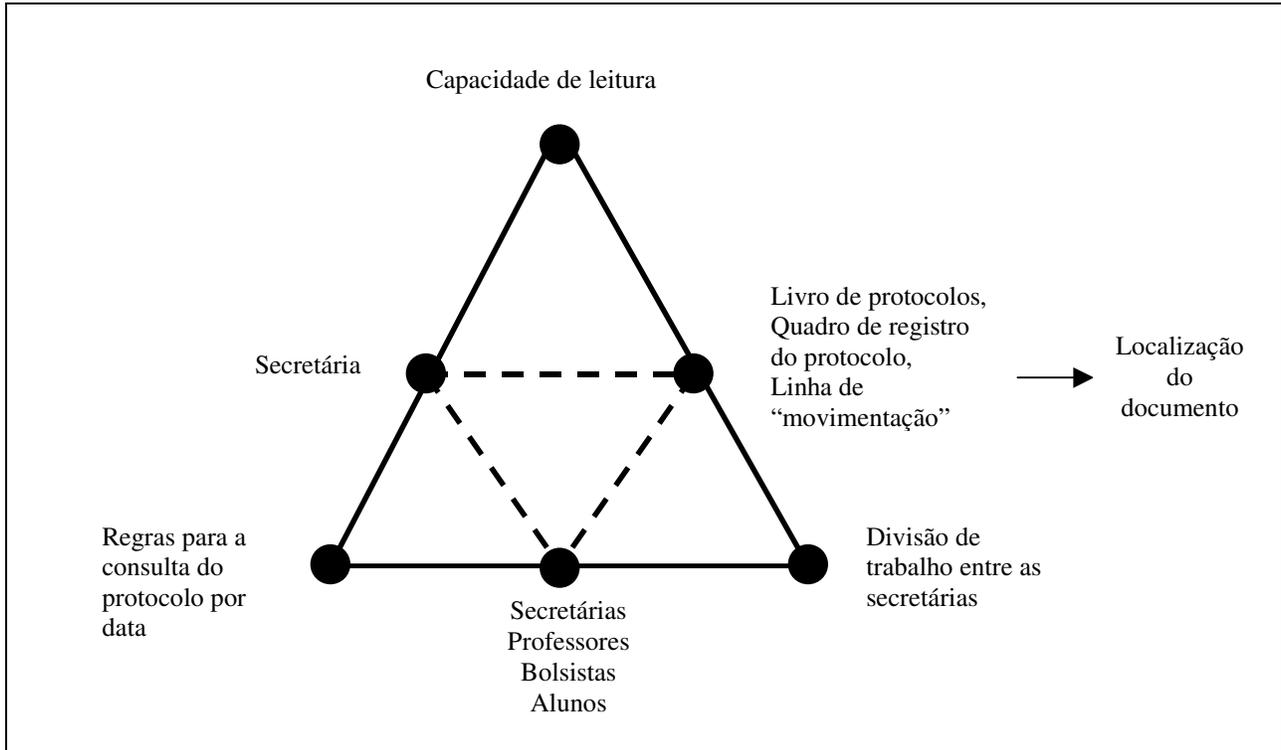


Figura 6.4 - Modelo sistêmico para a atividade “Consultar Protocolo por Data”

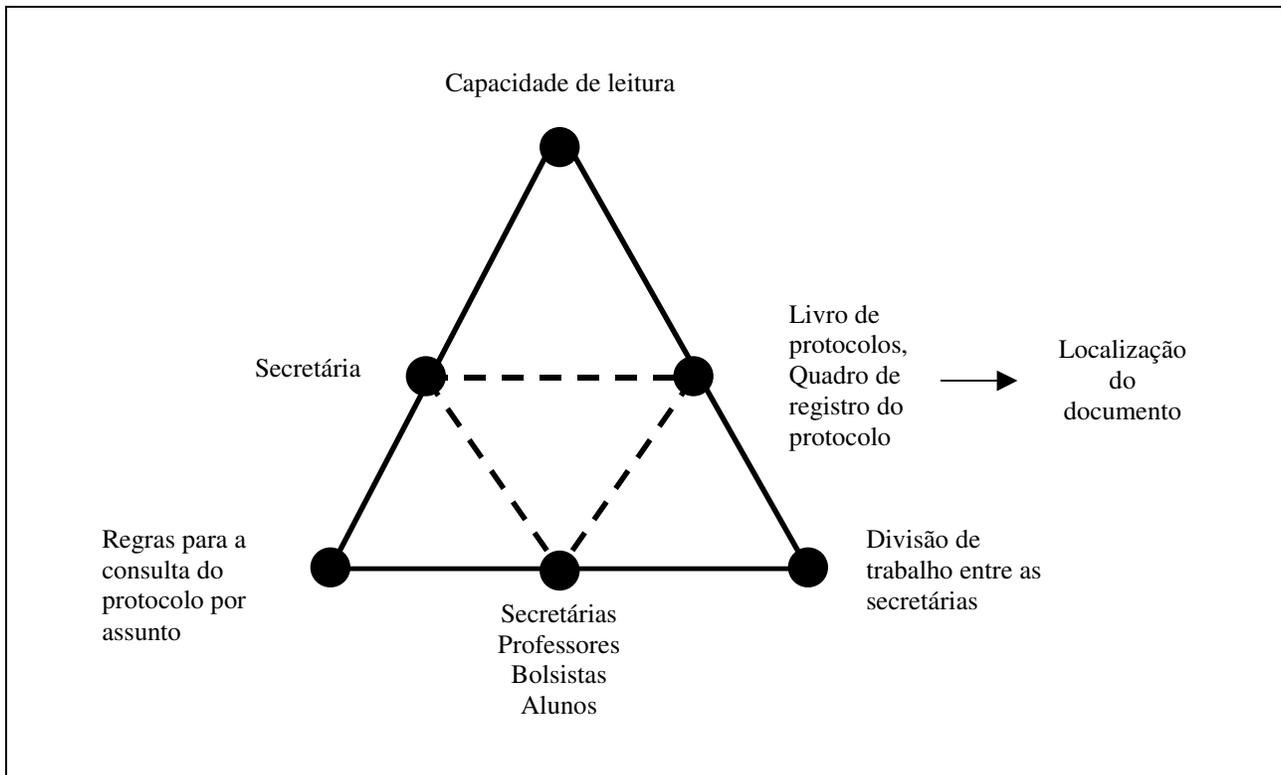


Figura 6.5 - Modelo sistêmico para a atividade “Consultar Protocolo por Assunto”

3. Descrição da estrutura hierárquica de cada atividade

Após definido o contexto de cada atividade, passamos para uma descrição da estrutura hierárquica das mesmas, ou seja, a definição das ações e operações que compõem as atividades e suas respectivas metas e condições de realização. Para isso nos baseamos nas *definições* (3) (4) (5) e (6).

3.1 Identificação das ações e operações da atividade

Nas *Tabelas 6.5 a 6.8* apresentamos a decomposição das atividades em ações e operações. Esta decomposição se baseia fortemente nos elementos que foram identificados nas atividades no nível social, como regras, comunidade e divisão do trabalho.

Tabela 6.5 - Decomposição da atividade “Criar Protocolo”

Atividade	Ações	Operações	
Criar protocolo	Gerar número do protocolo	Verificar número do último protocolo	
		Adicionar um ao número do último protocolo	
		Preencher campo de "número do protocolo"(1)	
	Preencher 1ª linha do quadro de registro de protocolos	Preencher campo (2)	
		Preencher campo (3)	
		Preencher campo (4)	
		Preencher campo (5)	
		Preencher campo (6)	
		Preencher campo (7)	
		Preencher campo (8)	
	Encaminhar documento	Preencher campo (9)	
		Carimbar documento a ser enviado	
		Copiar número do protocolo no documento carimbado	
			Enviar documento para destinatário

Tabela 6.6 - Decomposição da atividade “Atualizar Quadro de Registro do Protocolo”

Atividade	Ações	Operações
Atualizar protocolo	Encontrar quadro de registro de protocolo no livro de protocolos	Verificar número do protocolo no documento recebido
		Buscar número correspondente no livro de protocolos
	Preencher próxima linha do quadro de registro do protocolo	Preencher campo (4)
		Preencher campo (5)
		Preencher campo (6)
		Preencher campo (7)
		Preencher campo (8)
		Preencher campo (9)

Tabela 6.7 - Decomposição da atividade “Consultar Protocolo por Data”

Atividade	Ações	Operações
Consultar protocolo por data	Encontrar protocolos baseado num intervalo de tempo	Especificar data para consulta
		Buscar números de protocolos de acordo com a data especificada
		Informar protocolos encontrados

Tabela 6.8 - Decomposição da atividade “Consultar Protocolo por Assunto”

Atividade	Ações	Operações
Consultar protocolo por assunto	Encontrar protocolos baseado num assunto	Especificar assunto para consulta
		Buscar números de protocolos de acordo com o assunto especificado
		Informar protocolos encontrados

3.2 Descrição das metas das ações

Baseado na *definição (3)*, toda ação tem a intenção de atingir uma ou mais metas bem definidas no contexto da atividade. Nas *Tabelas 6.9 a 6.12* apresentamos as metas das ações que compõem as atividades selecionadas.

Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade

Tabela 6.9 – Descrição das metas das ações que compõem a atividade “Criar Protocolo”

Atividade	Ações	Metas
Criar protocolo	Gerar número do protocolo	<ul style="list-style-type: none"> • Criar uma identificação para o documento que está sendo protocolado
	Preencher 1ª linha do quadro de registro de protocolos	<ul style="list-style-type: none"> • Manter dados sobre o envio/recebimento do documento protocolado
	Encaminhar documento	<ul style="list-style-type: none"> • Comprovar o registro do documento • Enviar documento para o seu destino

Tabela 6.10 - Descrição das metas das ações que compõem a atividade “Atualizar Quadro de Registro do Protocolo”

Atividade	Ações	Metas
Atualizar protocolo	Encontrar quadro de registro de protocolo no livro de protocolos	<ul style="list-style-type: none"> • Buscar quadro de registro do protocolo cujo número de protocolo seja igual ao do documento em mãos
	Preencher próxima linha do quadro de registro do protocolo	<ul style="list-style-type: none"> • Manter dados sobre o envio/recebimento do documento protocolado

Tabela 6.11 - Descrição das metas das ações que compõem a atividade “Consultar Protocolo por Data”

Atividade	Ações	Metas
Consultar protocolo por data	Encontrar protocolos baseado num intervalo de tempo	<ul style="list-style-type: none"> • Buscar quadros de registro de protocolos cujas datas estejam no intervalo de tempo especificado

Tabela 6.12 - Descrição das metas das ações que compõem a atividade “Consultar Protocolo por Assunto”

Atividade	Ações	Metas
Consultar protocolo por assunto	Encontrar protocolos baseado num assunto	<ul style="list-style-type: none"> • Buscar quadros de registro de protocolos cujos assuntos estejam relacionados ao assunto especificado

3.3 Descrição das condições de realização das operações

De acordo com as *definições (5) e (6)* uma operação é executada dependendo das condições existentes no momento de sua execução. Assim, é importante descrever todas as condições de execução das operações que compõem as atividades selecionadas. Nas *Tabelas 6.13 a 6.16* são descritas as condições de execução das operações identificadas no estudo de caso.

Tabela 6.13 – Descrição das condições de realização das operações da atividade “Criar Protocolo”

Atividade	Ações	Operações	Condições de Realização
Criar protocolo	Gerar número do protocolo	Verificar número do último protocolo	<ul style="list-style-type: none"> Ter acesso ao livro de protocolos
		Adicionar um ao número do último protocolo	<ul style="list-style-type: none"> Ter disponível o último número de protocolo gerado
		Preencher campo de “número do protocolo”(1)	<ul style="list-style-type: none"> Ter disponível o novo número de protocolo gerado
	Preencher 1 linha do quadro de registro de protocolos	Preencher campo (2)	<ul style="list-style-type: none"> Ter disponível o nome de quem encaminhou o documento
		Preencher campo (3)	<ul style="list-style-type: none"> Ter disponível o assunto de que trata o documento
		Preencher campo (4)	<ul style="list-style-type: none"> Ter disponível a origem do documento
		Preencher campo (5)	<ul style="list-style-type: none"> Ter disponível a data atual
		Preencher campo (6)	<ul style="list-style-type: none"> Ter disponível o nome de quem recebeu o documento
		Preencher campo (7)	<ul style="list-style-type: none"> Ter disponível o destino do documento

**Uma Metodologia de Elicitação de Requisitos de Software
Baseada na Teoria da Atividade**

		Preencher campo (8)	<ul style="list-style-type: none"> Ter disponível a data em que o documento foi encaminhado para o destinatário
		Preencher campo (9)	<ul style="list-style-type: none"> Ter o destinatário disponível para assinatura
	Encaminhar documento	Carimbar documento a ser enviado	<ul style="list-style-type: none"> Quadro de registro do protocolo estar preenchido
		Copiar número do protocolo no documento carimbado	<ul style="list-style-type: none"> Ter disponível o novo número de protocolo gerado
		Enviar documento para destinatário	<ul style="list-style-type: none"> Ter <i>office-boy</i> disponível para o envio

(continuação da Tabela 6.13)

Tabela 6.14 - Descrição das condições de realização das operações da atividade “Atualizar Quadro de Registro do Protocolo”

Atividade	Ações	Operações	Condições de Realização
Atualizar protocolo	Encontrar quadro de registro de protocolo no livro de protocolos	Verificar número do protocolo no documento recebido	<ul style="list-style-type: none"> Ter número do protocolo disponível
		Buscar número correspondente no livro de protocolos	<ul style="list-style-type: none"> Ter acesso ao livro de protocolos
	Preencher próxima linha do quadro de registro do protocolo	Preencher campo (4)	<ul style="list-style-type: none"> Ter disponível a origem do documento
		Preencher campo (5)	<ul style="list-style-type: none"> Ter disponível a data atual
		Preencher campo (6)	<ul style="list-style-type: none"> Ter disponível o nome de quem recebeu o documento
		Preencher campo (7)	<ul style="list-style-type: none"> Ter disponível o destino do documento

Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade

	Preencher campo (8)	<ul style="list-style-type: none"> Ter disponível a data em que o documento foi encaminhado para o destinatário
	Preencher campo (9)	<ul style="list-style-type: none"> Ter o destinatário disponível para assinatura

(continuação da Tabela 6.14)

Tabela 6.15 - Descrição das condições de realização das operações da atividade “Consultar Protocolo por Data”

Atividade	Ações	Operações	Condições de Realização
Consultar protocolo por data	Encontrar protocolos baseado num intervalo de tempo	Especificar data para consulta	<ul style="list-style-type: none"> Ter a data de recebimento do documento disponível
		Buscar números de protocolos de acordo com a data especificada	<ul style="list-style-type: none"> Ter acesso ao livro de protocolos
		Informar protocolos encontrados	<ul style="list-style-type: none"> Chegar a um resultado para a consulta

Tabela 6.16 - Descrição das condições de realização das operações da atividade “Consultar Protocolo por Assunto”

Atividade	Ações	Operações	Condições de Realização
Consultar protocolo por assunto	Encontrar protocolos baseado num assunto	Especificar assunto para consulta	<ul style="list-style-type: none"> Ter o assunto do documento disponível
		Buscar números de protocolos de acordo com o assunto especificado	<ul style="list-style-type: none"> Ter acesso ao livro de protocolos
		Informar protocolos encontrados	<ul style="list-style-type: none"> Chegar a um resultado para a consulta

Conclusão do Capítulo

Da experiência obtida com o estudo de caso, podemos destacar que o processo de elicitação de requisitos pode de fato ser organizado em atividades, que passam a ser vistas como unidades de elicitação, embutindo uma série de conceitos relacionados à atividade. A atividade forma um contexto de análise onde as ações e operações que são identificadas neste contexto, em grande parte, revelam ser os requisitos do futuro sistema de software.

Um ponto importante a ser destacado é sobre a quantidade de tabelas (e as informações presentes nelas) que foram elaboradas para descrever os resultados da elicitação de requisitos desenvolvida. Embora o número de tabelas tenha sido elevado, é necessário lembrar que a atividade relaciona muitos conceitos, e todos eles nos parecem relevantes para o entendimento das reais necessidades da comunidade (ou comunidades) de usuários envolvida no sistema.

Portanto, nos parece adequado apontar que a metodologia proposta pode ser melhor aproveitada para a elicitação de requisitos em sistemas que visem atender aos anseios e necessidades de uma comunidade complexa, ou ainda de várias comunidades, o que nos leva a indicar o uso dessa metodologia para sistemas de software de natureza mais complexa do que o apresentado no estudo de caso, como por exemplo sistemas de informação corporativos.

Capítulo 7

Ligação da Metodologia de
Elicitação de Requisitos
Baseada em Atividades com
Algumas Técnicas de
Análise e Especificação de
Requisitos

7. Ligação da Metodologia de Elicitação de Requisitos Baseada em Atividades com Algumas Técnicas de Análise e Especificação de Requisitos

No Capítulo 5 apresentamos uma metodologia de elicitação de requisitos baseada na Teoria da Atividade, a qual aplicamos no estudo de caso apresentado no Capítulo 6, onde procuramos demonstrar a viabilidade de utilização da metodologia e os resultados que podem ser obtidos com a sua aplicação.

Como pode-se observar no estudo de caso realizado, a metodologia proposta oferece uma estrutura organizacional para o processo de elicitação de requisitos, no qual a atividade é reconhecida como uma unidade básica de elicitação, estabelecendo um contexto bem definido para o entendimento das ações realizadas pelos atores do sistema. Técnicas consagradas de elicitação de requisitos, como entrevistas, observação, análise de discurso, JAD, entre outras (conforme apresentadas na *Seção 3*) podem ser empregadas de maneira produtiva no âmbito da metodologia, que não é uma proposta de substituição dessas técnicas, mas sim uma proposta de estruturação do processo de elicitação de requisitos.

Consideramos, portanto, que a metodologia de elicitação de requisitos apresentada neste trabalho, vem somar esforços para a melhoria da qualidade do processo de elicitação de requisitos, oferecendo uma estrutura organizativa para o entendimento e captura dos requisitos de um sistema, centrada no conceito de atividade.

Nas seções a seguir abordamos como esta metodologia de elicitação de requisitos pode contribuir com algumas técnicas de análise e especificação de requisitos atualmente em uso pela comunidade de Engenharia de Requisitos. Apresentamos algumas formas de transição entre os resultados obtidos com a metodologia de elicitação proposta e algumas técnicas de análise e especificação de requisitos.

7.1 UML

A *UML* oferece um conjunto de diagramas que possibilita a modelagem de vários aspectos de um sistema de software. A modelagem desses aspectos é representada através dos diagramas, que constituem artefatos importantes para a construção e implementação de um software.

Entre os diagramas da *UML* (num total de nove [Boo99]) podemos destacar o diagrama de casos de uso como sendo o mais voltado para a modelagem inicial dos requisitos de um sistema. Além do diagrama de casos de uso, os diagramas de seqüência e de atividade também podem oferecer contribuição importante na modelagem dos requisitos, proporcionando um maior detalhamento destes durante as fases de análise e especificação dos requisitos [Kul00].

7.1.1 Da Elicitação Baseada em Atividades ao Diagrama de Casos de Uso

Conforme declarado na *definição (1)* da metodologia proposta uma “atividade é uma unidade de elicitação de requisitos que oferece um contexto mínimo para o entendimento de um conjunto de ações cooperantes que agem sobre um ou mais objetos, transformando-os num resultado”, de onde destacamos duas asserções: i) uma atividade abrange um conjunto de ações; e ii) uma atividade fornece algum resultado (para o sujeito da atividade ou qualquer outro sujeito da comunidade).

Segundo Jacobson [Jac98], um caso de uso “é um pedaço de funcionalidade que o sistema oferece para adicionar um resultado de valor para seus atores”. Jacobson ainda destaca que “mais estritamente, um caso de uso especifica uma seqüência de ações, incluindo alternativas de seqüência que o sistema pode executar, interagindo com atores do sistema”, de onde podemos destacar duas asserções: i) um caso de uso abrange um conjunto de ações; e ii) um caso de uso fornece um resultado de valor para seus atores.

De fato, tanto atividades quanto casos de uso são unidades que possuem ênfase na representação da funcionalidade de um sistema, ambas abrangendo um conjunto de ações que fornece um resultado para um ator do sistema (sujeito, na perspectiva da Teoria da Atividade). Portanto, uma relação direta entre o modelo de atividade preconizado na metodologia de elicitação proposta e o modelo de caso de uso, é o mapeamento de atividade para caso de uso e de sujeito para ator. A *Figura 7.1* fornece um modelo de casos de uso extraído da elicitação de requisitos desenvolvida no estudo de caso apresentado no *Capítulo 6*.

