

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Um Modelo de Estruturação de Requisitos  
para o Método Fusion**

por

GIOVANNI ELY ROCCO

Dissertação submetida à avaliação,  
como requisito parcial para a obtenção do grau de  
Mestre em Ciência da Computação

Prof<sup>a</sup>. Dr<sup>a</sup>. Ana Maria de Alencar Price  
Orientadora

Porto Alegre, novembro de 2001.

## CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Rocco, Giovanni Ely

Um modelo de estruturação de requisitos para o Método Fusion / por Giovanni Ely Rocco - Porto Alegre: PPGC da UFRGS, 2001.

121f. : il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2001. Orientadora: Price, Ana Maria de Alencar.

1. Engenharia de Requisitos. 2. Modelo de Requisitos. 3. Engenharia de Requisitos Orientada a Objetivos. 4. Elicitação de Requisitos Baseada em Cenários. 5. Teoria da Atividade. 6. Método Fusion. I. Price, Ana Maria de Alencar. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profª. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## **Agradecimentos**

A minha orientadora, Prof<sup>a</sup>. Ana Maria de Alencar Price.

Aos meus colegas de projeto, Maurício Floriano Galimberti, José Eduardo Carvalho Bussmann e Fabrício Lazzari, pelas inestimáveis colaborações na avaliação da proposta.

Ao Serviço Autônomo Municipal de Água e Esgoto, na pessoa do diretor geral, Sr. Leonir José Taufé, pela oportunidade proporcionada para o desenvolvimento dos estudos de casos.

A Sra. Maria de Loudes Pasquali Guerra, que gentilmente proporcionou e muito atenciosamente colaborou na definição das necessidades para o desenvolvimento de um dos estudos de casos apresentados.

## Sumário

<b>Lista de Abreviaturas</b> .....	6
<b>Lista de Figuras</b> .....	7
<b>Resumo</b> .....	10
<b>Abstract</b> .....	11
<b>1 Introdução</b> .....	12
<b>2 Conceitos da Engenharia de Requisitos</b> .....	15
<b>2.1 Atividades da Engenharia de Requisitos</b> .....	17
<b>3 Objetivos</b> .....	19
<b>4 Argumentação</b> .....	21
<b>4.1 Abordagens Orientadas a Objetivos</b> .....	21
<b>4.2 Teoria da Atividade</b> .....	28
<b>5 Modelo de Requisitos</b> .....	32
<b>5.1 Os Objetivos</b> .....	32
5.1.1 Objeto .....	34
5.1.2 Sujeito .....	34
5.1.3 Ferramenta .....	35
5.1.4 Glossário de Definição de Termos .....	36
<b>5.2 A Estrutura Hierárquica</b> .....	36
<b>5.3 Os Requisitos</b> .....	40
5.3.1 Ação .....	42
5.3.2 Agente .....	42
5.3.3 Produto .....	43
5.3.4 Recurso .....	43
5.3.5 Anotação .....	44
5.3.6 Um exemplo .....	44
<b>5.4 Os Cenários</b> .....	48
<b>6 Rastreabilidade</b> .....	53
<b>6.1 Casos de Uso</b> .....	54
<b>6.2 Modelos do Método Fusion</b> .....	57
6.2.1 Modelo de Objetos .....	57
6.2.2 Modelo Ciclo-de-vida .....	61
6.2.3 Modelo de Operações .....	63
6.2.4 Usabilidade .....	65
<b>6.3 Ferramenta de Suporte ao Modelo</b> .....	66
<b>7 Estudos de Casos</b> .....	72
<b>7.1 Métodos de Elicitação de Requisitos</b> .....	72
7.1.1 Técnicas de Entrevistas .....	73
7.1.2 Técnicas de Reuniões .....	73
7.1.3 Técnicas de Observações .....	74
7.1.4 Abordagem Etnográfica .....	75
7.1.5 <i>Inquiry Cycle Model</i> .....	76
<b>7.2 Os Estudos de Casos</b> .....	77
7.2.1 Caso: Controle de Serviços Operacionais .....	77
7.2.2 Caso: Sistema de Protocolo .....	77
7.2.3 Caso: Controle de Imóveis .....	80

7.2.4 Avaliação da Aplicação do Modelo de Estruturação de Requisitos .....	101
<b>8 Conclusão</b> .....	102
<b>Anexo 1 Estudo de Caso: Sistema de Protocolo</b> .....	105
<b>Bibliografia</b> .....	117

## Lista de Abreviaturas

CRC	Class-Responsability-Collaboration
CSM	Current State Model
FILM	Método Fusion Expandido e Adaptado a UML
HCI	Human Computer Interface
IEEE	Institute of Electrical and Eletronical Engineers
JAD	Joint Application Design
OMT	Object Modeling Technique
OOSE	Object-Oriented Software Engineering
OSI	Open System Interconnection
PD	Participatory Design
RWE	Real World Example
RWE	Real World Example Fragment
RWS	Real World Scenes
UML	Unified Modeling Language