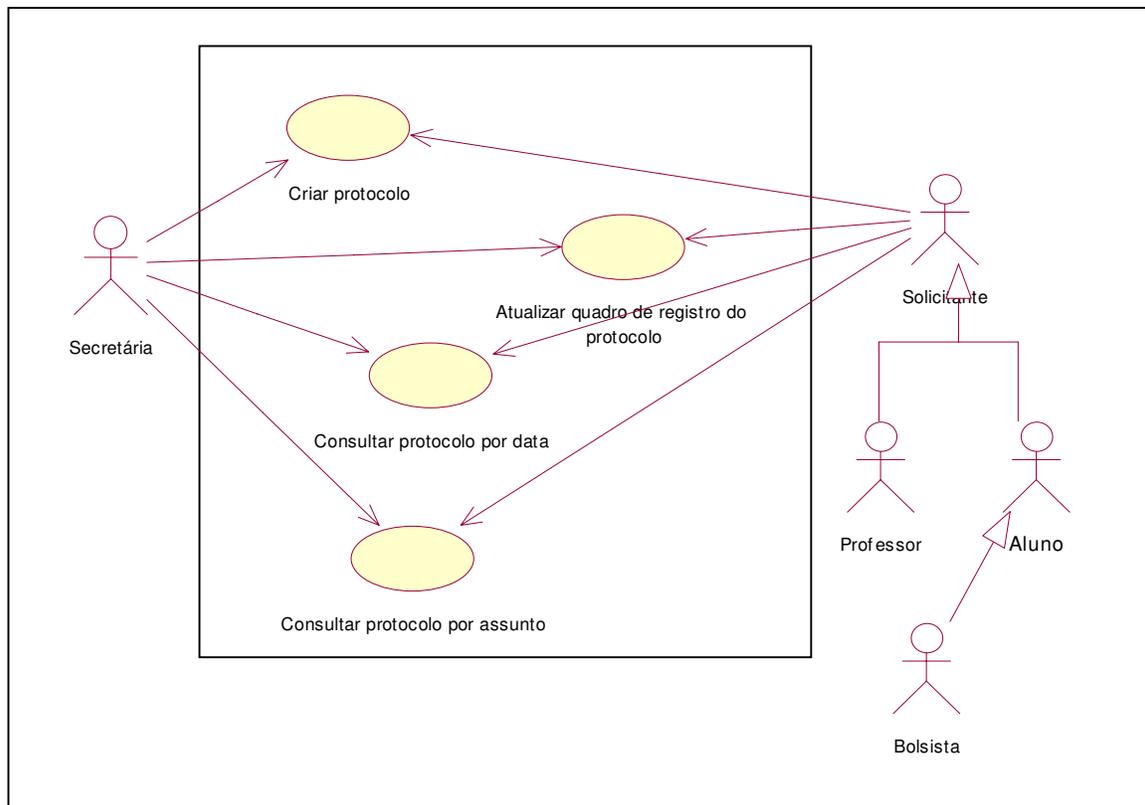


Figura 7.1 – Modelo de casos de uso para o sistema de protocolos

No modelo acima, as atividades “Criar protocolo”, “Atualizar Quadro de Registro do Protocolo”, “Consultar Protocolo por Data” e “Consultar Protocolo por Assunto”, elicidadas no estudo de caso, foram mapeadas diretamente como casos de uso, os sujeitos secretária (sujeito da atividade), professor, aluno e bolsista (demais sujeitos da comunidade onde ocorrem as atividades) foram mapeados como atores.

Conforme discutido na *Seção 3.9* a modelagem de casos de uso inclui, além do diagrama de casos de uso, uma descrição textual que deve ser elaborada para cada caso de uso apresentado no diagrama. A descrição do caso de uso deve especificar o que o sistema faz quando o caso de uso ocorre.

Adotamos, para efeito de ilustração, o formato de descrição de casos de uso proposto em [Rum97]. Neste formato o caso de uso é identificado pelo seu nome no início do quadro descritivo, seguido por resumo, atores, pré-condições, descrição, exceções e pós-condições. O nome do caso de uso o identifica entre os demais casos de uso do sistema. O resumo é uma breve descrição que procura oferecer uma visão geral do caso de uso. Os atores são os agentes externos ao sistema, envolvidos durante a ocorrência do caso de uso. As pré-condições descrevem estados e situações que devem estar em “ordem” para que o caso de uso possa ocorrer. A descrição é uma narrativa detalhada do curso básico de ações observado durante a ocorrência do caso de uso (a descrição também é chamada de curso básico de eventos [Kul00]). As exceções são cursos alternativos de execução do caso de uso, ou previsões de “erros” que podem ocorrer no desenvolvimento do mesmo. Finalmente, as pós-condições descrevem estados e situações que devem estar em “ordem” após o desenvolvimento do caso de uso.

As *Figuras 7.2 a 7.5* mostram os quadros descritivos dos casos de uso “Criar Protocolo”, “Atualizar Quadro de Registro do Protocolo”, “Consultar Protocolo por Data” e “Consultar Protocolo por Assunto”.

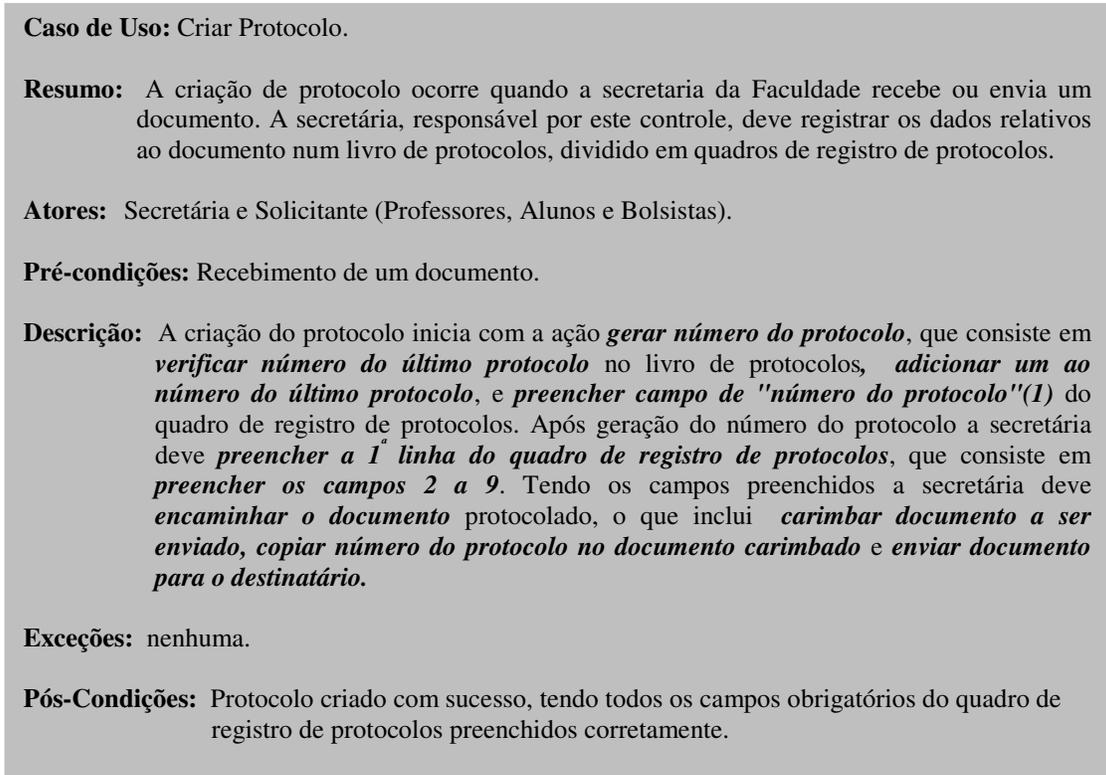


Figura 7.2 – Quadro descritivo do caso de uso “Criar Protocolo”

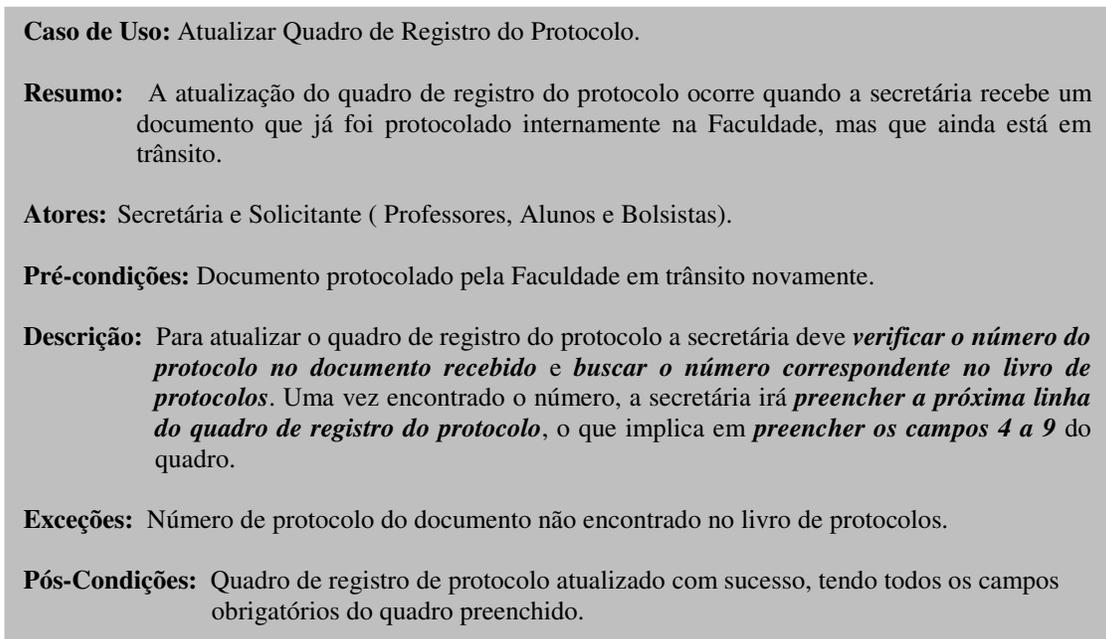


Figura 7.3 – Quadro descritivo do caso de uso “Atualizar Quadro de Registro do Protocolo”

Caso de Uso: Consultar Protocolo por Data.

Resumo: A consulta de protocolo por data ocorre quando um solicitante deseja saber onde se encontra um determinado documento, informando apenas a data em que o documento foi enviado.

Atores: Secretária e Solicitante (Professores, Alunos e Bolsistas).

Pré-condições: Ter conhecimento da data do documento desejado pelo solicitante.

Descrição: Na consulta de protocolo por data, a secretária, a partir do período indicado pelo solicitante, vai *especificar a data para consulta, buscar os números de protocolos de acordo com a data especificada e informar os protocolos encontrados* nesta data.

Exceções: Nenhum protocolo encontrado.

Pós-Condições: Protocolos encontrados encaminhados ao solicitante.

Figura 7.4 – Quadro descritivo do caso de uso “Consultar Protocolo por Data”

Caso de Uso: Consultar Protocolo por Assunto.

Resumo: A consulta de protocolo por assunto ocorre quando um solicitante deseja saber onde se encontra um determinado documento, informando apenas o assunto do documento.

Atores: Secretária e Solicitante (Professores, Alunos e Bolsistas).

Pré-condições: Ter conhecimento do assunto do documento desejado pelo solicitante.

Descrição: Neste tipo de consulta, a secretária vai *especificar o assunto para a consulta, buscar os números de protocolos de acordo com o assunto especificado e informar os protocolos encontrados* com este assunto.

Exceções: Nenhum protocolo encontrado.

Pós-Condições: Protocolos encontrados encaminhados ao solicitante.

Figura 7.5 – Quadro descritivo do caso de uso “Consultar Protocolo por Assunto”

No quadro descritivo, o tópico **descrição** é o componente central no detalhamento do caso de uso. Conforme mencionamos acima, a descrição é uma narrativa do curso básico de ações executado pelo caso de uso. Nas *Figuras 7.2 a 7.5* apresentamos o tópico descrição, para cada caso de uso, extraíndo as ações diretamente da decomposição das atividades realizadas no estudo de caso (conforme apresentado nas *Tabelas 6.5 a 6.8 da Seção 6.1.2*).

As frases que aparecem negritadas e em itálico no tópico descrição de cada caso de uso exemplificado, correspondem às ações e operações das atividades definidas no estudo de caso. Assim, demonstramos que o curso básico de ações de um caso de uso pode ser extraído diretamente, e de forma quase automática, das ações e operações de uma atividade.

Outros elementos do quadro descritivo do caso de uso também podem ser mapeados diretamente da atividade, como por exemplo os atores (mapeados a partir do sujeito e comunidade) e as pré-condições, exceções e pós-condições (que podem ser mapeadas a partir das regras de execução da atividade). A *Figura 7.6* ilustra este mapeamento.

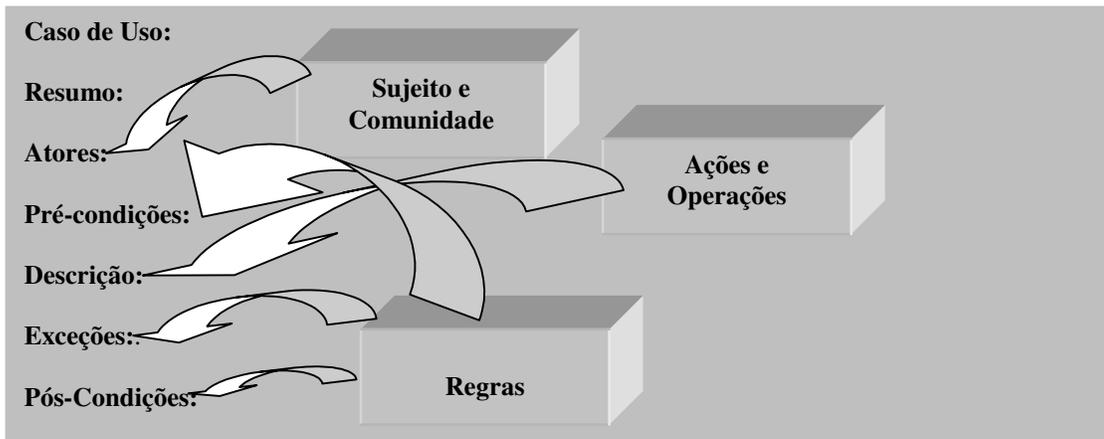


Figura 7.6 – Mapeamento dos elementos da atividade para os componentes do quadro descritivo do caso de uso

No entanto, quando descrevemos uma atividade, identificamos alguns elementos que não são mencionados na modelagem baseada em caso de uso, são eles: motivos, objetos, ferramentas e divisão do trabalho. A adição destes elementos para a descrição do caso de uso pode contribuir para um melhor entendimento do caso de uso, uma vez que são elementos ligados à funcionalidade descrita no caso de uso. A *Figura 7.7* apresenta um quadro descritivo para o caso de uso “Criar Protocolo” onde adicionamos os elementos da atividade que não possuem similar no quadro descritivo do caso de uso.

Os elementos que aparecem com asteriscos (*) no quadro descritivo do caso de uso foram adicionados a partir do contexto definido pela atividade “Criar Protocolo”. O item *motivo* destaca

o *porque* do caso de uso, a razão dele existir no contexto do sistema. O item *objetos* destaca os objetos que sofrem transformação durante a realização do caso de uso. O item *ferramenta* destaca os artefatos utilizados pelos atores durante a execução do caso de uso, e o item *divisão do trabalho* destaca as responsabilidades dos atores envolvidos no caso de uso.

Caso de Uso: Criar Protocolo.

Resumo: A criação de protocolo ocorre quando a secretária da Faculdade recebe ou envia um documento. A secretária, responsável por este controle, deve registrar os dados relativos ao documento num livro de protocolos, dividido em quadros de registro de protocolos.

Atores: Secretária e Solicitante (Professores, Alunos e Bolsistas).

***Motivos:** Necessidade de controle dos documentos que entram e saem da Faculdade; necessidade de saber o destino final de um documento que entrou ou saiu da Faculdade; e necessidade de comprovação do envio/recebimento do documento.

***Objetos:** Quadro de registro do protocolo e documento encaminhado.

***Ferramentas:** Caneta, carimbo e capacidade de escrita.

Pré-condições: Recebimento de um documento.

Descrição: A criação do protocolo inicia com a ação *gerar número do protocolo*, que consiste em *verificar número do último protocolo* no livro de protocolos, *adicionar um ao número do último protocolo*, e *preencher campo de "número do protocolo"(1)* do quadro de registro de protocolos. Após geração do número do protocolo a secretária deve *preencher a 1ª linha do quadro de registro de protocolos*, que consiste em *preencher os campos 2 a 9*. Tendo os campos preenchidos a secretária deve *encaminhar o documento* protocolado, o que inclui *carimbar documento a ser enviado, copiar número do protocolo no documento carimbado e enviar documento para o destinatário*.

Exceções: nenhuma.

Pós-Condições: Protocolo criado com sucesso, tendo todos os campos obrigatórios do quadro de registro de protocolos preenchidos corretamente.

***Divisão do Trabalho:** As secretárias são responsáveis pelo envio/recebimento dos documentos que entram e saem da Faculdade; os professores, bolsistas e alunos recebem e solicitam o envio dos documentos.

Figura 7.7 – Quadro descritivo do caso de uso “Criar Protocolo” com elementos complementares extraídos da atividade “Criar Protocolo”

7.1.2 Da Elicitação Baseada em Atividades ao Diagrama de Seqüência

O diagrama de seqüência é uma das ferramentas da UML para a modelagem de aspectos dinâmicos do sistema, oferecendo uma visão da interação entre objetos que cooperam para realização de um caso de uso (no nível da análise) ou de uma operação (no nível do projeto), com enfoque na troca de mensagens realizadas durante a interação [Boo99] [Eri98] [UML97b] [UML97d].

Partindo da idéia do mapeamento da atividade para caso de uso, podemos extrair vários elementos constitutivos de um diagrama de seqüência, quais sejam, objetos e mensagens, oriundos da decomposição de uma atividade. Os objetos do diagrama de seqüência podem ser extraídos dos sujeitos, objetos e comunidade da atividade, e as mensagens entre eles das ações e operações da atividade. As Figuras 7.8 e 7.9 apresentam diagramas de seqüências extraídos a partir das atividades “Consultar Protocolo por Data” e “Consultar Protocolo por Assunto”, respectivamente.

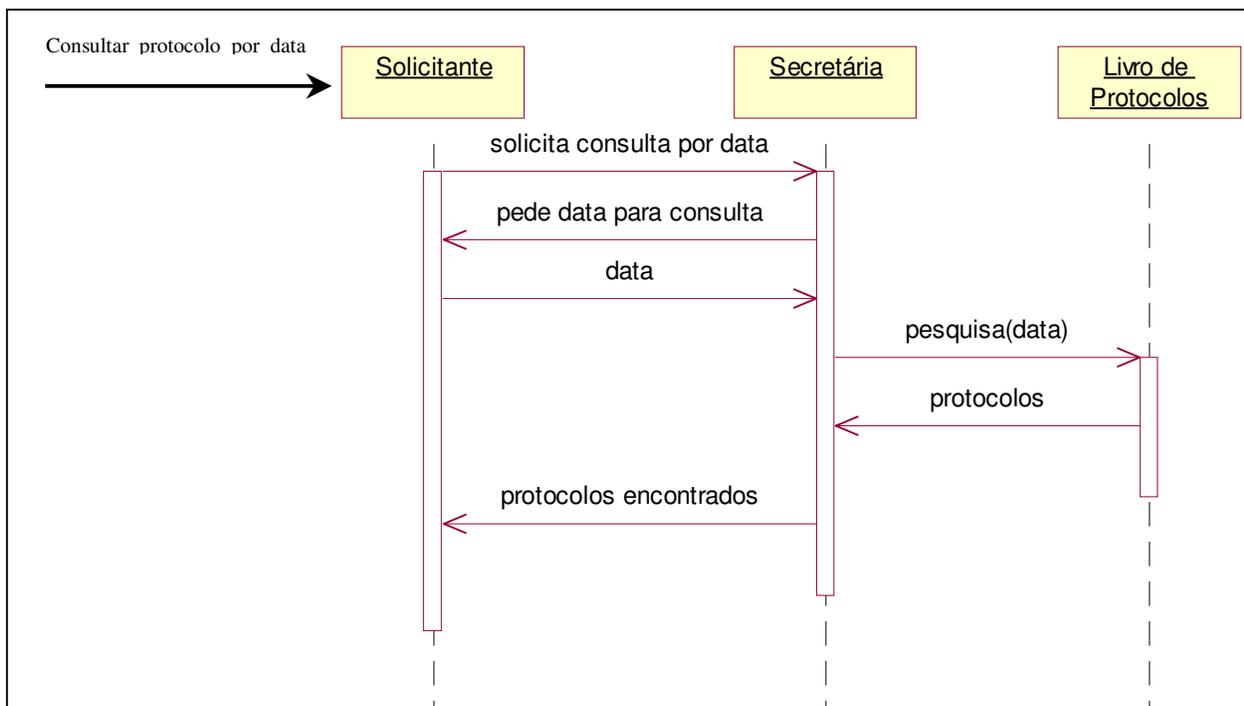


Figura 7.8 – Diagrama de seqüência extraído a partir da atividade “Consultar Protocolo por Data”

O objeto *solicitante* é uma generalização dos atores professor, aluno e bolsista, conforme apresentado na *Figura 7.1*. Conforme apresentado no estudo de caso, professores, alunos e bolsistas, mais a secretária (que também é sujeito das atividades), formam a comunidade onde ocorre a atividade “Criar Protocolo”. O livro de protocolo é objeto de transformação nas atividades, mapeado como objeto envolvido nas interações “Consultar Protocolo por Data” e “Consultar Protocolo por Assunto”.

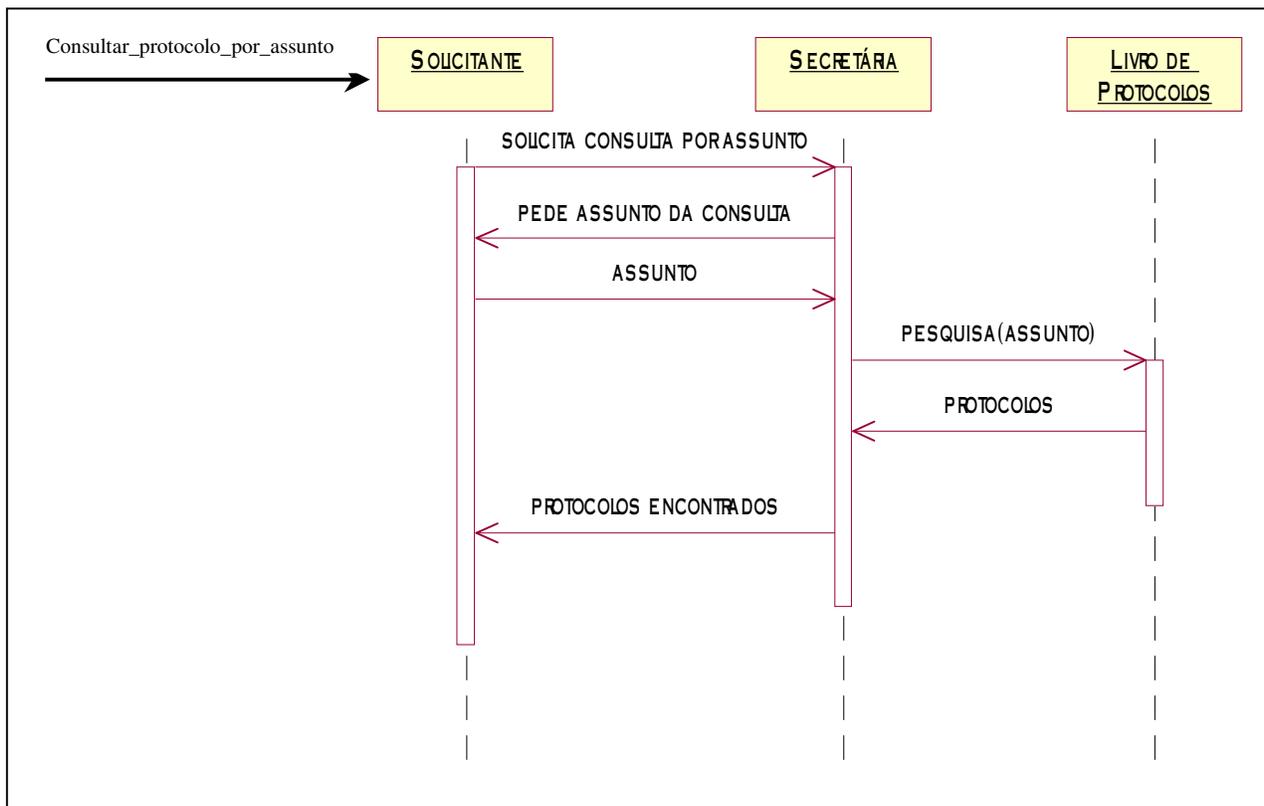


Figura 7.9 – Diagrama de seqüência extraído a partir da atividade “Consultar Protocolo por Assunto”

7.1.3 Da Elicitação Baseada em Atividades ao Diagrama de Atividades

O diagrama de atividades¹⁸ é outra ferramenta da UML para a modelagem de aspectos dinâmicos do sistema. Diferentemente do diagrama de seqüência, com enfoque na troca de mensagens entre

¹⁸ O diagrama de atividades é uma variação do diagrama de estados, destacando o fluxo de atividades existente enquanto um objeto permanece num determinado estado ou executa uma operação. Também pode ser usado para detalhar o fluxo de controle de um caso de uso.

os objetos do sistema, o diagrama de atividades oferece uma representação gráfica do fluxo de atividades existente durante a execução de um caso de uso ou de uma operação realizada por um objeto do sistema [Boo99] [Eri98] [UML97b] [UML97d]. Na modelagem de atividades preconizada pela UML, uma atividade é um estado de ação, cuja transição ocorre quando a atividade se encerra, não estando, portanto, vinculada à ocorrência de um evento externo.

As atividades representadas no diagrama de atividades têm correspondência direta com os conceitos de ações e operações no âmbito da Teoria da Atividade. Assim, também podemos extrair da atividade elementos constitutivos do diagrama de atividades. A *Figura 7.10* apresenta um diagrama de atividades para o caso de uso “Atualizar Quadro de Registro do Protocolo”, extraído a partir das ações e operações de sua atividade correspondente.

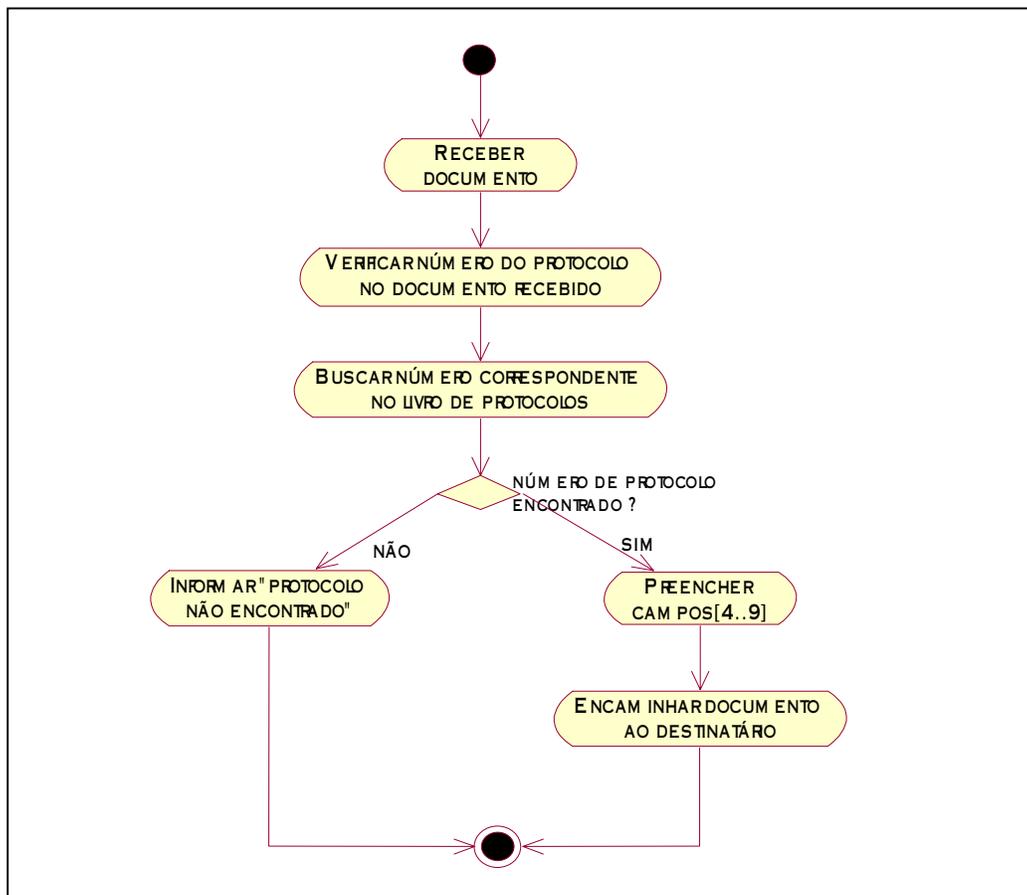


Figura 7.10 – Diagrama de atividades para “Atualizar Quadro de Registro do Protocolo”

Conclusão do Capítulo

Conforme o que foi apresentado neste capítulo, podemos perceber que a elicitação de requisitos organizada a partir do conceito de atividade pode ser facilmente “acoplada” às várias técnicas de análise e especificação de requisitos, oferecendo uma base a partir da qual o engenheiro de requisitos pode analisar e especificar os requisitos do sistema de forma mais confortável.

O mapeamento da atividade para outros conceitos que embutem funcionalidade do sistema se torna algo praticamente automático, e com a vantagem de levar consigo todos os elementos que formam o contexto da atividade, bem como sua estrutura hierárquica.

Conclusão

Conclusão

A elaboração deste trabalho se orientou na necessidade dos desenvolvedores de software e de seus usuários em superar as dificuldades essenciais que estão inerentes ao processo de elicitação de requisitos. Na medida em que envidamos esforços em direção à superação de tais dificuldades, culminando em resultados úteis que possam ser aplicados para a melhoria da qualidade do processo de elicitação, entendemos estar contribuindo para que a Engenharia de Requisitos possa se tornar uma engenharia “de fato”. Conseqüentemente, nos aproximamos de uma realidade onde as dificuldades essenciais da elicitação de requisitos (e da Engenharia de Requisitos como um todo) possam ser enfrentadas de maneira mais adequada.

Para chegarmos à proposta da metodologia de elicitação de requisitos apresentada neste trabalho, realizamos inicialmente um estudo sobre a Engenharia de Requisitos e uma pesquisa sobre as principais técnicas que vêm sendo empregadas na elicitação de requisitos. Esse estudo ofereceu um panorama dos problemas enfrentados no processo de Engenharia de Requisitos, com ênfase nas dificuldades relativas ao processo de elicitação. Outra etapa fundamental para a elaboração da metodologia proposta, foi o estudo da Teoria da Atividade, trazendo como tema central de discussão o conceito de atividade humana, que é rico e complexo, envolvendo uma gama de elementos, como sujeito, ferramentas de mediação, objetos de transformação, comunidade, regras e divisão do trabalho, e que ainda se apresenta como uma estrutura hierárquica onde a atividade subdivide-se em ações e operações. Pois é este conceito de atividade que é adotado na metodologia proposta, onde utilizamos a atividade como uma unidade organizativa do processo de elicitação de requisitos.

No estudo de caso apresentado, buscamos demonstrar a viabilidade do uso da metodologia proposta em processos de elicitação de requisitos. A quantidade e a relevância das informações obtidas sobre o sistema em questão (sistema de protocolos), organizadas em torno do conceito de atividade e decompostas em termos dos elementos constitutivos da atividade, nos ofereceu um exemplo de um processo de elicitação de requisitos promissor.

Organizar o processo de elicitação de requisitos em torno do conceito de atividade (oriundo da Teoria da Atividade) é em um novo paradigma de elicitação, o qual entendemos ser rico na definição de elementos orientativos durante a elicitação de requisitos, e que também vão auxiliar nas etapas seguintes de análise e especificação de requisitos. Conforme já apontado por vários pesquisadores [Cas00][Bre99][Ber98][Axt97][Car97][Gog97][DeM91], as dificuldades da elicitação de requisitos não podem ser superadas de uma “forma puramente tecnológica”, uma vez que o contexto social é mais crucial durante a elicitação (e na Engenharia de Requisitos, de uma forma geral) do que nas fases de projeto e implementação do software. As necessidades dos clientes e usuários de um sistema de software, que se constituem na questão central da elicitação de requisitos [Tha97], estão intimamente relacionadas com os desejos e motivos que levam à realização de uma atividade.

Desejos e motivos, que buscam ser contemplados pela realização de uma atividade, bem como as ações e operações que dão “vida” à atividade que se põe em curso, podem ser melhor compreendidos quando temos um contexto de análise bem definido para eles: este contexto é fornecido pela atividade.

Como sabemos, a elicitação de requisitos é realizada dentro do contexto da Engenharia de Requisitos, que se desdobra em outras atividades além da elicitação, como análise, especificação e validação de requisitos. Dessa forma, entendemos que uma metodologia de elicitação de requisitos deva ser adaptável ao contexto maior, ou seja, ao da Engenharia de Requisitos, onde os resultados do processo de elicitação servirão como material de entrada, por exemplo, para os processos de análise e especificação de requisitos.

Esta adaptabilidade ao processo da Engenharia de Requisitos foi demonstrada com os exemplos de transição fornecidos no Capítulo 7, onde apresentamos algumas formas de transição entre os resultados obtidos com o uso da metodologia de elicitação proposta e algumas técnicas de análise e especificação de requisitos (modelagem de casos de uso, diagramas de seqüências e diagramas de atividades, empregadas no âmbito da UML/USDP). Nesses exemplos pudemos perceber que um mapeamento da atividade para outras técnicas que trabalham com conceitos que embutem

funcionalidade do sistema (como o caso de uso, ou a operação de uma classe) se torna praticamente automático, e com a vantagem de levar consigo todos os elementos que formam o contexto da atividade, bem como sua estrutura hierárquica, contribuindo para análise dos requisitos que deve ocorrer de forma mais detalhada posteriormente.

É importante destacar que, embora o foco deste trabalho tenha sido a utilização dos conceitos da Teoria da Atividade no processo de elicitação de requisitos, entendemos que a riqueza de conceitos da Teoria da Atividade pode se estender para uso nas demais fases da Engenharia de Software.

Como trabalhos futuros está prevista uma formalização da metodologia de elicitação de requisitos proposta, utilizando heurísticas que orientem de forma mais efetiva a execução das etapas da metodologia.

Também como trabalhos futuros pretendemos desenvolver a formalização de guias que automatizem o mapeamento da atividade, e seus elementos constitutivos, para elementos conceituais modelados em técnicas de especificação de requisitos, como em diagramas de casos de uso, diagramas de seqüências e diagramas de atividades da UML, modelo de ciclo de vida e modelo de operações do método *Fusion*, entre outros. Outro possível desdobramento deste trabalho é a implementação de um ambiente computacional que suporte o processo de elicitação de requisitos preconizado na metodologia proposta.

Referências Bibliográficas

- [Ack92] Ackroyd, S., Harper, R. et al., "Information Technology and Practical Police Work", Open University Press, 1992.
- [Ama94] Amako, K., "The Fusion Method - Summary Note", Version: Dec. 26, 1994, <http://arkhp1.kek.jp/managers/./fusionBook/fusionBookContents.html>.
- [Amb00] Ambler, S. W. and Constantine, L. L., "The Requirements Workflow" in The Unified Process: Inception Phase, CMP Books, 2000, pp. 13-14.
- [Axt97] Axtell, C. M., Waterson, P. E. and Clegg, C. W., "Problems Integrating User Participation into Software Development", Int. J. Human-Computer Studies, Vol. 47, August/1997, pp. 323-344.
- [Ber98] Berry, D. M. and Lawrence, B., "Requirements Engineering – Guest Editors' Introduction", IEEE Software, March/April 1998, pp 26-29.
- [Boe81] Boehm, B., "Software Engineering Economics", Prentice-Hall, 1981.
- [Boo94] Booch, G., "Object-Oriented Analysis and Design with Application", Benjamin/Cummings, 1994.
- [Boo99] Booch, G., Rumbaugh, J. and Jacobson, I., "The Unified Modeling Language User Guide", Addison Wesley, 1999.
- [Bre98] Breitman, K. K. e Leite, J. C. S. P., "Suporte Automatizado à Gerência da Evolução de Cenários", Workshop de Engenharia de Requisitos - WER98, 1998.

- [Bre99]** Breitman, K. K., Leite, J. C. S. P. and Finkelstein, A., “The World’s a Stage: A Survey on Requirements Engineering using a Real-Life Case Study”, Journal of the Brazilian Computer Society, N. 1, Vol. 6, July/1999, pp. 13-37.
- [Bri95]** Brinkkemper, S. *et al.*, “Object-Oriented Analysis and Design Methods: A Comparative Review”, University of Twente, January/1995, <http://wwwis.cs.utwente.nl:8080/dmrg/OODOC/oodoc/oo.html>
- [Bro87]** Brooks, F., "No Silver Bullet: Essence and Accidents of Software Engineering", Computer, Apr. 1987, pp. 10-19.
- [Bur99]** Burd, L., "Desenvolvimento de Software para Atividades Educacionais", Dissertação de Mestrado apresentada na Faculdade de Engenharia Elétrica e de Computação, Unicamp, 1999.
- [Cas00]** Castro, J., Alencar, F. e Cysneiros, G., “Closing the GAP Between Organizational Requirements and Object Oriented Modeling”, Journal of the Brazilian Computer Society, N. 1, Vol. 7, July 2000, pp. 5-16.
- [Car97]** Carrol, J. M., “Human-Computer Interaction: Psychology as a Science of Design”, Int. J. Human-Computer Studies, Vol. 46, April/1997, pp. 501-522.
- [Chi95]** Chin, K. F., "A JAD Experience", Proceedings of the 1995 ACM SIGCPR Conference on Supporting Teams, Groups, and Learning Inside and Outside the IS Function Reinventing IS , 1995.
- [Coc00]** Cockburn, A., “Writing Effective Use Cases”, Addison-Wesley, October/2000.

- [Col93] Cole, M. and Engeström, T., “A Cultural-Historical Approach to Distributed Cognition”, In G. Solomon, ed., *Distributed Cognition* (pp. 1-47), Cambridge: Cambridge University Press, 1993.
- [Col94] Coleman, D. *et al.*, “Object-Oriented Development: The Fusion Method”, Prentice Hall, 1994.
- [Dam97] Damian, A. *et al.* “Joint Application Development and Participatory Design”, 1997. <<http://www.cpsc.ucalgary.ca/~pand/seng/613/report.html>>
- [Dav97] Davis *et al.*, “Identifying and Measuring Quality in a Software Requirements Specification”, in *Software Requirements Engineering*, 2nd Ed., IEEE CS Press, 1997, pp 164-175.
- [DeC93] DeChampeaux, D., "Object Oriented System Development", Addison-Wesley, 1993.
- [DeM91] De Marco, T., “Non-Technological Issues in Software Engineering”, *Proceedings of the 13th IEEE International Conference on Software Engineering (ICSE)*, 1991.
- [Dix93] Dix, A. *et al.*, “Human-Computer Interaction”, Prentice Hall, 1993.
- [Dor97] Dorfman, M., "Requirements Engineering", in *Software Requirements Engineering*, 2nd Ed., IEEE CS Press, 1997, pp 7-22.
- [Ell98] Ellison M. and McGrath G. M., “Business Process Modelling Using Activity Theory: An Approach to Data Capture and Analysis”, *Australian Computer Journal*, (30,4), 1998, pp. 146-152.

- [Eng87]** Engeström, Y., "Learning by Expanding", Helsinki: Orienta-Konsultit, 1987.
- [Eri84]** Ericsson, K. A. and Simon, H. A., "Protocols Analysis: Verbal Reports as Data", MIT, 1984.
- [Eri98]** Ericsson, H. and Penker, M. "UML Toolkit", Wiley Computer Publishing, 1998.
- [Fau97]** Faulk, S. R., "Software Requirements: A Tutorial", in Software Requirements Engineering, 2nd. Ed., IEEE CS Press, 1997, pp 128-149.
- [Fou94]** Fournier, R., "Guia Prático para Desenvolvimento e Manutenção de Sistemas Estruturados", Makron Books, 1994.
- [GAO92]** US General Accounting Office, "Mission Critical Systems: Defense Attempting to address Major Software Challenges", GAO/IMTEC-93-13, December 1992.
- [Gog97]** Goguen, J. A. and C. Linde, "Techniques for Requirements Elicitation", in Software Requirements Engineering, 2nd. Ed., IEEE CS Press, 1997, pp 110-122.
- [Har91]** Harper, R., Hughes, J. et al., "Harmonious Working and CSCW: Computer Technology and Air Traffic Control" in Studies in Computer-Supported Cooperative Work, 1991, pp. 225-234.
- [Hug95]** Hughes, J., O'Brien, J. et al., "Presenting Ethnography in the Requirements Process", Proceedings RE'95, IEEE Computer Society Press, 1995.