## Lista de Figuras

FIGURA 2.1 - Domínios da Engenharia de Requisitos .....	16
FIGURA 2.2 - Processo da Engenharia de Requisitos .....	18
FIGURA 4.1 - Processo de inferência de objetivos [LAM98] .....	23
FIGURA 4.2 - Processo de definição de objetivos, com sistema existente [HAU98] .	24
FIGURA 4.3 - Modelo hierárquico de objetivos e fragmentos de exemplos [HAU98]	25
FIGURA 4.4 - Associação bidirecional entre objetivos e cenários [ROL98] .....	25
FIGURA 4.5 - <i>Requirement chunk</i> [ROL98] .....	26
FIGURA 4.6 - Visão geral do processo de descoberta de objetivos [ROL98] .....	27
FIGURA 4.7 - Elementos básicos de uma atividade .....	30
FIGURA 4.8 - Níveis hierárquicos de uma atividade .....	30
FIGURA 5.1 - Definição de um objetivo .....	33
FIGURA 5.2 - Composição do objetivo exemplo do setor de controle operacional .	35
FIGURA 5.3 - Estrutura para definição de objetivo, no processo de elicitação .....	36
FIGURA 5.4 - Glossário de termos do objetivo do setor de controle operacional .....	37
FIGURA 5.5 - Estrutura hierárquica dos requisitos .....	38
FIGURA 5.6 - Hierarquia dos requisitos do setor de controle operacional .....	40
FIGURA 5.7 - Exemplo de hierarquia de requisitos alternativos .....	41
FIGURA 5.8 - Definição de um requisito .....	42
FIGURA 5.9 - Representação do requisito "coleta de informações diárias" .....	45
FIGURA 5.10 - Representação do requisito "emissão de ordem de serviço" .....	46
FIGURA 5.11 - Representação do requisito "informação diária da equipe" .....	47
FIGURA 5.12 - Hierarquia dos requisitos com cenários .....	49
FIGURA 5.13 - Representação de um cenários .....	49
FIGURA 5.14 - Cenário do requisito "coleta de informações diárias" .....	50
FIGURA 6.1 - Casos de uso do requisito "coleta de informações diárias" .....	55
FIGURA 6.2 - Casos de uso do requisito "cálculo de custo total da obra" .....	56
FIGURA 6.3 - Diagrama de casos de uso do requisito "coleta informações diárias" .....	56
FIGURA 6.4 - O Método Fusion [COL96] .....	58
FIGURA 6.5 - Reprodução do requisito "emissão de ordem de serviço" .....	59
FIGURA 6.6 - Reprodução do requisito "informação diária da equipe" .....	59
FIGURA 6.7 - Modelo de objetos do setor de controle operacional .....	60
FIGURA 6.8 - Estrutura hierárquica dos requisitos do setor controle operacional ...	62
FIGURA 6.9 - Modelo ciclo-de-vida do setor de controle operacional .....	62
FIGURA 6.10 - Reprodução do cenário do requisito "coleta de informações diárias" .....	64
FIGURA 6.11 - Esquema da operação <i>EmissãoOrdemServiço</i> .....	65
FIGURA 6.12 - Esquema da operação <i>InformaçãoDiáriaEquipe</i> .....	65
FIGURA 6.13 - Modelo de estruturação de requisitos .....	66
FIGURA 6.14 - Objetivo de especificação do modelo de estruturação de requisitos ..	67
FIGURA 6.15 - Objetivo de geração dos diagramas de caso de uso .....	67
FIGURA 6.16 - Estrutura hierárquica dos requisitos da ferramenta de suporte .....	68
FIGURA 6.17 - Requisito "definir objetivo" .....	68
FIGURA 6.18 - Requisito "definir estrutura do objetivo" .....	68
FIGURA 6.19 - Requisito "definir requisito" .....	69
FIGURA 6.20 - Requisito "definir cenário" .....	69
FIGURA 6.21 - Requisito "definir glossário de termos" .....	69
FIGURA 6.22 - Requisito "definir casos de uso" .....	69
FIGURA 6.23 - Requisito "definir relacionamento entre casos de uso" .....	70