- [Jac92]** Jacobson, I. *et al.*, “Object-Oriented Software Engineering”, Addison-Wesley, 1992.
- [Jac98]** Jacobson, I., Booch, G. and Rumbaugh, J., "The Unified Software Development", Object Technology Series, Addison-Wesley, 1998.
- [Joh92]** Johnson, P., “Human-Computer Interaction - Psychology, Task Analysis and Software Engineering”, McGraw-Hill, 1992.
- [Joh96]** Johnson, D. M., “The Systems Engineer and the Software Crisis”, ACM SIGSOFT, Software Engineering Notes, Vol. 21, N. 2, March/1996, pp. 64-73.
- [Kap96]** Kaptelinin, V., “Activity Theory: Implications for Human-Computer Interaction” in Context and Consciousness - Activity Theory and Human-Computer Interaction, MIT Press, 1996, pp. 104-116.
- [Kap97]** Kaptelinin, V. and Nardi, B. A., “Activity Theory: Basic Concepts and Applications”, CHI 97 Electronics Publications: Tutorials, march/1997, <http://www.cwi.nl/~steven/chi97/proceedings/tutorial/bn.html>.
- [Kap98]** Kaptelinin, V. and Nardi, B. A., “The Activity Checklist: A Tool for Representing the ‘Space’ of Context”, submitted to CHI 98.
- [Kot98]** Kotonya, G. and Sommerville, I., “Requirements Engineering: Processes and Techniques”, John Wiley and Sons, 1998.
- [Kul00]** Kulak, D. and Guiney, E., “Use Cases – Requirements in Context”, Addison-Wesley, 2000.

- [Kuu96]** Kuuti, K., “Activity Theory as a Potential Framework for Human-Computer Interaction” in Context and Consciousness - Activity Theory and Human-Computer Interaction, MIT Press, 1996, pp. 17-44.
- [Lei97]** Leite, J.C.S.P. *et al.*, "Enhancing a Requirements Baseline with Scenarios", Requirements Engineering Journal: Springer-Verlag London Limited Vol. 2, N. 4, 1997, pp 184-198.
- [Leo78]** Leontiev, A. N., “O Desenvolvimento do Psiquismo”, Livros Horizonte, 1978.
- [Leo81]** Leont`ev, A. N., “The Problem of Activity in Psychology”, in The Concept of Activity in Soviet Psychology, , M. E. Sharp, 1981. pp. 37-71
- [Luf94]** Luff, P., Jirotko, M., Heath, C. and Greatbatch, D., “Tasks and Social Interaction: the Relevance of Naturalistic Analyses of Conduct for Requirements Engineering”, Proceedings of the First IEEE International Symposium on Requirements Engineering, 1994.
- [Mar99a]** Martins, L. E. G. and Daltrini, B. M., "An Approach to Software Requirements Elicitation Using Precepts from Activity Theory", Proceedings of the 14th IEEE International Conference on Automated Software Engineering, 1999.
- [Mar99b]** Martins, L. E. G. and Daltrini, B. M., “Utilização dos Preceitos da Teoria da Atividade na Elicitação dos Requisitos do Software” Anais do XIII Simpósio Brasileiro de Engenharia de Software, SBC, 1999.
- [Mar99c]** Martins, L. E. G. and Daltrini, B. M., “Activity Theory: a Framework to

Software Requirements Elicitation”, WER’99 - Workshop en Requerimentos, 28 JAIO – Jornadas Argentinas de Informática e Investigación Operativa, SADIO – IFIP, 1999.

[McG00] McGrath, G. M. and Uden, L., “Modelling ‘Softer’ Aspects of the Software Development Process: An Activity Theory Based Approach”, Proceedings of the 33rd Hawaii International Conference on System Sciences, IEEE, 2000.

[Min86] Minick, N., “The Early History of the Vygotskian School: The Relationship between Mind and Activity”, The Quarterly Newsletter of the Laboratory of Comparative Human Cognition, Vol. 8, N. 4, October/1986, pp. 119-125.

[Nar96] Nardi, B. A., “Studing Context” in Context and Consciousness - Activity Theory and Human-Computer Interaction, MIT Press, 1996, pp. 69-102.

[Net00] Neto, J. M. S., Leite, J. C. S. P. e Cysneiros, L. M., “Non-Functional Requirements for Object-Oriented Modeling”, Anais do III Workshop de Engenharia de Requisitos, 2000.

[Par86] Parnas, D. and Clements, P., “A Rational Design Process: How and Why to Fake It”, IEEE Trans. Software Engineering, Vol. 12, No. 2, Feb. 1986, pp. 251-257.

[Pre95] Pressman, R. S., “Engenharia de Software”, Makron Books, 1995.

[Rum91] Rumbaugh, J. *et al.*, “Object-Oriented Modeling and Design”, Prentice Hall, 1991.

[Rum97] Rumbaugh, J., “Getting Started: Using Use Cases to Capture

Requirements”, In: Software Requirements Engineering, IEEE-CS Press, 1997.

- [Sil94]** Silva, A. E. A., “The AIM Methodology Analysis, Interpretation and Modelling: A Methodology For The Role Of The Knowledge Analyst In Knowledge Acquisition”, Master Thesis in Computer Science at Massey University, February/1994.
- [Sut98]** Sutcliffe, A. et al., “*Supporting Scenario-Based Requirements Engineering*”, IEEE Transactions on Software Engineering, Vol. 24, No. 12, December/1998.
- [Tha97]** Thayer, R. H. and Dorfman, M., "Introduction to Tutorial Software Requirements Engineering", in Software Requirements Engineering, 2nd Ed., IEEE CS Press, 1997, pp 1-2.
- [UML97a]** Rational Software et al., “UML Summary”, Version 1.1, September/1997.
- [UML97b]** Rational Software et al., “UML Semantics”, Version 1.1, September/1997.
- [UML97c]** Rational Software et al., “UML Extension for Business Modeling”, Version 1.1, September/1997.
- [UML97d]** Rational Software et al., “UML Notation Guide”, version 1.1, September/1997.
- [Vyg81]** Vygostky, L. S., “The Instrumental Method in Psychology”, in The Concept of Activity in Soviet Psychology, M. E. Sharp, 1981. pp. 134-146
- [Vyg94]** Vygostky, L. S., “A Formação Social da Mente: O Desenvolvimento dos

Processos Psicológicos Superiores”, 5^a edição, Martins Fontes, 1994.

- [Wer81]** Werstch, J. V., “The Concept of Activity in Soviet Psychology: An Introduction”, in The Concept of Activity in Soviet Psychology, M. E. Sharp, 1981. pp. 03-36
- [Wil90]** Wilkerson, B. and Wiener, L., “Designing Object-Oriented Software” Prentice Hall, 1990.

Apêndice A

Artigo publicado nos anais da *14th IEEE International Conference on Automated Software Engineering (ASE'99)* realizada em *Cocoa Beach, Florida-USA*, em outubro de 1999.

An Approach to Software Requirements Elicitation Using Precepts from Activity Theory

Luiz Eduardo Galvão Martins
martins@cce.unimep.br
Methodist University of Piracicaba - Unimep

Beatriz Mascia Daltrini
beatriz@dca.fee.unicamp.br
State University of Campinas - Unicamp

Abstract

The goal of this work is to show that precepts from Activity Theory can be used in software requirements elicitation. Thus, we propose an approach to requirements elicitation using principles from this theory. Initially, we mention common problems found in requirements elicitation. Then we explain the main precepts of Activity Theory. Finally, we present a case study using the suggested approach to requirements elicitation.

1. Introduction

One fundamental question in Requirements Engineering is how to determine real user needs. Researches have proved that many software projects have failed because of the problems in software requirements elicitation [1] [7]. Specifically, the requirements that are obtained are often incomplete, misunderstood, and ambiguous.

The correct identification of software requirements is not an easy task because of the abstract nature of software. An approach that can be used to obtain a better understanding of the problems found in requirements elicitation is to divide the problems into two categories [4] [6]: accidental problems and essential problems. The accidental problems emerge because of poor control over the activities developed in requirements engineering: low effort in the requirements elicitation with the user, poor documentation about the requirements, poor revision of the requirements, incorrect specification of the requirements, and tendency to initiate prematurely the software development process. The essential problems are embedded in

requirements elicitation: difficulty of the user to know exactly what he wants, difficulty of the communication between user and developer, and the changing nature of the requirements.

The accidental problems can be considered less difficult to overcome. The adoption of a systematic process that orient the elicitation, analysis, specification, validation and management of the requirements tend to solve, or at least minimize, the problems of that category.

Nevertheless, the essential problems are more difficult to be overcome, once they are contained in the requirements nature. The adoption of a systematic process to requirements engineering, mainly to specification, validation and management of the requirements, will also aid in overcoming the essential problems. However, the problems that naturally exist in the human comprehension and communication process, which is in the nucleus of the requirements elicitation, will need an approach which takes into account the context in which the persons develop their activities and recognize the objects needed to develop them, the historic of evolution of these activities and mediation tools, and others aspects of psychological and social relevancy that affect the users of the software to be developed.

Thus the essential problems of requirements elicitation will not be solved in a purely technological approach when social aspects have strong importance in the activity [8]. The majority of software is developed with no one help from the social sciences (such as psychology, sociology, anthropology etc.), not approaching in a systematic way the users' necessities, both at individual and organizational levels. We argue in this article that the utilization of some precepts from

Activity Theory, coming from Soviet psychology, can bring important benefits to the software elicitation

2. Activity Theory

Activity Theory can be defined as an interdisciplinary and philosophical framework for studying the different human practices of the development process, both at the individual and social levels. Activity Theory has three historic roots: the Germany classic philosophy from the 18th and 19th century (from Kant and Hegel); the Marx and Engels's manuscripts, which worked on the concept of activity; and the Soviet psychology, funded by Vygotsky, Leont'ev and Luria. The term "Activity Theory" emerged between 1920 and 1930, in the Soviet Historic-Cultural School of Psychology.

2.1 Basic Principles of Activity Theory

Activity Theory is formed by a set of principles that constitutes a general conceptual system [10] [12]. The basic principles of Activity Theory are as follows:

- (1) *Principle of the unit between activity and consciousness.* It is considered the fundamental principle of Activity Theory, that activity and consciousness are treated in an integrated way. Consciousness means the human mind as a whole, and activity means the human interaction with its objective reality. This principle states that the human mind emerges and exists like a special component of the human interaction with its environment. The mind is a special organ that appears in the evolution process to help organisms to survive. Thus, it can be analyzed and understood only within the human activity context.
- (2) *Principle of object orientation.* This principle focuses on the approach of Activity Theory for the environment where the human being interacts. Human beings live in an environment that is very important for them. This environment consists of entities that combine all kinds of objective features, including those culturally determined, which influence the ways individuals interact with those entities.
- (3) *Principle of the hierarchical structure of the activity.* Activity Theory differentiates the human procedures at several levels (activity, action and operation), taking into account the objectives to which these procedures are oriented. The importance of that distinction is determined by the ecological attitude of Activity Theory. In a real

process.

situation, this distinction is frequently necessary to preview human behavior. This distinction is very important for differentiating among motives, goals, and conditions, which are associated with activities, actions, and operations, respectively.

- (4) *Principle of internalization-externalization.* This principle describes the basic mechanisms about the source of mental processes. It states that mental processes are derived from external actions by way of internalization. Internalization is the information absorption process (in several ways) achieved by the human mind, which derives from the contact with the environment where the person is located. Externalization is the inverse process to internalization, manifested through acts, in such a way they can be verified and fixed, if necessary.
- (5) *Principle of mediation.* Human activity is mediated by several tools, both externals (e.g.: an axe, a computer etc.) and internals (e.g.: an heuristic, a concept etc.). The tools are "vehicles" of the social experience and cultural knowledge.
- (6) *Principle of development.* According to Activity Theory, understanding a phenomenon means knowing how it has developed by itself until reaching its current shape, since it is changed by time. Understanding these changes can help in understanding its current state.

These principles are not isolated ideas, but are closely connected. The nature of Activity Theory is manifested in that set of principles.

2.2. The concept of Activity

At the individual level, an activity has three elements: subject, object and mediation tool. The subject is the agent who acts upon the object of the activity. The object is the element to which the actions will be directed. An object may be something material, or something at least tangible, as for example, a plan or an idea.

The reciprocal relationship between the subject and the object is always mediated by one or more tools (also called mediation artifacts), that can be instruments, signs, procedures, machines, methods, laws, ways for organization of work etc. Tools always have a role in the mediation process and are used in the process of the object transformation [12].

Figure 1 represents the relationship structure, in the individual level, between the subject and the object in the context of an activity, where the tool assumes an

important role in the mediation among them. Through this mediation, some result is obtained. Transforming an object to a result motivates the existence of an activity.

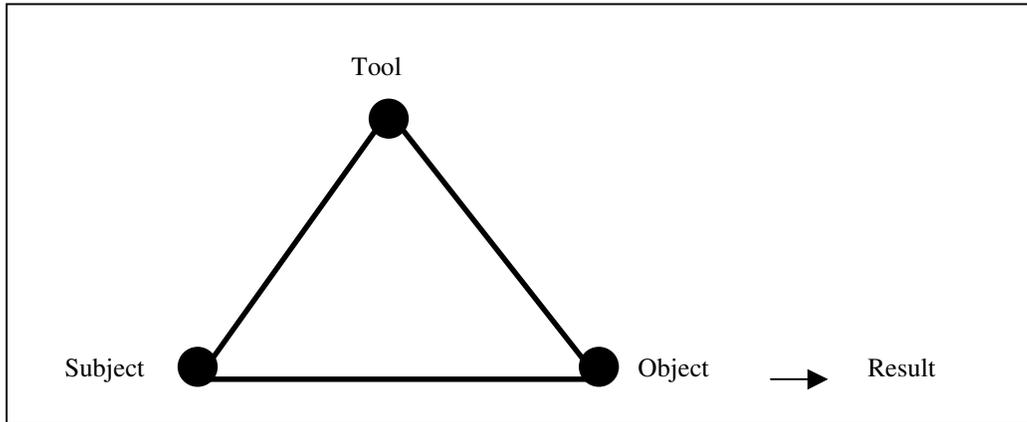


Figure 1 – Relationship mediated between the subject and the object in the individual level [5] [12].

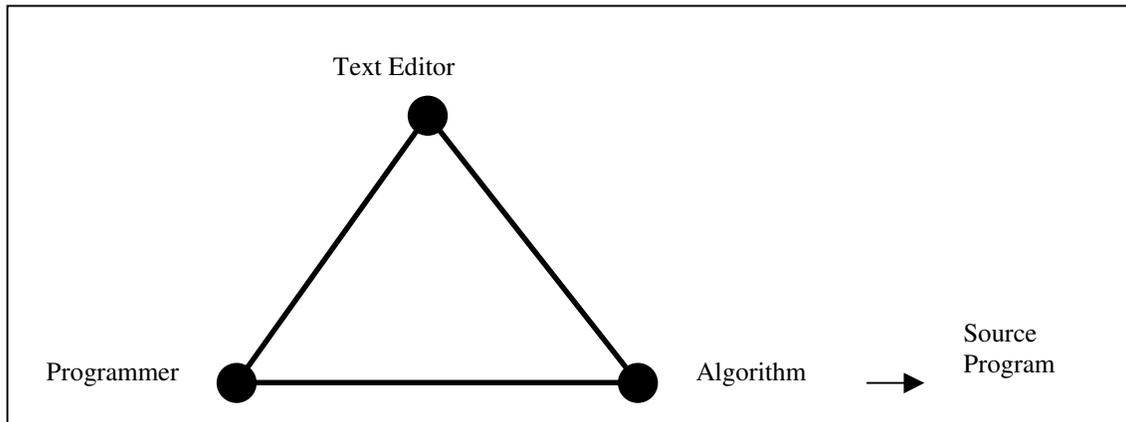


Figure 2 – Structure of the activity "Write a program".

In order to exemplify the figure shown above, consider the following activity: "Write a program". In this case, the subject of the activity would be a programmer; the mediation tool would be a text editor; the object to be transformed would be an algorithm and the result would be the source program ready for compilation (see figure 2).

Although the representation of the relationship mediated by the subject and the object in the individual level is useful, this structure is too simple to represent the considerations of the existing systemic relations between the subject and its environment, once these relations are found in a lot of activities.

So, a new element should be added to the structure of the activity: the community. The community is formed by all the subjects which share the same object.

When the concept of community is presented, new ways of mediation arise (besides that possibility through the tools). These new ways of mediation are called rules and division of labor (see figure 3).

Rules as a form of mediation between the subject and the community are implicit or explicit norms established by conventions and social relations in the society. The division of labor is a form of mediation between the community and the object, refers to a form of organizing a community, related to the process of transforming an object into a result.

All the mediation forms (tools, rules and division of labor) have a historical development of their own, with particular characteristics related to the context in which they were developed.

According to Activity Theory, activities are not static, they evolve, normally in a non-linear way. Each activity has its own history, embedding past phases. One

historical analysis of its development is often needed to understanding the current situation.

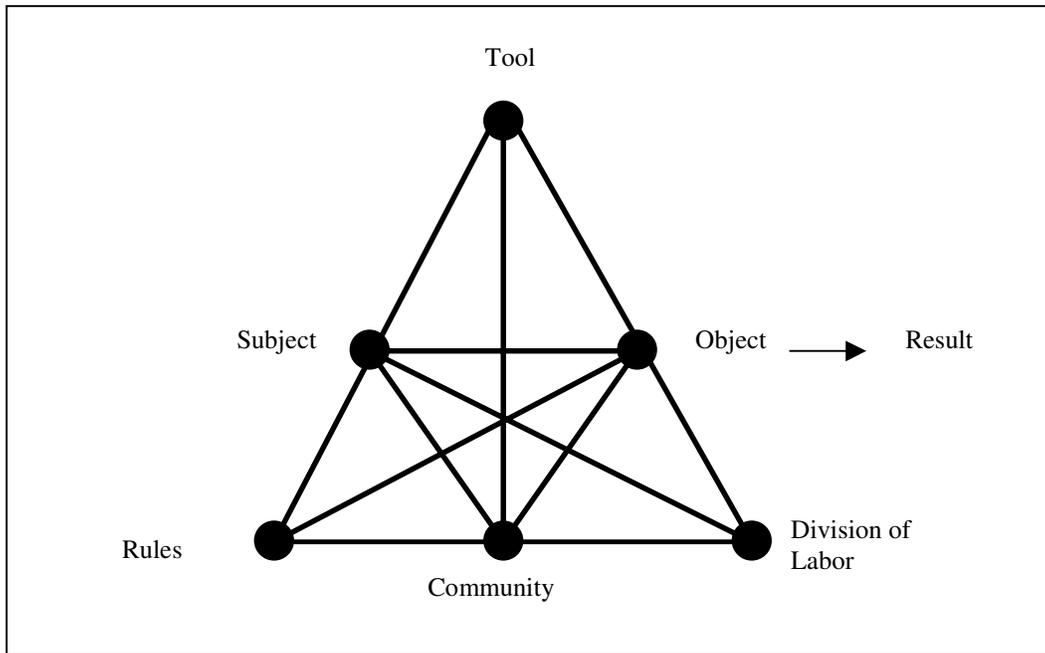


Figure 3 – Systemic Model of an activity [5] [12].

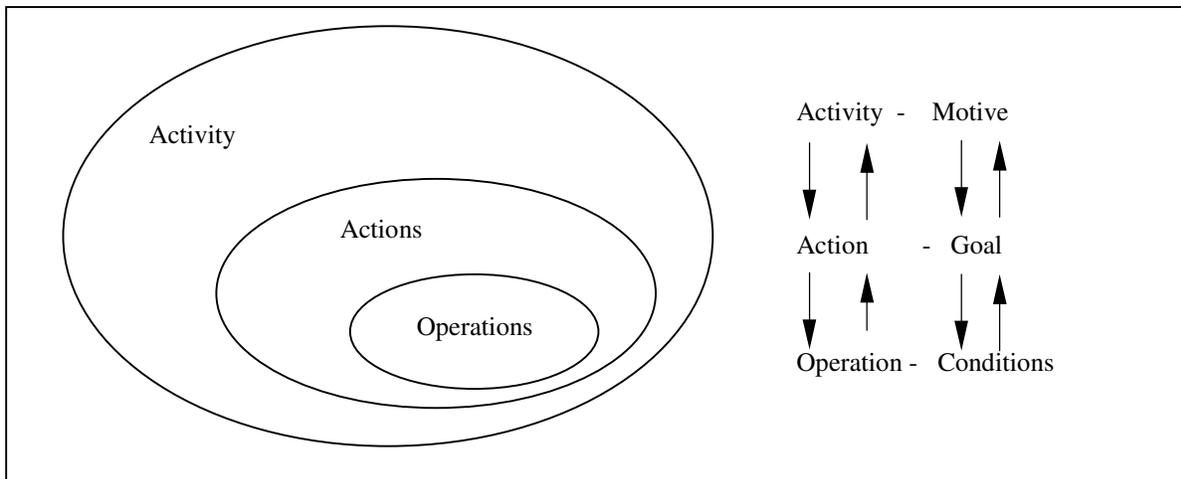


Figure 4 – Hierarchical levels of an activity.