FIGURA 6.24 - Requisito "definir atores associados aos casos de uso" .....	70
FIGURA 6.25 - Macro cenário da ferramenta de suporte ao modelo proposto .....	71
FIGURA 7.1 - Fases do <i>Inquiry Cycle Model</i> .....	76
FIGURA 7.2 - Estrutura de objetivos e requisitos do sistema de protocolo .....	79
FIGURA 7.3 - Definição do objetivo de cadastramento dos imóveis .....	81
FIGURA 7.4 - Definição do objetivo de manutenção da situação dos imóveis .....	81
FIGURA 7.5 - Definição do objetivo de controle da situação dos imóveis .....	81
FIGURA 7.6 - Estrutura dos requisitos do objetivo de cadastramento de imóveis ...	82
FIGURA 7.7 - Requisito "cadastrar proprietário" .....	83
FIGURA 7.8 - Requisito "cadastrar loteamento" .....	83
FIGURA 7.9 - Requisito "registrar imóvel" .....	83
FIGURA 7.10 - Requisito "cadastrar lote" .....	83
FIGURA 7.11 - Requisito "cadastrar edificação" .....	84
FIGURA 7.12 - Requisito "informar contrato de locação" .....	84
FIGURA 7.13 - Requisito "cadastrar imobiliária" .....	84
FIGURA 7.14 - Requisito "cadastrar inquilino" .....	84
FIGURA 7.15 - Cenário de cadastrar lote livre .....	85
FIGURA 7.16 - Cenário de cadastrar edificação livre .....	85
FIGURA 7.17 - Cenário de cadastrar lote com edificação .....	86
FIGURA 7.18 - Cenário de cadastrar edificação locada .....	86
FIGURA 7.19 - Estrutura dos requisitos do objetivo de manutenção da situação .....	87
FIGURA 7.20 - Requisito "atualizar situação judicial do imóvel" .....	87
FIGURA 7.21 - Requisito "informar dados do processo judicial" .....	88
FIGURA 7.22 - Requisito "informar andamento do processo" .....	88
FIGURA 7.23 - Requisito "vender imóvel" .....	88
FIGURA 7.24 - Cenário de informar processo judicial do imóvel .....	89
FIGURA 7.25 - Cenário de atualizar situação judicial do imóvel .....	89
FIGURA 7.26 - Cenário de vender imóvel .....	89
FIGURA 7.27 - Estrutura dos requisitos do objetivo de controle da situação .....	90
FIGURA 7.28 - Requisito "consultar imóvel" .....	91
FIGURA 7.29 - Requisito "consultar imóvel por inscrição" .....	91
FIGURA 7.30 - Requisito "consultar imóvel por processo" .....	91
FIGURA 7.31 - Cenário de consulta imóvel por inscrição .....	92
FIGURA 7.32 - Cenário de consulta imóvel por processo .....	92
FIGURA 7.33 - Glossário de termos dos requisitos do controle de imóveis .....	93
FIGURA 7.34 - Casos de uso de cadastramento de imóveis .....	94
FIGURA 7.35 - Casos de uso de manutenção da situação dos imóveis .....	95
FIGURA 7.36 - Casos de uso de controle da situação dos imóveis .....	95
FIGURA 7.37 - Modelo de objetos do controle de imóveis .....	96
FIGURA 7.38 - Modelo ciclo-de-vida do controle de imóveis .....	97
FIGURA 7.39 - Operação cadastrarProprietário .....	97
FIGURA 7.40 - Operação cadastrarLoteamento .....	97
FIGURA 7.41 - Operação registrarImóvel .....	98
FIGURA 7.42 - Operação cadastrarLote .....	98
FIGURA 7.43 - Operação cadastrarEdificação .....	98
FIGURA 7.44 - Operação cadastrarImobiliária .....	98
FIGURA 7.45 - Operação cadastrarInquilino .....	99
FIGURA 7.46 - Operação informarContratoLocação .....	99
FIGURA 7.47 - Operação informarProcesso .....	99
FIGURA 7.48 - Operação informarAndamento .....	99

FIGURA 7.49 - Operação venderImóvel .....	100
FIGURA 7.50 - Operação consultarImóvelInscrição .....	100
FIGURA 7.51 - Operação consultarImóvelProcesso .....	100

## Resumo

Uma definição confiável dos requisitos de um software depende diretamente da completa e correta compreensão sobre as necessidades do sistema e sua conseqüente representação de forma adequada ao processo de desenvolvimento. Uma proposta de modelagem de requisitos deve apresentar qualidades que colaborem para a compreensão mútua das necessidades entre os envolvidos no processo e que organizem os requisitos de forma a permitir o acompanhamento no desenvolvimento do software.

O presente trabalho apresenta um modelo de estruturação de requisitos fundamentado em metodologias orientadas a objetivos com utilização de cenários e preceitos da Teoria da Atividade. O modelo tem sua argumentação nas premissas que cliente e usuários normalmente expressam suas necessidades através de objetivos almejados e que a ação humana deve ser analisada dentro de um contexto para que possa fazer sentido e ser compreendida.