2.3 Levels of an Activity

An activity is decomposed into actions, and each action is decomposed into operations (see figure 4). Activities are long way formations, their objects are

transformed into results not only once, but through a process of several phases or steps.

Thus, an activity, created in a given moment, passed through a process of evolution, in which actions and operations have been created, eliminated, and

transformed in order for the activity to reach its current “format” [12].

Although an activity is oriented by a motive, the actions are oriented to goals, and the operations oriented to conditions. One activity is performed through cooperative or individual actions and chains or nets of actions that are related one to another for aiming the same goal.

One important characteristic of an action is that it is planned before its effective execution. On the other

hand, an operation is executed in an automatic way, without previous planning. Only an analysis of the current conditions leads to its execution. The planning of an action is done in a conscious way, using some mental model. The better the model, the more success the action will reach. This planning to the execution of an action is called orientation.

Activity	Actions	Operations
Process Sale	Emit the Sale Bill	Fill the fields of the sale bill
		Compute the taxes
		Print the sale bill
	Emit the Sale Payments	Divide the sale into several payment receipts according to their due dates.
		Print the payment receipts

Table 1 – The activity “Process Sale”.

When an action is performed several times and reaches a level of maturity enough to be executed automatically, that is, without previous planning, then it reaches the level of operation. In this way, one operation that was an action became common in the context of an activity.

In Table 1, an example of the decomposition of an activity is shown. The activity “Process Sale ” was decomposed into actions “Emit the Sale Bill” and “Emit the Sale Payments”.

3. An Approach for Requirements Elicitation from Activity Theory

Some psychological approaches use human action as the basic unit of analysis of real life situations. This approach can offer good results as the action in question is analyzed in an isolated way, e.g.: situations designed to laboratory experiments.

However, in real situations, the human actions should be analyzed into a specific context, necessary in order to make understandable actions. According to Activity Theory, a minimum context is given when human action is analyzed inside an activity (according to the concept of activity mentioned in section 2.2). Hence the concept of activity is then presented as the basic unit of analysis of situations.

In order to obtain software requirements in an adequate way, it is necessary to understand, among other things, the activities performed by the agents involved in the system, which will implement the future software.

3.1 A Case Study

The following example involves the construction of software to control the protocols of a secretary’s office of an university. The following problem declaration was obtained from a real situation: an open interview that was performed with the secretary [8].

Initial Declaration of the Problem

The system of protocols consists in controlling documents in and out of the secretary’s office. Considering any document that goes in and out of the office, it is generated a number for the protocol and the fields from the registration board of protocols are filled (figure 5).

The fields presented in the registration board of protocols mean:

1. Number of the protocol
2. Name of the person who submitted the protocol (local/department)
3. Area of Interest
4. From (local source/ person who signed)
5. Arriving date
6. Person who received (name of the person in charge)
7. Destination (the person to whom the protocol is addressed)
8. Leaving Date (date of the submission of the protocol to the receiver)
9. Receiver Signature (or some kind of information that can identify that the receiver got the protocol).

The information is annotated in a way to maintain registrations about the documents which circulate by the secretary’s office.

Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade

Nowadays, this type of control is done without the use of computers. Protocols are registered in a book, where each page can have until 10 registers of protocols. All pages have a number.

(1) Protocol n° _____	(4)From	(5)Date	(6)Rec	(7)To	(8)Date	(9)Rec
(2) Name: _____						
(3) Area: _____						

Figure 5 - Registration Board of Protocols.

3.2 The Approach for the Requirements Elicitation

The approach that has been adopted for requirements elicitation of the case study presented consists of the following steps:

1. Identify procedures performed in the system, which can be classified as activities.
2. Identify for each activity: subject, tool, object, community, rules, division of labor and results (representation of the systemic model of activity).
3. From the systemic model of activity, decompose the activities into actions and operations.

1. Procedures Classified as Activities

In order to identify procedures (or processes) that can be classified as activities the principles (1), (2), and

(3) from Activity Theory (mentioned in section 2.1) will be utilized. Thus, for example, the following procedures can be classified as activities:

- Create a protocol
- Update a protocol
- Consult a protocol by its date

2. Systemic Models

After the identification of the activities, the systemic models of the activities can be developed. Through the systemic models, the elements that compose the activities “create a protocol”, “update a protocol” and “consult a protocol by its date” should be obtained, according to the figures 6, 7, and 8 respectively. Those models take into account the principles (2) and (5) from Activity Theory.

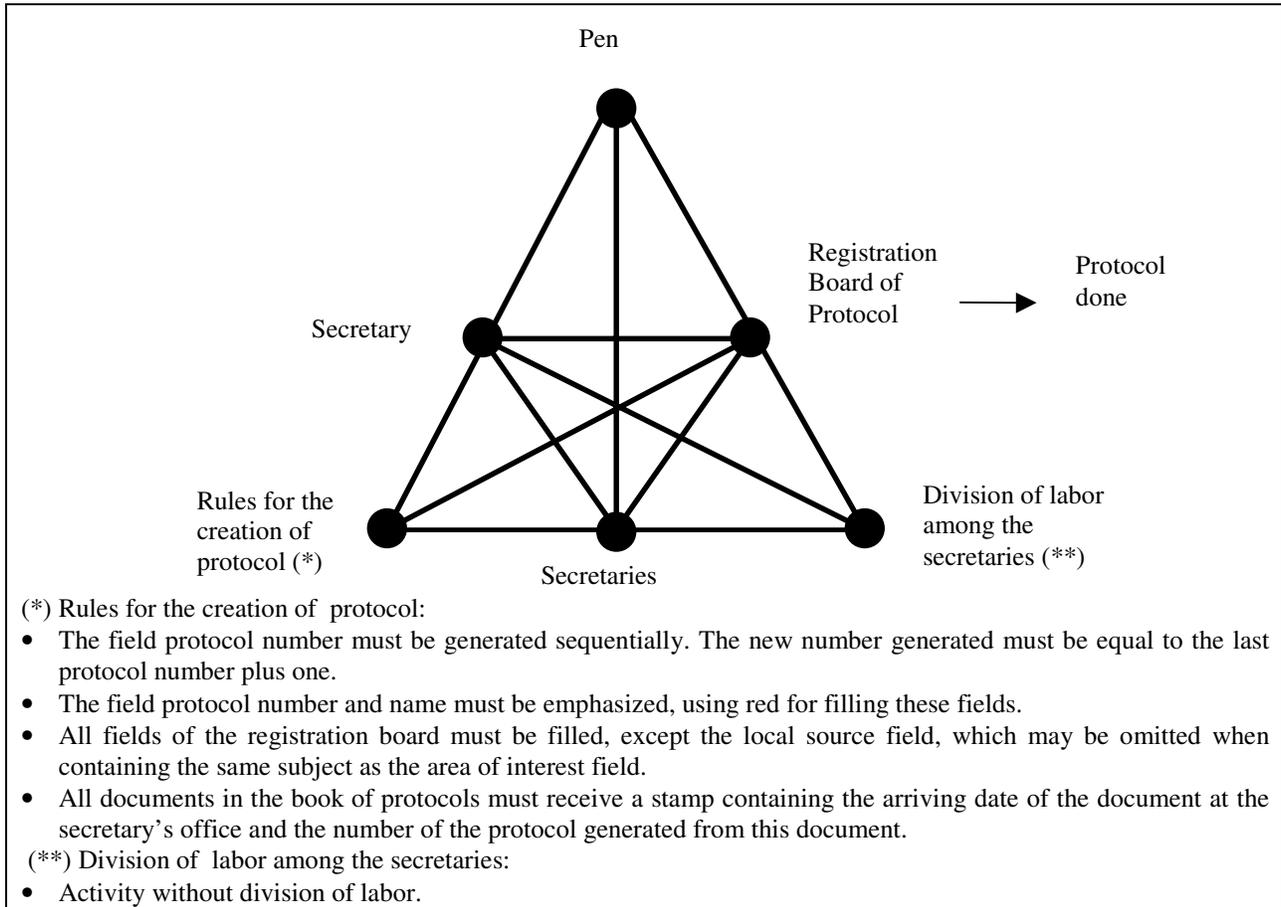


Figure 6 – Systemic Model of the activity "Create Protocol".

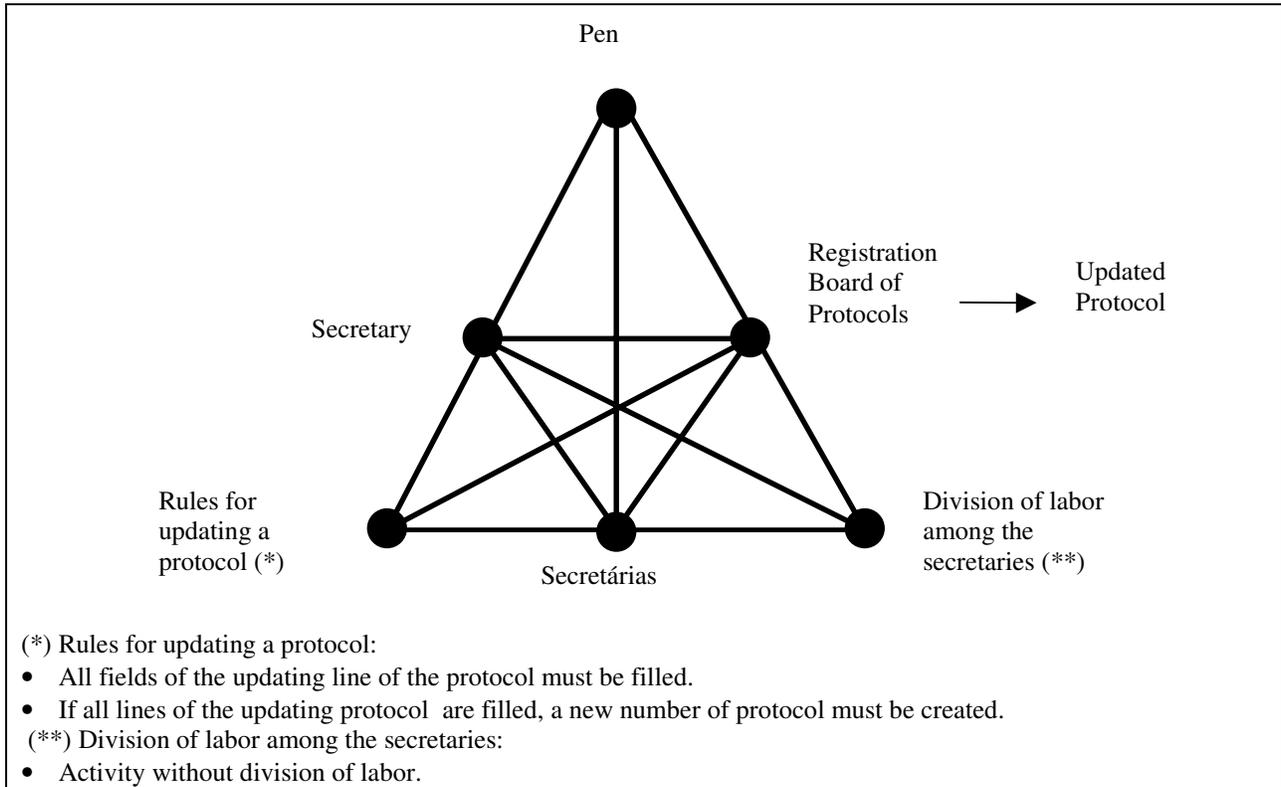


Figure 7 – Systemic Model of the activity "Update Protocol".

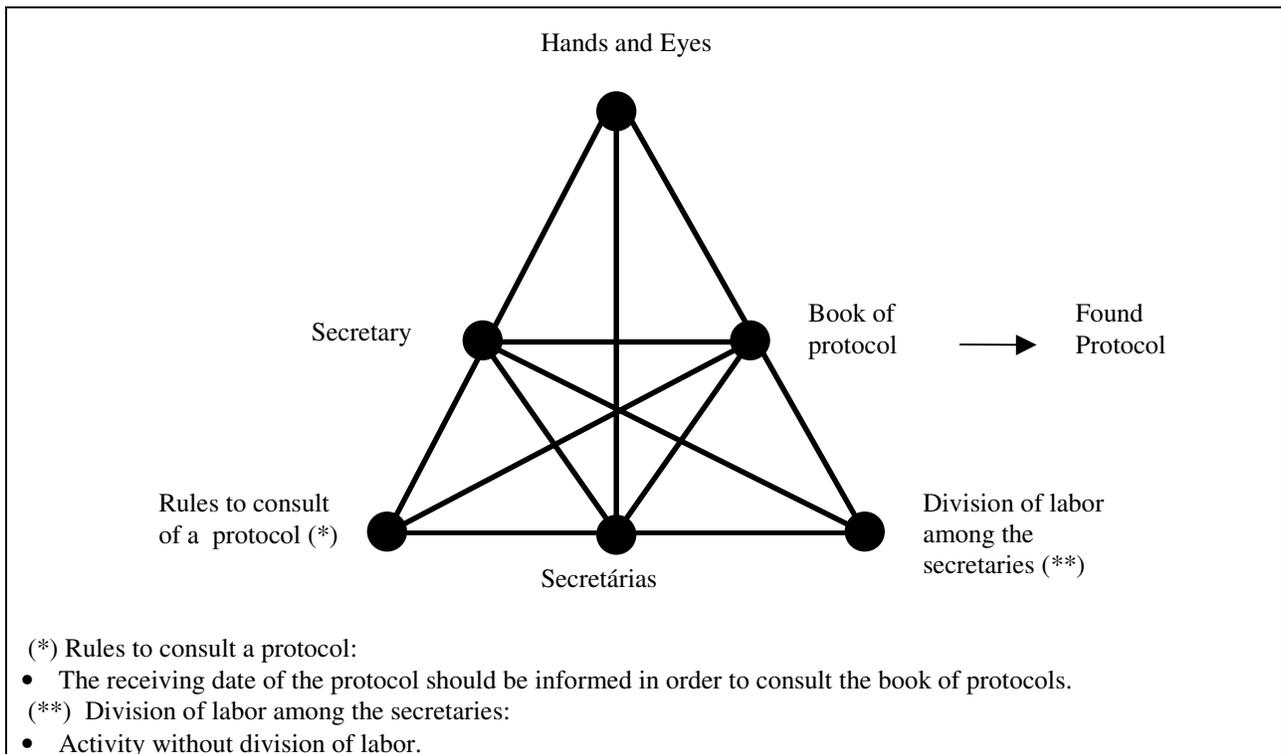


Figure 8 - Systemic Model of the activity "Consult protocol by its date".

Activity	Actions	Operations
Create protocol	Generate protocol number	Verify last protocol number
		Add one to the last protocol number
		Fill field "protocol number"(1)
	Fill first line of the registration board of protocol	Fill field (2)
		Fill field (3)
		Fill field (4)
		Fill field (5)
		Fill field (6)
		Fill field (7)
		Fill field (8)
	Submit document	Stamp document to be sent
		Copy protocol number in the stamped document
		Send document to receiver

Table 2- Decomposition of the activity "Create protocol".

Activity	Actions	Operations
Update protocol	Find registration board of protocol in the book of protocols	Verify protocol number in the received document
		Find the correspondent number in the book of protocols
	Fill next line in the registration board of protocols	Fill field (4)
		Fill field (5)
		Fill field (6)
		Fill field (7)
		Fill field (8)
		Fill field (9)

Table 3- Decomposition of the activity "Update Protocol".

Activity	Actions	Operations
Consult protocol by date	Find protocols	Specify date to consult
		Find protocols numbers according to the specified date
		Inform consulted protocols

Table 4 - Decomposition of the activity "Consult protocol by date".

3. Decomposition of an activity into Actions and Operations.

In order to represent in a more detailed way the activities mentioned in the previous section, they will be decomposed into actions and operations, according to the principle (3) from Activity Theory.

The decomposition of activities "Create Protocol", "Update Protocol" and "Consult Protocol by its Date" is presented through the tables 2, 3, and 4.

2. Conclusions

It seems that the systemic relations that exist in an activity context contribute to more careful requirements elicitation. However, it is essential for the person

performing the elicitation to consider important elements necessary to the understanding of a problem. Such elements include subject, mediation tools, object, community, rules, and division of labor.

The hierarchical structure of an activity, composed by actions and operations, and their "movements" along the historical development of an activity also contribute to a better understanding of the analyzed problem.

Nowadays there are some research methods being developed in the requirements elicitation area that use the concept of scenario¹⁹ to support requirements elicitation of the macro-system [3][11]. Considering that the concept of scenario can be structured through

¹⁹ The concept of scenario is used in several objected-oriented analyses method [2][9][13].

concepts like context, episode, objective, actor, and resource [3], a parallel can be presented between the concepts of scenario and activity. This parallel shows that a lot of the elements are equivalent. An actor in the scenario concept is a subject in the activity concept. A resource in the scenario concept is an object in the activity concept. Context in the scenario concept is determined by the rules of the activity while the episode in the scenario concept is an action in the activity concept and the goal of the scenario is the result of an activity.

Nevertheless the activity concept brings more elements inside its structure than the scenario concept: community, mediation tools, division of labor inside the community, and operations. Thus, we believe that several precepts from Activity Theory can contribute for process of requirements elicitation based on scenarios.

We argued in the introduction that we can divide the problems faced in the requirements elicitation of the software into two major groups: accidental problems and essential problems. The essential problems contain the real difficulties in the requirements elicitation process. We believe that some of the precepts of Activity Theory can contribute to overcoming some of those difficulties.

References

- [1] Boehm, B., *Software Engineering Economics*, Prentice-Hall, 1981.
- [2] Booch, G., *Object-Oriented Analysis and Design with Application*, Benjamin/Cummings, 1994.
- [3] Breitman, K. K. e Leite, J. C. S. P., "Suporte Automatizado à Gerência da Evolução de Cenários", Workshop de Engenharia de Requisitos - WER98, 1998.
- [4] Brooks, F., "No Silver Bullet: Essence and Accidents of Software Engineering", *Computer*, Apr. 1987, pp. 10-19.
- [5] Engeström, Y., "Learning by Expanding", Helsinki: Orienta-Konsultit, 1987.
- [6] Faulk, S. R., "Software Requirements: A Tutorial", *Software Requirements Engineering*, 2nd. Ed., IEEE CS Press, 1997, pp 128-149.
- [7] US General Accounting Office, *Mission Critical Systems: Defense Attempting to address Major Software Challenges*, GAO/IMTEC-93-13, December 1992.
- [8] Goguen, J. A. and C. Linde, "Techniques for Requirements Elicitation", *Software Requirements Engineering*, 2nd. Ed., IEEE CS Press, 1997, pp 110-122.
- [9] Jacobson, I. *et al.*, *Object-Oriented Software Engineering*, Addison-Wesley, 1992.
- [10] Kaptelinin, V. and Nardi, B. A., "Activity Theory: Basic Concepts and Applications", CHI 97 Electronics Publications: Tutorials, march/1997.
- [11] Leite, J.C.S.P. *et al.*, "Enhancing a Requirements Baseline with Scenarios", *Requirements Engineering Journal: Springer-Verlag London Limited* Vol. 2, N. 4 pp 184-198, 1997.
- [12] Nardi, B. A., *Context and Consciousness - Activity Theory and Human-Computer Interaction*, MIT Press, 1996.
- [13] Rumbaugh, J. *et al.*, *Object-Oriented Modeling and Design*, Prentice Hall, 1991.

Apêndice B

Artigo apresentado no *XIX Concurso de Trabalhos de Iniciação Científica* da SBC (XIX CTIC), evento vinculado ao XX Congresso da Sociedade Brasileira de Computação, realizado na PUC-PR em julho de 2000.

Estudo Comparativo sobre as Técnicas de Elicitação de Requisitos do Software

Anderson Belgamo
u9816034@fcti.unimep.br

Luiz Eduardo Galvão Martins (Orientador)
lgmartin@unimep.br

Universidade Metodista de Piracicaba - UNIMEP
Rod. do Açucar, Km 156 13400-901
"Campus" Taquaral - Piracicaba-SP

Resumo

Este artigo tem como objetivo apresentar um estudo comparativo sobre as técnicas de elicitação de requisitos do software, sendo que a elicitação de requisitos é a primeira fase dentro da Engenharia de Requisitos. A elicitação de requisitos é de extrema importância para a construção do software, oferecendo significativas contribuições para a Engenharia de Software. Na elicitação de requisitos existem dois grupos de problemas: essenciais e acidentais. Para o enfrentamento desses problemas várias técnicas de elicitação de requisitos tem sido propostas, as quais estão sendo estudadas e avaliadas de forma a apontar as diferenças e semelhanças entre elas. Na avaliação das técnicas estudadas, propomos vários parâmetros de avaliação. Entendemos que estes parâmetros são importantes dentro do processo de avaliação das técnicas, pois são balizas úteis para a comparação entre as mesmas.

Abstract

The goal of this article is to show a comparative study about the software requirement elicitation techniques, where the requirements elicitation is the first step in the Requirements Engineering. The requirements elicitation is very important for software construction, offering meaningful contributions for Software Engineering. During requirements elicitation there are two problem groups: essential and accidental. To overcome these problems several requirement elicitation techniques have been proposed, which are being evaluated and analyzed to point out the differences and similarities among them. In evaluation of the techniques studied we propose a lot of parameters of evaluation. We understand that these parameters are important in the process of evaluation of the techniques, because they are useful references in the comparison among them.

Palavras-Chave

Requisitos, Elicitação de Requisitos, Engenharia de Requisitos

1. Introdução

Este artigo é resultante das atividades desenvolvidas no projeto de iniciação científica intitulado “Um Estudo Comparativo sobre as Principais Técnicas de Elicitação²⁰ de Requisitos do Software”. Este projeto, que ainda encontra-se em desenvolvimento, tem como objetivo estudar as principais técnicas de elicitação de requisitos do software, comparando-as de forma a explicitar pontos fortes e fracos de cada técnica. A elicitação de requisitos do software é uma atividade importante da Engenharia de Requisitos, que está inserida dentro do contexto da Engenharia de Software. A Engenharia de Requisitos se preocupa com a elicitação, especificação, análise, verificação e gerenciamento dos requisitos do software. A elicitação de requisitos oferece a base para as demais atividades da Engenharia de Requisitos.

2. Engenharia de Requisitos

2.1. Definição

A engenharia de requisitos, segundo (Thayer, 1997), é a primeira etapa dentro de todo o processo da engenharia de software, a qual estuda como coletar, entender, armazenar, verificar e gerenciar os requisitos. A principal preocupação na Engenharia de Requisitos é entender quais são os reais requisitos do sistema, bem como a documentação dos mesmos.

2.2. Fases da Engenharia de Requisitos

Como especificado em (Thayer, 1997) as fases da Engenharia de Requisitos são: elicitação, análise, especificação, verificação e gerenciamento. Elicitação: é o processo através do qual clientes e usuários são questionados por um desenvolvedor para falarem “o quê” o software deve fazer (ou seja, os requisitos). Esta fase será mais detalhada na Seção 2.3. Análise: é o processo onde são analisadas as necessidades dos clientes e usuários para se chegar na definição dos requisitos de software. Especificação: é o processo de criação de um documento no qual estão definidos todos os requisitos analisados. Verificação: é o processo que busca assegurar que a especificação de requisitos de software está em concordância com os requisitos elicitados e analisados nas fases anteriores, ou seja, estão de acordo com o desejo do cliente. Gerenciamento: é o planejamento e controle da atividade de elicitação, especificação, análise e verificação dos requisitos.

2.3. A Elicitação de Requisitos do Software

A elicitação de requisitos é o início para toda a atividade de desenvolvimento de software, onde técnicas de elicitação são utilizadas. Embora a elicitação de requisitos seja a primeira atividade na Engenharia de Requisitos, esta atividade não acontece somente uma vez, seu processo é iterativo, ou seja, todas as demais etapas da Engenharia de Requisitos podem conter elicitação de requisitos. Para começarmos a elicitação precisamos identificar as pessoas certas, ou seja, os usuários finais. Ao iniciar a elicitação, os usuários podem ter

²⁰ A palavra elicitação vem do inglês *elicitation*, que significa descobrir algo obscuro.

um forte papel no processo de desenvolvimento, através de seus conhecimentos e experiências (Goguen, 1997). Para a escolha dos usuários precisamos nos preocupar que todas as informações dadas por eles sejam de confiança, ou seja, estas informações serão usadas para a fase de projeto, estando de acordo com a necessidade da empresa. Portanto, este usuário é de extrema importância, pois a escolha de uma pessoa errada acarretará em uma perda de tempo e dinheiro para ambas as partes envolvidas no desenvolvimento do software.

Como a natureza dos requisitos é mutante, ao iniciarmos a fase de elicitação de requisitos, os clientes podem mudar seus pensamentos, uma vez que eles vêem várias possibilidades mais claramente em momentos posteriores (Goguen, 1997), fazendo com que os requisitos mudem e, conseqüentemente, toda a fase de elicitação e análise dos requisitos sofrem alterações. Essas possibilidades podem vir de mudanças nos fatores sociais, financeiros, psicológicos e/ou políticos. Além da identificação dos usuários o desenvolvedor precisa conhecer o contexto no qual as informações estão situadas. Segundo (Jark, 1994) o contexto é dividido em 3 mundos: sujeito, uso e sistema. O sujeito representa uma parte do mundo exterior, onde o sistema – representado por alguma descrição estruturada – existe para servir algum indivíduo ou usuário (Siddiqi, 1997). Segundo Goguen, requisitos são informações e todas as informações estão situadas num contexto e isto determina o significado dos requisitos. Levar em consideração o contexto, significa prestar atenção em fatores sociais e técnicos. Requisitos emergem de situações sociais entre usuários e analistas (Siddiqi, 1997). Dessa interação com o usuário é que nascem os requisitos para a construção do software, porém, a parte mais difícil da elicitação de requisitos não é o ato de registrar o que os usuários querem; na verdade a atividade de elicitação é exploratória e desenvolvimental, ajudando os usuários a compreenderem melhor o que eles querem (McConnel, 1998). Uma antecipação do projeto, onde todos os requisitos ainda não estejam elicitados, levaria a um erro, pois o software desenvolvido não estaria de acordo com as necessidades do cliente, acarretando gastos financeiros e de tempo.

2.4. Problemas encontrados durante a Elicitação de Requisitos do Software

Durante a elicitação de requisitos, e até mesmo no início da elicitação, aparecerão muitos problemas. No início da elicitação temos o problema da escolha dos usuários a serem entrevistados, pois uma escolha errada pode levar à perda de tempo e prejuízos financeiros para o desenvolvedor e para os clientes. Já durante a elicitação aparecerão problemas de escopo, de entendimento e de volatilidade. Além dos problemas citados acima teremos dois grandes grupos de problemas: problemas acidentais e problemas essenciais (Faulk, 1997).

a) Problemas Acidentais: São aqueles oriundos da falta de controle sobre aquilo que precisa ser construído, dentre os quais podemos destacar: pouco esforço despendido no levantamento de informações junto ao usuário; documentação pobre sobre os requisitos obtidos, pouca revisão dos requisitos obtidos; especificações incorretas dos requisitos e tendência em iniciar logo o processo de desenvolvimento do software (Martins, 1999), o que leva o usuário a não estar satisfeito com o resultado final do projeto. As dificuldades acidentais, segundo (Faulk, 1997), são: 1) Documentação escrita como uma reflexão tardia:

isto continua sendo uma prática comum, onde a documentação de requisitos é desenvolvida depois que o software tenha sido escrito. Tal procedimento inevitavelmente viola o princípio da definição do “o que” o sistema deve fazer antes de “como”. 2) Não projetada para ser proveitosa: freqüentemente, na vontade implementar logo, pouco esforço é despendido para projetar, escrever, checar, ou gerenciar a administração da criação e evolução do documento de especificação. O resultado mais óbvio é uma organização pobre do software. O documento resultante não é uma referência técnica efetiva. A especificação é difícil para usar e difícil para manter.

b) Problemas Essenciais: São aqueles inerentes à elicitação de requisitos, dentre os quais podemos destacar: dificuldades do usuário em saber efetivamente o que ele quer, dificuldade de comunicação entre usuário e desenvolvedor e a natureza mutante dos requisitos (Martins, 1999). Segundo (Faulk, 1997) os problemas essenciais seriam causados por dois fatores primordiais: 1) Compreensão: pessoas não sabem o que elas querem. Isto não significa que as pessoas não tenham uma idéia geral do que o software fará. Ao invés disso, eles não começam com um preciso e detalhado entendimento de que funções pertencem ao software, o que as saídas devem fazer para cada possível entrada, quanto tempo cada operação deveria levar, como uma decisão afetará outra, e assim por diante. Muitas decisões ainda não foram tomadas, e expectativas mudarão na medida que o problema é melhor entendido. 2) Comunicação: requisitos de software são difíceis para se comunicar efetivamente. A dificuldade inerente da comunicação é composta pela diversidade de pessoas e audiências para a especificação de requisitos.

c) Diferenças: Podemos afirmar que os problemas acidentais são mais fáceis de se evitar, dependendo apenas da utilização das fases da Engenharia de Requisitos citadas na Seção 1.2. Porém, os problemas essenciais envolvem a comunicação entre pessoas e esse processo deve levar em conta o contexto, as habilidades pessoais do entrevistado, o lado psicológico, ou seja, apenas uma abordagem tecnológica não poderá solucionar esses problemas, pois os aspectos sociais assumem grande importância na elicitação dos requisitos (Martins, 1999).

3. Técnicas de Elicitação de Requisitos

Para uma boa elicitação de requisitos, existem técnicas para o melhor entendimento e comunicação entre clientes e analistas, para que problemas, como os citados nas seções anteriores, não ocorram, ou se ocorrerem, que sejam mais facilmente resolvidos. Segundo (Jackson, 1995), uma técnica deve explorar as características específicas do problema e como as características do problema variam muito, nós precisamos de um repertório de métodos para cada classe de problema (Siddiqi, 1997). O problema da elicitação de requisitos não pode ser resolvido com uma abordagem puramente tecnológica, porque nesta fase o contexto social é mais crucial do que na fase de programação, especificação e projeto. A seguir será apresentado um elenco de técnicas que foram pesquisadas e que serão comparadas na Tabela da Seção 4.

a) Observação

A observação possibilita um contato pessoal e estreito do pesquisador com o fenômeno pesquisado, o que apresenta uma série de vantagens. (Damian, 1997) . As

técnicas de observação são extremamente úteis para “descobrir” aspectos novos de um problema. Isto se torna crucial nas situações em que não existe uma base teórica sólida que oriente a coleta de dados. Ao mesmo tempo em que o contato direto e prolongado do pesquisador com a situação pesquisada apresenta as vantagens mencionadas, envolve também uma série de problemas. Algumas críticas são feitas ao método de observação, primeiramente por provocar alterações no ambiente ou no comportamento das pessoas observadas. Outra crítica é a de que este método se baseia muito na interpretação pessoal. Além disso há críticas no sentido de que o grande envolvimento do pesquisador leve a uma visão distorcida do fenômeno ou a uma representação parcial da realidade.

b) Entrevista

Entrevista é uma técnica de elicitação de requisitos muito usada. O engenheiro de requisitos ou analista discute o sistema com diferentes usuários, a partir da qual elabora um entendimento de seus requisitos. Há basicamente, segundo (Kotonya, 1998), 2 tipos de entrevista: a) entrevistas fechadas onde o engenheiro de requisitos procura as perguntas para um conjunto um pré-definido de questões; b) entrevistas abertas onde não há agenda pré-definida e o engenheiro de requisitos discute, de modo aberto, o que os usuários querem do sistema. Entrevistas podem ser efetivas para desenvolver um entendimento do problema e para elicitar muitos requisitos gerais do sistema. Usuários finais são usualmente felizes para descreverem seus trabalhos e as dificuldades que eles enfrentam de forma relativamente natural, entretanto eles podem ter expectativas não realistas sobre o suporte que o computador dará. Portanto, entrevistas são muito menos efetivas para entendimento do domínio da aplicação e para o entendimento das questões organizacionais as quais afetam os requisitos.

c) Análise de Protocolo

A análise de protocolo pede à pessoa se engajar em alguma tarefa e correntemente falar sobre esta tarefa, explicando o seu pensamento do processo. Usuários alegam que esse tipo de linguagem pode ser considerada uma “verbalização direta do processo cognitivo específico”. A análise de protocolo não é um guia confiável sobre o que as pessoas estão pensando, ela está sujeita a problemas de interpretações pelos analistas. A restrição em estudar protocolos é que as pessoas podem produzir linguagens que oferece um perfil de atividade cognitiva autônoma. Segundo (Goguen, 1997), mesmo se fosse possível conseguir um perfil de atividade cognitiva autônoma da pessoa, tal objeto seria inapropriado para o processo de requisitos, porque o cliente não tem qualquer modelo mental pré-existente do sistema desejado. Os clientes têm conhecimento sobre negócios e necessidades organizacionais, enquanto o time de requisitos tem conhecimento sobre as possibilidade técnicas.

d) JAD²¹

²¹ *Joint Application Development*

JAD é uma marca registrada da IBM. O tema principal do JAD é colocar autoridades representativas e gerenciais juntas dentro de um *workshop* estruturado para promover decisões. Segundo (Damian, 1997) JAD consiste de 5 fases: definição do projeto, pesquisa, preparação para a sessão JAD, a sessão JAD, o documento final. As fases de definição de projeto e pesquisa no processo JAD lidam com a coleta de informações. A sessão JAD é então usada para validar as informações recolhidas nas fases anteriores. O processo JAD concentra-se na sessão JAD, e deste modo JAD contribui para a elicitação de requisitos como significado para validar as informações já colhidas. Na sessão JAD, as pessoas certas têm que estar envolvidas, e a presença de um facilitador pode ajudar a manter a sessão focalizada e minimizar ataques e defesas emocionais improdutivas. JAD promove cooperação, entendimento, e trabalho em grupo entre os vários grupos de usuários e o pessoal de sistemas de informação.

e) PD²²

Tradicionalmente valores democráticos, os quais tem sido levados em conta no processo de projeto, tem sido somente aqueles concentrados em fatores técnicos e econômicos. Mas com o uso do PD fatores técnicos-sociais tem sido levados em conta. O projeto deveria ser feito com o usuário. Aprendizado mútuo deveria ser uma parte importante do trabalho em um grupo de projeto, não sendo meramente uma visita aos laboratórios de pesquisa (Damian, 1997). Trabalhadores e clientes são inteligentes e criativos, contribuidores produtivos para as organizações, desde que sejam encorajados a expressarem seus desejos, aplicarem sua esperteza e exercitarem suas capacidades de tomar decisões, assumindo responsabilidades do impacto de suas ações.

f) QFD²³

O termo qualidade é definido como “um conjunto de meios para produzir economicamente produtos ou serviços, os quais satisfaçam os requisitos do cliente”. QFD é “um conceito que provê meios de interpretar requisitos do cliente em requisitos técnicos, apropriados para cada estágio do desenvolvimento e produção do produto” (Damian, 1997). As fases iniciais do QFD podem ser descritas como sendo “simplesmente um sistema de identificação e priorização das necessidades do cliente obtidas de cada recurso avaliado”. QFD é um conceito que aplica-se bem para a elicitação de requisitos, especialmente num modelo de elicitação onde a voz do cliente é o guia para a criação de requisitos.

g) CRC²⁴

Como definido em (Zhu, 1998), CRC é uma sessão de grupo, que é similar ao JAD, onde os papéis dos participantes e o papel do facilitador são claramente definidos. Em CRC, participantes consistem não somente de usuários e facilitador, mas também de outras

²² *Participatory Design*

²³ *Quality Function Deployment*

²⁴ *Cooperative Requirements Capture*

peessoas envolvidas indiretamente no sistema. CRC é diferente de JAD e QFD pois ele foca-se no usuário operativo.

h) Prototipação

Este método para elicitação de requisitos utiliza-se do uso de uma técnica de prototipação para a avaliação do usuário. O conjunto inicial de requisitos é usado como base para criar o Protótipo de Interface do Usuário com o sistema inicial (simplificado). Para essa criação, o projetista precisa manter o protótipo tão simples quanto possível. O ponto forte desta atividade é apresentar muitas alternativas para o usuário antes de se gastar muito esforço para qualquer protótipo em particular. Após a aceitação do protótipo pelos usuários, os desenvolvedores precisam criar um documento de especificação dos requisitos paralelo ao protótipo de interface (McConnel, 1998).

i) Cenários

Usuários finais e outros agentes do sistema acham a utilização de cenários mais fáceis para relacionar-se aos exemplos da vida real do que descrições abstratas das funções realizadas pelo sistema. Por essa razão, é freqüentemente útil desenvolver um conjunto de interação dos cenários, e usar estes para elicitar e clarear requisitos de sistema. Cenários são exemplos de sessões de interação as quais são concentradas com um tipo único de interação entre um usuário final e o sistema. Usuários finais simulam suas interações usando cenários. Eles explicam para o time de engenheiros de requisito o que eles estão fazendo e a informação da qual eles precisam do sistema para descrever a tarefa descrita no cenário.

4. Parâmetros de Avaliação das Técnicas de Elicitação de Requisitos

A partir das técnicas citadas na Seção 3, foram propostos vários parâmetros para a avaliação das técnicas (*vide Tabela 1*). Os parâmetros propostos são explicados abaixo:

a) Grupo/Indivíduo: indica se a técnica é aplicada em grupo ou individualmente; **b) Contexto:** indica se a técnica leva em conta o ambiente onde está se realizando a elicitação; **c) Caráter de interação:** indica se o entrevistador e entrevistado sentem-se a vontade, num clima de estímulo e de aceitação mútua; **d) Usa lado introspectivo:** voltar-se a si próprio e pensar como seria o serviço; **e) Confiabilidade:** se as informações colhidas são confiáveis para o desenvolvimento do projeto; **f) Custo:** esforço gasto no uso da técnica; **g) Qualidade:** é um critério de validação onde é perguntado se na técnica há democracia, aprendizado mútuo, educação e resolução de conflito; **h) Padronização:** se a técnica possui uma regra para seu uso; **i) Produtividade:** se é uma técnica produtiva; **j) Quantidade:** é um critério de validação onde é perguntado se na técnica há índices de performance e economia de tempo; **k) Descrição de ações dos usuários:** se a técnica registra as gesticulações e faces do rosto do usuário; **l) Compartilhamento de informações:** se todos os indivíduos do grupo compartilham as informações; **m) Tempo:** tempo despendido para a elicitação de requisitos; **n) Promover cooperação:** se a técnica promove a cooperação entre os indivíduos do grupo; **o) Facilitador:** se possui uma pessoa com a função de guiar, levantar questões e discussões

Uma Metodologia de Elicitação de Requisitos de Software Baseada na Teoria da Atividade

num grupo; p) Validar requisitos com os usuários: se a técnica valida seus requisitos com os usuários; q) Conflitos entre usuários do grupo: se na técnica existe um meio para lidar com conflitos em grupo; r) Atividade prematura de projeto: se a técnica evita que os analistas pensem que todos os requisitos já foram elicitados e que o projetista já possa começar a elaboração do sistema.

Tabela 1 - Cruzamento entre as técnicas de elicitação com os parâmetros de avaliação propostos

Técnicas de Elicitação	Parâmetros de Avaliação								
	Observação	Entrevista	Análise de Protocolo	JAD	PD	QFD	CRC	Prototipação	Cenários
Grupo/Indivíduo	Grupo/Indivíduo	Indivíduo	Indivíduo	Grupo	Grupo	Grupo	Grupo	Grupo/Indivíduo	Grupo/Indivíduo
Considera o contexto	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Caráter de Interação	Não	Sim	Não	Sim	Sim	Sim	Sim	Sim	Sim
Liberdade de Percurso	Sim	Sim	Não	Não	Não	Não	Não	Não	Sim
Usa lado introspectivo	Sim	Sim	Sim	Não	Não	Não	Não	Não	Não
Confiabilidade	Média	Alta	Baixa	Alta	Alta	Alta	Alta	Alta	Alta
Custo	Baixo	Médio	Médio	Alto	Alto	Alto	Alto	Alto	Alto
Qualidade	Média	Média	Baixa	Alta	Alta	Alta	Alta	Alta	Alta
Padronização	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Produtividade	Baixa	Média	Baixa	Alta	Alta	Média	Alta	Alta	Alta
Quantidade	Baixa	Alta	Baixa	Alta	Média	Média	Média	Média	Média
Descreve ações dos usuários	Sim	Não	Não	Não	Não	Não	Não	Não	Sim
Compartilhamento de informações	Não	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim
Tempo	Longo	Médio	Médio	Longo	Longo	Longo	Longo	Médio	Longo
Promove cooperação	Não	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim
Facilitador	Não	Não	Não	Sim	Sim	Não	Sim	Não	Não
Valida requisitos com os usuários	Não	Sim	Não	Sim	Sim	Sim	Sim	Sim	Sim
Conflitos entre os usuários do grupo	Não	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim
Evita atividade de projeto prematura	Não	Não	Não	Sim	Sim	Sim	Sim	Sim	Sim

5. Conclusão

O projeto exposto deverá resultar em uma monografia sobre o assunto, que auxiliará no desenvolvimento do ensino de engenharia de software em disciplinas do curso de Ciência da Computação. Os parâmetros de avaliação das técnicas foram obtidos estudando e analisando as diferentes técnicas de elicitação encontradas nos artigos, livros e páginas na

Internet. Durante o levantamento bibliográfico sobre o tema não foi encontrado nenhum material propondo parâmetros de comparações entre as técnicas, reforçando a relevância do estudo comparativo das técnicas de elicitação de requisitos.

Um estudo de caso está sendo realizado e servirá como uma forma de aplicação das técnicas, que também auxiliará no refinamento dos parâmetros de avaliação propostos. O motivo desse estudo de caso é realizar uma avaliação efetiva entre as técnicas, podendo diferenciá-las, obtendo dados para avaliar onde cada técnica seja melhor aplicada, de maneira a ajudar o engenheiro de requisitos a escolher a técnica mais adequada para a situação defrontada. Durante o estudo de caso já foram aplicadas as técnicas de observação, questionário, análise de protocolo, JAD e entrevista. Após a aplicação das demais técnicas poderá ser feita uma melhor avaliação de seu uso, oferecendo subsídios para o engenheiro de requisitos escolher a técnica de elicitação mais adequada para a situação defrontada.

6. Referências Bibliográficas

- DAMIAN, Adrian, et al. “Joint Application Development and Participatory Design” 1997. <<http://www.cpsc.ucalgary.ca/~pand/seng/613/report.html>>
- FAULK, Stuart R. “Software Requirements: A Tutorial” in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p. 128-149
- GOGUEN, Joseph A. “Techniques for Requirements Elicitation” in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p.110 –122
- HARWELL, Richard. “What Is a Requirement” in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p. 23-29
- JACKSON, Michael. “Software Requirements and Specifications”, Addison-Wesley, Reading, Mass., 1995.
- JARK, M e POHL, K. “Requirements Engineering in 2001: (Virtualy) Managing a Changing Reality”, Software Requirements Engineering, Nov. 1994, p.p. 257-266.
- KOTONYA, Gerald e SOMMERVILLE, Ian. Requirements Engineering Processes e Techniques. John Wiley and Sons, 1998
- MARTINS, Luiz E. G. “Utilização dos Preceitos da Teoria da Atividade na Elicitação de Requisitos do Software”, XII Simpósio Brasileiro de Engenharia de Software, Out. 1999.
- McCONNEL, Steve. “Software Project Survival Guide: How to Be Sure Your First Important Project isn’t Your Last”. 1998, p.p 113-124.
- SIDDIQI, Jawed. “Requirements Engineering: the emerging wisdom” in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p. 36-40.
- THAYER, R. H. e DORFMAN, M.; “Introduction to Tutorial Software Requirements Engineering” in Software Requirements Engineering, IEEE-CS Press, Second Edition, 1997, p.p. 1-2.
- ZHU, David. “Requirements Engineering”. 1998. <<http://www.cpsc.ucalgary.ca/~zhud/seng693.html>>

Apêndice C

Artigo apresentado no *V Workshop de Teses em Engenharia de Software (V WTES)*, evento vinculado ao XIV Simpósio Brasileiro de Engenharia de Software, realizado em outubro/2000 no CEFET de João Pessoa - PB.

**An Approach of Software Requirements Elicitation
Based on Activity Theory**

Luiz Eduardo Galvão Martins
(*Graduate Student*)
Methodist University of Piracicaba
lgmartin@unimep.br

Beatriz Mascia Daltrini
(*Supervisor*)
State University of Campinas
beatriz@dca.fee.unicamp.br

Abstract

The goal of this work is to show an approach of software requirements elicitation based on Activity Theory. This theory has been developed in the psychology area and takes into account several important elements to analyze the context where people perform their actions. Considering the activity as unit of analysis the Activity Theory offers an interesting framework to structuring and organizing the actions performed by the system stakeholders. We intend to use this framework to help in the discovery and capture of software requirements.

Keywords

Activity, Activity Theory, Software Requirements, Software Requirements Elicitation

1. Introduction

A classical problem in Requirements Engineering is how to determine real user needs and transform them in safe software requirements. Researches have proved that many software projects have failed because of the problems in Software Requirements Engineering [Boe81] [Gao92]. Specifically, the requirements that are obtained are often incomplete, misunderstood, and ambiguous.

The discovery of software requirements is not an easy task because of the abstract nature of software. To discover software requirements is a task performed in the elicitation process, the first step of Requirements Engineering [Kot98]. The problems found in requirements elicitation can be divided into two categories [Bro87] [Fau97]: accidental problems and essential problems. The accidental problems emerge because of poor control over the activities developed in requirements engineering: low effort in the requirements elicitation with the user, poor documentation about the requirements, poor revision of the requirements, incorrect specification of the requirements, and tendency to initiate prematurely the software development process. The essential problems are embedded in requirements elicitation are: difficulty of the user to know exactly what he wants, difficulty concerning the communication between user and developer, and the changing nature of the requirements.

The accidental problems can be considered less difficult to overcome. The adoption of a systematic process that orient the elicitation, analysis, specification, validation and management of the requirements tend to solve, or at least minimize, the problems of that category.

Nevertheless, the essential problems are more difficulty to be overcome, once they are contained in the requirements nature. The adoption of a systematic process to requirements engineering, mainly to specification, validation and management of the requirements, will also aid in overcoming the essential problems. However, the problems that naturally exist in the human comprehension and communication process, which is in the nucleus of the requirements elicitation, will need an approach which takes into account the context in which the persons develop their activities and recognize the objects needed to develop them, the historic of evolution of these activities and mediation tools, and others aspects of psychological and social relevance that affect the users of the software to be developed.

Thus the essential problems of requirements elicitation will not be solved in a purely technological approach when social aspects have strong importance in the activity [Gog93]. The majority of software is developed with no one help from the social sciences (such as psychology, sociology, anthropology etc.), not approaching in a systematic way the users' necessities, both at individual and organizational levels. We argue in this article that the utilization of some precepts from Activity Theory, coming from Soviet psychology, can bring important benefits to the software elicitation process.

2. Related Works

In [McG00] is presented an approach to modeling “softer” aspects of the software development process. This approach considers that several notions from Activity Theory can offer useful support for the critical “people-related” applications. Such notions include intentionality, history, mediation, motivation, understanding, communication, culture and context.

There are some research methods being developed in the requirements elicitation area that use the concept of scenario to support requirements elicitation of the macro-system [Bre98]. Considering that the concept of scenario can be structured through concepts like context, episode, objective, actor, and resource [Bre98], a comparison can be presented between the concepts of scenario and activity.

Experiment results are presented in [Taw98] where textual scenario analysis were used in the requirements elicitation process. Such process is called CREWS²⁵-L’Ecritoire. This approach proposes to exploit a bi-directional coupling between goals and scenarios to support the requirements elicitation process.

3. Methodology

The methodology of work that has been adopted consisted of:

- Study of software development methodologies;
- Study of Activity Theory;
- Study of Requirements Engineering;
- Development of a case study;
- Preparation and submission of papers to conferences;
- Comparison between activity and use case concepts;
- Structure of the thesis.

Study of software development methodologies.

Several software development methodologies were studied, among them OOAD [Boo94], Fusion [Col94], OMT [Rum91], OOSE [Jac92] and USDP [Jac98]. All these methods offer a lot of modeling techniques to requirements specification, but few attention has been spent to requirements elicitation.

Study of Activity Theory.

The study of the Activity Theory has been focused upon the concept that an activity offers a context to understand the actions performed by the subjects involved in the activity concept [Kuu96]. According to the Activity Theory this context is formed by elements like subject,

²⁵ Cooperative Requirements Engineering With Scenarios

mediation tool, object, rules, community and division of labor. We believe that the capture of these notions during the elicitation process can help to discovering software requirements.

Study of Requirements Engineering.

The study of Requirements Engineering has been focused upon the requirements analysis and elicitation [Kot98] [Fau97], which are very tied activities.

Development of a case study.

A case study was developed to apply some precepts from Activity Theory to the requirements elicitation process. The case studied involved the construction of a software to control the protocols of a secretary's office of an university. The system of protocols consists in controlling documents in and out of the secretary's office. Considering any document that goes in and out of the office, it is generated a number for the protocol and the fields from the registration board of protocols are filled.

Preparation and submission of papers to conferences.

After the case study, we prepared and sent papers to some conferences. The goal of this initiative was to show the potential of usage of Activity Theory in the requirements elicitation process. This idea was presented at *14th IEEE International Conference on Automated Software Engineering (ASE'99)*, *13th Brazilian Symposium on Software Engineering (SBES'99)* and *II Workshop on Requirements Engineering (WER'99)*.

Comparison between activity and use case concepts.

A comparative study between activity and use case concepts was developed, searching by similarities and differences. Several elements present in the concept of activity have not similar in the concept of use case.

Structure of the thesis.

Finally, we are structuring the thesis and developing other case studies to refine the software requirements elicitation method based on Activity Theory.

4. Applicability

This work intends to purpose a method to software requirements elicitation taking into account that the context analysis is very important to help discover real user needs. We believe that an activity offers a good unit of analysis to understand the actions performed by the actors evolved in the system, as defended by Activity Theory.

The applicability of the approach purposed is directed to systems whose human factors, such as communication and comprehension (essential problems in requirements elicitation), assume a very important role in the requirements definition.

Nowadays there are some research methods being developed in the requirements elicitation area that use the concept of use case and scenarios to support requirements elicitation of the macro-system [Bre98]. We believe that some precepts from Activity Theory can be applied in use case and scenarios approaches.

We argued in the introduction that we can divide the problems faced in the requirements elicitation of the software into two major groups: accidental problems and essential problems. The essential problems contain the real difficulties in the requirements elicitation process. We believe that some of the precepts of Activity Theory can contribute to overcoming some of those difficulties.

5. Conclusion

The article presented tried to offer an overview about the work performed until now and the relevance and applicability of Activity Theory to the software requirements elicitation process. We defend that the systemic relations that exist in an activity context and the notion of unit of analysis through the activity contribute to more careful requirements elicitation. However, it is essential for the person performing the elicitation to consider important elements necessary to the understanding of a problem. Such elements include subject, mediation tools, object, community, rules, and division of labor.

The hierarchical structure of an activity composed by actions and operations, and their “movements” along the historical development of an activity also contribute to a better understanding of the analyzed problem.

References

- [Boe81] Boehm, B., *Software Engineering Economics*, Prentice-Hall, 1981.
- [Boo94] Booch, G., “Object-Oriented Analysis and Design with Application”, Benjamin/Cummings, 1994.
- [Bre98] Breitman, K. K. e Leite, J. C. S. P., "Suporte Automatizado à Gerência da Evolução de Cenários", Workshop de Engenharia de Requisitos - WER98, 1998.
- [Bro87] Brooks, F., "No Silver Bullet: Essence and Accidents of Software Engineering", *Computer*, Apr. 1987, pp. 10-19.
- [Col94] Coleman, D. *et al.*, “Object-Oriented Development: The Fusion Method”, Prentice Hall, 1994.
- [Fau97] Faulk, S. R., "Software Requirements: A Tutorial", in *Software Requirements*

Engineering, 2nd. Ed., IEEE CS Press, 1997, pp 128-149.

- [Gao92] US General Accounting Office, *Mission Critical Systems: Defense Attempting to address Major Software Challenges*, GAO/IMTEC-93-13, December 1992.
- [Gog93] Goguen, J. A. and C. Linde, "Techniques for Requirements Elicitation" in *Software Requirements Engineering*, 2nd. Ed., IEEE CS Press, 1997, pp 110-122.
- [Jac92] Jacobson, I. *et al.*, "Object-Oriented Software Engineering", Addison-Wesley, 1992.
- [Jac98] Jacobson, I., Booch, G. and Rumbaugh, J., "The Unified Software Development", Object Technology Series, Addison-Wesley, 1998.
- [Kot98] Kotonya, G. and Sommerville, I., "Requirements Engineering: Processes and Techniques", John Wiley and Sons, 1998.
- [Kuu96] Kuuti, K., "Activity Theory as a Potential Framework for Human-Computer Interaction" in *Context and Consciousness - Activity Theory and Human-Computer Interaction*, MIT Press, 1996, pp. 17-44.
- [McG00] McGrath, M. G. and Uden, L., "Modelling 'Softer' Aspects of the Software Development Process: An Activity Theory Based Approach", *Proceedings of the 33rd Hawaii International Conference on System Sciences*, 2000 IEEE.
- [Rum91] Rumbaugh, J. *et al.*, "Object-Oriented Modeling and Design", Prentice Hall, 1991.
- [Taw98] Tawbi, M. *et al.*, "Guiding the Process of Requirements Elicitation through Scenario Analysis: Results of an Empirical Study", *Proceedings of the 10th International Workshop on Database & Expert Systems Applications*, 1998 IEEE.

Apêndice D

Artigo enviado em março de 2001 para a “Revista de Ciência & Tecnologia”, publicada semestralmente pela Universidade Metodista de Piracicaba (UNIMEP). Esta revista é indexada por: Base de Dados do Centro de Informações Científicas e Tecnológicas (Comissão Nacional de Energia Nuclear); Base de Dados do IBGE; International Abstracts in Operations Research/IAOR (University of Exeter); Periódica – índice de Revistas Latinoamericanas em Ciências (Unam); Subis (Sheffield Academic Press)

Contribuições da Teoria da Atividade para Processos de Engenharia de Requisitos Baseado em Casos de Uso

Luiz Eduardo Galvão Martins
lgmartin@unimep.br
Universidade Metodista de Piracicaba

Beatriz Mascia Daltrini
beatriz@dca.fee.unicamp.br
Universidade Estadual de Campinas

Resumo

Apresentamos neste trabalho uma breve comparação entre os conceitos de atividade e caso de uso. O conceito de atividade apresentado é extraído da Teoria da Atividade, desenvolvida na área da psicologia, que leva fortemente em consideração o contexto em que uma atividade é realizada. A partir da comparação entre atividade e caso de uso, argumentamos que vários elementos que estão embutidos numa atividade, e que não aparecem na abordagem de casos de uso, podem contribuir para processos de Engenharia de Requisitos baseados em casos de uso, pois levam o engenheiro de requisitos à considerar diversos aspectos relativos ao contexto do sistema que normalmente não são analisados na abordagem de casos de uso.

Abstract

This work presents a slight comparison between activity and use case concepts. The activity concept presented is extracted from Activity Theory, developed in the psychology area. This theory strongly takes into account the context in which the activity is performed. Through the comparison between activity and use case we argue that several elements embedded in the activity are not considered in the use case approach. These elements may contribute to Requirements Engineering process based on use cases, because they induce the requirements engineer to pay attention in several aspects regarding the system context which usually is poorly analyzed in the use cases approach.

Palavras-Chave

Atividade, Caso de Uso, Engenharia de Requisitos, Teoria da Atividade

Keywords

Activity, Activity Theory, Requirements Engineering, Use Case

1. Introdução

Dentre os vários trabalhos desenvolvidos nos últimos anos na área de Engenharia de Requisitos, a utilização de casos de uso para a captura e representação dos requisitos de um sistema vem se estabelecendo como um nicho importante de pesquisa. O conceito de caso de uso também está muito relacionado ao conceito de cenário, que vem recebendo atenção de muitos pesquisadores da comunidade de Engenharia de Requisitos (LEITE, 1997).

Casos de uso são simples de ser utilizados e facilitam o diálogo entre desenvolvedor e usuário, que é um dos principais problemas encontrados durante a análise dos requisitos de um sistema (FAULK, 1997). Através de um diagrama de casos de uso podemos modelar as principais tarefas a serem executadas por um sistema, a partir do ponto-de-vista de seus atores.

No entanto, na maioria dos trabalhos sobre a utilização de casos de uso na Engenharia de Requisitos, pouca atenção tem sido despendida na direção de se obter uma análise mais abrangente sobre o contexto do sistema, que basicamente vem se restringindo à identificação dos atores envolvidos nos casos de uso.

Apresentamos neste trabalho o conceito de atividade extraído da Teoria da Atividade (NARDI, 1996; MARTINS e DALTRINI, 1999), a partir do qual realizamos uma análise sobre a similaridade entre caso de uso e atividade, ressaltando alguns elementos que estão embutidos ao conceito de atividade, mas que não na abordagem de casos de uso, podendo ajudar numa análise mais completa sobre o contexto em que o sistema esteja inserido.

2. Engenharia de Requisitos Baseada em Casos de Uso

A utilização de casos de uso (*use cases*) para a captura de requisitos do sistema vem recebendo um interesse crescente pela comunidade de Engenharia de Requisitos, e sendo bem aceita por muitos metodologistas (RUMBAUGH, 1997). Esta abordagem enfatiza a importância da captura dos requisitos do sistema do ponto-de-vista dos atores que interagem com ele.

A utilização de casos de uso para a captura de requisitos tem sido disseminada a partir do método *Objectory*, proposto por Ivar Jacobson (JACOBSON et al., 1992). Casos de uso também assumem um papel importante na UML²⁶ e USDP²⁷ (JACOBSON et al., 1998; RATIONAL SOFTWARE et al., 1997), que vêm sendo amplamente divulgadas nos últimos anos.

²⁶ Unified Modeling Language

²⁷ Unified Software Development Process

Casos de uso são utilizados para demonstrar a funcionalidade de um sistema. Em UML um caso de uso é definido como um conjunto de seqüências de ações que um sistema realiza produzindo um resultado de valor observável para um ator particular (ERIKSSON e PENKER, 1998). Casos de uso são representados graficamente através de diagramas de casos de uso, conforme mostrado na *Figura 1*.

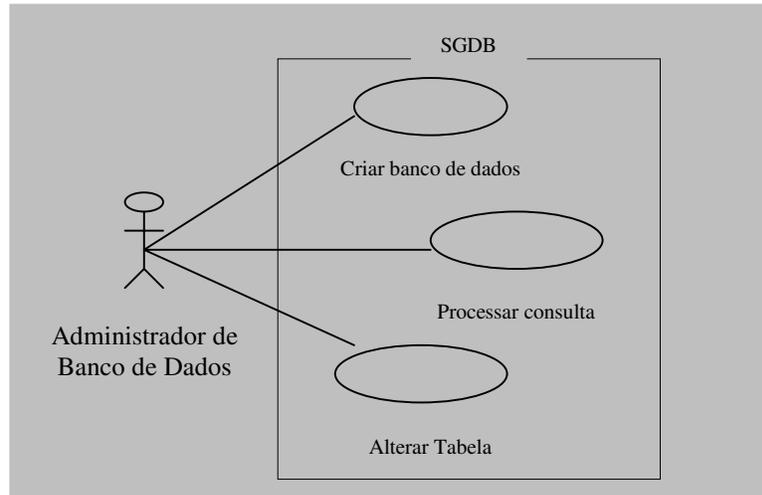


Figura 1 – Diagrama de casos de uso.

A *Figura 1* apresenta um exemplo de digrama de casos de uso, onde o administrador de banco de dados é o ator que interage com o SGDB²⁸ através dos caso de uso “criar banco de dados”, “processar consulta” e “alterar tabela”.

Casos de uso precisam ser descritos, uma vez que através do digrama de casos de uso obtemos apenas uma visão dos casos de uso existentes e dos atores envolvidos. Em (RUMBAUGH, 1997) é apresentada uma sugestão para descrição de casos de uso, conforme apresentado na *Figura 2*.

A descrição de casos de uso sugerida por Rumbaugh traz detalhes da execução do caso de uso que não podem ser demonstrados no diagrama de casos de uso.

Em (SUTCLIFFE et al, 1998) é apresentado um meta-esquema para a modelagem de requisitos baseada em casos de uso, onde são destacados elementos importantes para uma melhor compreensão do caso de uso. A *Figura 3* apresenta o meta-esquema proposto por Sutcliffe.

O meta-esquema proposto por Sutcliffe apresenta a idéia de que múltiplos cenários podem ser gerados a partir de um caso de uso. Além do relacionamento direto com cenário, o caso de uso também se relaciona diretamente com vários outros elementos (propriedade, agente, ação e meta), que contribuem para um desenvolvimento mais acurado da modelagem de requisitos baseada em casos de uso. Os elementos que não se relacionam diretamente com o

²⁸ Sistema Gerenciador de Banco de Dados

caso de uso (evento, objeto, estrutura do objeto, transição de estado e estado), na medida em que vão sendo obtidos durante o processo de Engenharia de Requisitos, auxiliam na formação do contexto em que o caso de uso é executado, e portanto contribuem para o refinamento da modelagem dos casos de uso.

Caso de Uso: Processar Consulta.

Resumo: O administrador de banco de dados elabora uma expressão de consulta SQL (*Structured Query Language*) e envia para o SGBD processar a consulta, que irá retornar com os dados solicitados na consulta.

Atores: Administrador de Banco de Dados.

Pré-condições: Todas as Tabelas envolvidas na expressão de consulta estarem disponíveis no momento do processamento da consulta.

Descrição: A expressão de consulta é interpretada pelo processador de consultas do SGBD, que irá verificar a sintaxe da expressão e os parâmetros da consulta (nome das Tabelas envolvidas para eventuais junções, condição de seleção e campos para projeção). A partir da análise sintática da expressão de consulta, o SGBD deverá abrir as Tabelas envolvidas, realizar as junções necessárias e buscar os campos para seleção e projeção. Filtrar as tuplas e as colunas a partir dos parâmetros estabelecidos na expressão de consulta e apresentar o resultado no formato de uma Tabela (relação gerada pela consulta).

Exceções: Problema na abertura das Tabelas envolvidas na consulta. Nome de campos estabelecidos na expressão de consulta não encontrados nas Tabelas envolvidas. Solicitar revisão na expressão de consulta.

Pós-Condições: Todas as Tabelas utilizadas na consulta devem ser fechadas após o processamento da consulta.

Figura 2 – Exemplo de descrição do caso de uso “Processar Consulta”.

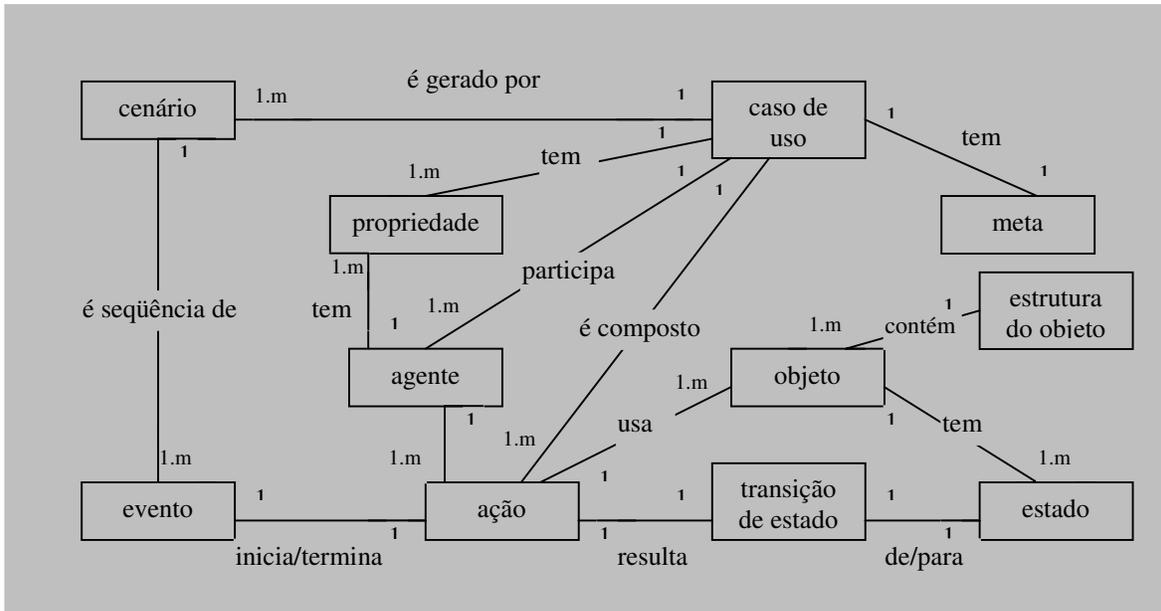


Figura 3 – Meta-esquema para modelagem de requisitos baseada em casos de uso (SUTCLIFFE et al., 1998).

3. O Conceito de Atividade na Teoria da Atividade

A Teoria da Atividade (TA) é formada por um conjunto de conceitos que buscam compreender e explicar porque e como as atividades humanas são desenvolvidas. A TA tem sua origem na psicologia soviética, com fortes influências advindas dos trabalhos de Marx e Engels (WERTSCH, 1981).

Embora a TA possua vários conceitos, destacaremos neste trabalho apenas o conceito de atividade, que será utilizado a seguir para fazermos um paralelo entre atividade e caso de uso.

A TA define uma atividade como uma unidade de análise que possui um conjunto de ações acoplado a um contexto mínimo que permite a sua compreensão. Para toda atividade existe um motivo, que em termos de atividade humana, é determinado pela necessidade ou desejo da pessoa que a realiza (BURD, 1999). A atividade possui uma estrutura hierárquica compreendida por ações e operações, conforme ilustrado na *Figura 4*.

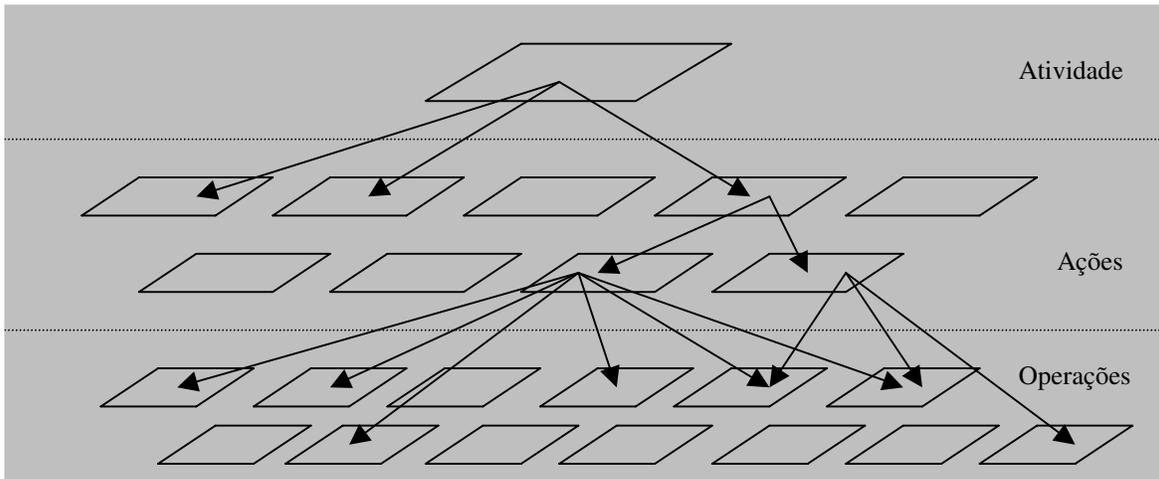


Figura 4 – Estrutura Hierárquica da Atividade (BURD, 1999).

Segundo a TA, uma atividade normalmente envolve a transformação de um ou mais objetos em outro objeto, que é o resultado que se pretende alcançar com a realização da atividade. No entanto, a transformação de um objeto não ocorre de forma direta, ela é realizada através de uma série de passos conscientes, que são chamados de ações. Uma ação possui metas bem definidas e imediatas, que irão contribuir para alcançar o resultado da atividade.

Embora exista uma relação importante entre atividades e ações, elas são relativamente independentes, pois uma mesma ação pode participar em várias atividades. Outro ponto importante a ser considerado sobre as ações, é que elas somente podem ser compreendidas no contexto da atividade a que pertencem. Ações podem ser decompostas em cadeias de operações. Uma operação é uma tarefa rotineira, realizada de forma não consciente.

Além da estrutura hierárquica da atividade, outro aspecto importante relativo ao conceito de atividade na TA é a questão da mediação social. De uma forma geral, toda atividade humana está dentro de um contexto social, e como tal envolve mediações sociais.

Uma pessoa (sujeito) ao realizar uma atividade faz uso de artefatos, que podem ser materiais ou conceituais. Os artefatos ajudam a pessoa a transformar o objeto (que também pode ser físico ou abstrato) no resultado pretendido. No entanto, geralmente as pessoas realizam atividades dentro de uma comunidade à qual elas pertencem no momento da execução da atividade, e portanto, formas de mediações sociais devem existir para que um mínimo de ordem seja mantida dentro da comunidade. Estas formas de mediações são estabelecidas através de regras e divisão do trabalho. A *Figura 5* apresenta o diagrama de Engeström, que modela os principais componentes envolvidos na atividade dentro da perspectiva das mediações sociais.

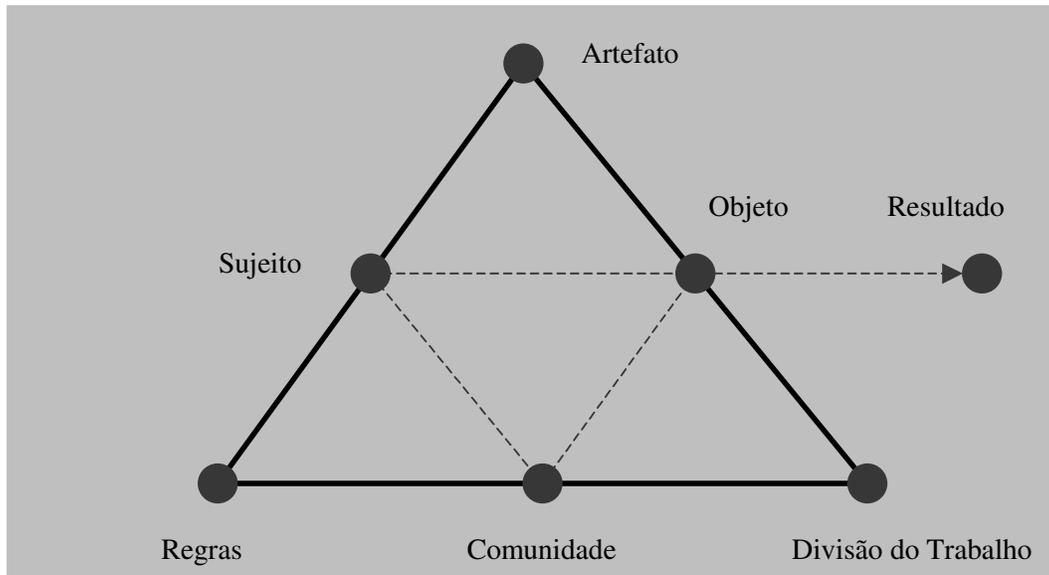


Figura 5 – O diagrama de Engeström (COLE e ENGSTRÖM, 1993).

No diagrama de Engeström, sujeito, comunidade e objeto são representados como elementos presentes numa atividade (estabelecendo o contexto da mesma), mediados por artefato, regras e divisão do trabalho. A relação entre sujeito e objeto é sempre mediada por um artefato. A relação entre sujeito e comunidade é mediada por um conjunto de regras, e a relação entre comunidade e objeto é mediada pela divisão do trabalho.

Segundo Engeström, as regras são “as normas e as sanções que especificam e regulam os procedimentos corretos esperados e as interações aceitáveis entre os participantes” e a divisão do trabalho é a “distribuição de tarefas, poderes e responsabilidades entre os participantes do sistema da atividade” (COLE e ENGSTRÖM, 1993).

4. Atividade Versus Caso de Uso

Pela natureza funcional dos conceitos de atividade e caso de uso expostos nas seções anteriores, nos parece intuitivo a existência de uma forte relação entre eles. Para uma melhor investigação da relação entre atividade e caso de uso, apresentamos na *Figura 6* um modelo que destaca o relacionamento de uma atividade com os seus componentes. A notação utilizada para a representação do modelo é a mesma que foi utilizada por Sutcliffe para representar o meta-esquema do caso de uso (*Figura 3*), esta opção foi feita apenas para facilitar a comparação entre os dois modelos.

Uma atividade se relaciona diretamente com motivo, sujeito, objeto e ação, que são os elementos básicos para se compreender uma atividade. Os demais elementos, que não aparecem se relacionando diretamente com a atividade, são importantes para entender o contexto em que ela ocorre.

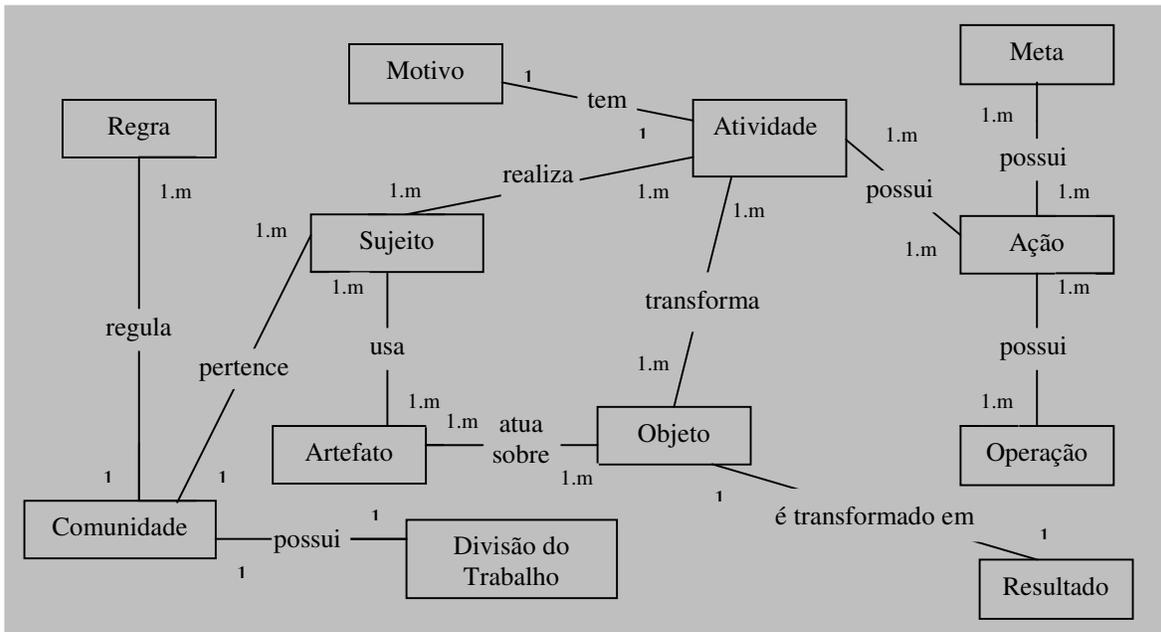


Figura 6 – Os componentes de uma atividade e seus relacionamentos.

A partir da comparação feita entre os modelos apresentados nas *Figuras 3 e 6*, e dos conceitos de caso de uso e atividade discutidos nas *seções 2 e 3*, propomos um mapeamento entre atividade e caso de uso, conforme mostrado na *Tabela 1*.

Tabela 1 – Mapeamento entre os conceitos de atividade e caso de uso.

Atividade	Caso de Uso
Ação	Ação
Sujeito	Agente
Objeto	Objeto
Motivo	-
Resultado	Meta
Artefato	-
Meta (da ação)	-
Operação	-
Regra	Pré-condição, Pós-condição ou Exceção²⁹
Comunidade	-
Divisão do Trabalho	-
-	Cenário
-	Evento
-	Propriedade
-	Estado
-	Estrutura do Objeto

²⁹ Embora os conceitos de pré-condição, pós-condição e exceção não apareçam no meta-esquema de Sutcliffe, eles foram considerados no mapeamento, pois aparecem como elementos importantes para a descrição de casos de uso, conforme descrito em (RUMBAUGH, 1997).

A *Tabela 1* mostra que dos 11 elementos relacionados ao conceito de atividade e dos 10 elementos relacionados ao conceito de caso de uso, 5 foram mapeados como equivalentes (ação - ação, sujeito - agente, objeto - objeto, resultado - meta, regra - pré-condição/pós-condição/exceção).

Os 6 elementos restantes relativos à atividade (motivo, artefato, meta da ação, operação, comunidade e divisão do trabalho) não encontram um equivalente direto entre os elementos considerados na modelagem de requisitos baseada em casos de uso. Como estes elementos auxiliam para uma melhor compreensão do contexto em que a atividade ocorre, e que atividade e caso de uso são conceitos de mesma natureza, defendemos a idéia de que a inclusão destes elementos na modelagem de requisitos baseada em casos de uso trará benefícios para a concepção de um sistema.

Considerando o caso de uso “Processar Consulta” descrito na *Figura 2*, poderíamos adicionar novas informações à sua descrição e melhorar o entendimento do sistema através da identificação dos motivos para a sua realização, da comunidade afetada e da divisão do trabalho entre os atores envolvidos.

Por exemplo, num contexto de gerenciamento de uma empresa, o *motivo* para se “Processar Consulta” num banco de dados poderia ser a tomada de decisões estratégicas a médio e longo prazo. A *comunidade* seria formada pelas pessoas (dentro e fora da empresa) que seriam afetadas pela tomada das decisões e a *divisão do trabalho* seriam as responsabilidades atribuídas aos gerentes e diretores envolvidos nas tomadas de decisões. Entendemos que elementos como *motivo, comunidade e divisão do trabalho* (que são os mais significativos, para este exemplo, dentre os 6 elementos citados) auxiliam a expandir o entendimento sobre o sistema, e dessa forma contribuem para uma melhor compreensão e definição dos requisitos do mesmo.

Conclusão

A comparação feita entre atividade e caso de uso revela que vários elementos presentes na abordagem de casos de uso encontram equivalentes no modelo de atividade, o que reforça a idéia de similaridade entre esses dois conceitos.

Por outro lado, também existem vários elementos presentes, tanto no modelo de atividade como na abordagem de casos de uso, que não encontram uma correspondência direta entre si. Como o enfoque deste trabalho é a contribuição que a Teoria da Atividade pode dar para a Engenharia de Requisitos baseada em casos de uso, e não vice-versa, apenas os elementos relacionados à atividade que não encontraram correspondência direta na abordagem de casos de uso foram analisados.

O exemplo trabalhado na *Seção 4*, salientou a contribuição dos elementos *motivo, comunidade e divisão do trabalho*, que têm uma ênfase para a expansão da análise do

contexto em que um caso de uso está inserido. No entanto, os elementos *artefato*, *metas (da ação)* e *operação* também podem oferecer contribuições para o processo.

Em casos de uso onde as ferramentas, ou instrumentos, utilizados para a sua execução assumirem uma papel importante, o conceito de artefato advindo da TA pode auxiliar o engenheiro de requisitos a definir melhor a sua funcionalidade. Sistemas onde a qualidade da interface com o usuário assume um papel crítico para o sucesso, o conceito de artefato, enquanto uma forma de mediação entre usuário e sistema, pode trazer significativos benefícios. Trabalhos na área de IHC³⁰ têm explorado o conceito de artefato conforme exposto na TA (NARDI, 1996).

As metas das ações, conforme prescrito na TA, devem ser bem definidas e imediatas, de forma a se identificar claramente a sua relação com a atividade analisada. Transpondo o conceito de metas para a abordagem de casos de uso³¹, cada ação deverá ter suas metas definidas, facilitando a descrição da funcionalidade dos casos de uso.

O conceito de operação poder trazer um nível a mais de decomposição para o caso de uso, além do nível da ação, levando a um maior detalhamento da funcionalidade do caso de uso.

Trabalhos futuros devem aprofundar e delinear melhor a similaridade existente entre atividade e caso de uso. A riqueza de elementos relacionados ao conceito de atividade, principalmente aqueles relacionados com a questão da mediação social quando uma atividade é realizada, nos trazem claros indicativos do potencial de contribuição da TA para a melhoria dos processos de Engenharia de Requisitos baseados em caso de uso.

Referências Bibliográficas

- BURD, Leo. *Desenvolvimento de Software para Atividades Educacionais*. Dissertação de Mestrado, FEEC-UNICAMP, 1999.
- COLE, Michael e ENGESTRÖM, Yrjö. *A Cultural-Historical Approach to Distributed Cognition*. In: SOLOMON, Gavriel, Ed. *Distributed Cognitions: Psychological and Educational Considerations*. Cambridge University Press, 1993.
- ERIKSSON, Hans-Erik e PENKER, Magnus. *UML Toolkit*. Wiley Computer Publishing, 1998.
- FAULK, Stuart R. *Software Requirements: A Tutorial*. In: *Software Requirements Engineering*. IEEE-CS Press, 1997.

³⁰ Interação Humano-Computador

³¹ Em (SUTCLIFFE et al., 1998) o conceito de meta é trabalhado, mas num nível mais abstrato do que no modelo de atividade, pois a meta é definida para o caso de uso, e não para suas ações, tendo uma correspondência com o conceito de resultado no modelo de atividade.

- JACOBSON, Ivar et al. *Objected-Oriented Software Engineering*. Addison-Wesley, 1992.
- JACOBSON, Ivar et al. *The Unified Software Development Process*. Addison-Wesley, 1998.
- LEITE, J.C.S.P. et al., *Enhancing a Requirements Baseline with Scenarios*, Requirements Engineering Journal: Springer-Verlag London Limited Vol. 2, N. 4 pp 184-198, 1997.
- MARTINS, Luiz E.G.; DALTRINI, Beatriz M. *An Approach to Software Requirements Elicitation Using the Precepts from Activity Theory*. Proceedings of the Fourteenth IEEE International Conference on Automated Software Engineering, Florida, 1999.
- NARDI, Bonnie A. *Context and Consciousness: Activity Theory and Human-Computer Interaction*. MIT Press, 1996.
- RUMBAUGH, James. *Getting Started: Using Use Cases to Capture Requirements*. In: *Software Requirements Engineering*, IEEE-CS Press, 1997.
- SUTCLIFFE, Alistair et al. *Supporting Scenario-Based Requirements Engineering*. IEEE Transactions on Software Engineering, Vol. 24, No. 12, December/1998.
- RATIONAL SOFTWARE et al. *UML Notation Guide*. Version 1.1, September/1997.
- WERTSCH, James V. *The Concept of Activity in Soviet Psychology: An Introduction*. In: *The Concept of Activity in Soviet Psychology*. Armonk, M. E. Sharpe, 1981. p.3-36