Inserido no contexto do Projeto FILM<sup>1</sup>, cujo objetivo é expandir o Método Fusion, agregando uma etapa de modelagem de requisitos, o trabalho estabeleceu a qualidade de usabilidade como motivadora da definição de um modelo de estruturação de requisitos. A usabilidade é uma qualidade que visa facilitar a utilização do modelo como uma ferramenta de representação dos requisitos de forma inteligível, atuando tanto na especificação dos requisitos como na validação dos mesmos entre os envolvidos.

Os requisitos são estruturados segundo uma abordagem voltada aos clientes e usuários do sistema. O modelo definido tem por objetivo prover a construção gradual e incremental do entendimento compartilhado entre os envolvidos sobre os domínios do problema e da solução, na concepção e no desenvolvimento do software. Metodologias orientadas a objetivos, operacionalizadas através de cenários, conjugadas a princípios da atividade oferecem um suporte adequado a estruturação de requisitos provendo usabilidade ao modelo.

A avaliação da aplicabilidade do modelo é realizada com a modelagem de requisitos em três estudos de casos. Em cada caso são aplicadas técnicas de elicitação no sentido de afinar a sintonia com a estrutura do modelo de requisitos. A concepção do modelo, embasada em conceitos da Teoria da Atividade, é bastante adequado às atividades de elicitação em uma abordagem voltada aos clientes e usuários.

**Palavras-Chaves:** Engenharia de Requisitos, Modelo de Requisitos, Engenharia de Requisitos Orientada a Objetivos, Elicitação de Requisitos Baseada em Cenários, Teoria da Atividade, Método Fusion.

---

<sup>1</sup> Método Fusion Expandido e Adaptado à UML é um projeto institucional financiado e desenvolvido pela Universidade de Caxias do Sul cujo objetivo é adaptar os modelos do Fusion às notações da *Unified Modeling Language* e expandir o método prevendo a elicitação e modelagem de requisitos.

**TITLE: “A MODEL TO REQUIREMENTS STRUCTURING FOR THE FUSION METHOD”**

## **Abstract**

A reliable definition of software requirements depends directly on the complete and correct understanding of the system’s needs and its consequent proper representation in the development process. A proposal for requirements modeling should present qualities that collaborate for the mutual understanding of the needs among everyone involved in the process and that organize the requirements to allow anyone to follow the software development process.

This work presents a model to requirements structuring based in goal-oriented methodologies by using scenarios and precepts of the Activity Theory. The model has its conception in the premises that customer and users often express their needs as desired goals and that human action should be analyzed in context to make sense and be understood.

Inserted in the FILM<sup>1</sup>'s Project context, whose objective is to expand the Fusion Method by incrementing a stage of requirements modeling, this work had established quality of usability as motivation for a model of requirements structuring. Usability aims to make it easier to employ the model as a tool for requirements representation in an intelligible way, by playing a role in requirements specification and also in their validation among the people involved.

Requirements are structured according to an approach directed towards clients and users of the system. The model’s goal is to provide a gradual and incremental construction of the understanding shared among everyone involved in the system’s definition and development about the problem domain and its solution domain. Goal-oriented methodologies, instrumented through scenarios, and coupled with activity principles offer an appropriate support to requirements structuring and also add usability to the model.

The model’s applicability had been evaluated through the modeling of requirements of three case studies. For each case study, elicitation techniques had been applied to adjust the syntony with the structure of the requirements model. The conception of the model is based in concepts of Activity Theory, and has been shown as an appropriate way to conduct elicitation activities in a clients and users oriented approach.

**Keywords:** Requirements Engineering, Requirements Model, Goal-Oriented Requirements Engineering, Scenario-Based Requirements Elicitation, Activity Theory, Fusion Method.

---

<sup>1</sup> Método Fusion Expandido e Adaptado à UML is a institutional project financed and developed at the Caxias do Sul University, whose objective is to adapt the Fusion model to the notations of Unified Modeling Language and to expand the method foreseeing the elicitation and modeling of requirements.

# 1 Introdução

*“A parte mais difícil na construção de um sistema de software é decidir precisamente o que construir. Nenhuma outra parte do trabalho conceitual é tão difícil quanto estabelecer os requisitos técnicos detalhados, incluindo todas as interfaces com pessoas, com máquinas e com outros sistemas. Nenhuma outra parte do trabalho inutiliza o sistema resultante se for feita de forma errada. Nenhuma outra parte é mais difícil de retificar mais tarde.”*

*Fred Brooks [BRO87]*

O processo de desenvolvimento de software caracteriza-se por um conjunto de atividades cujo objetivo é produzir software que atenda às necessidades do usuário. Independentemente do paradigma de engenharia de software escolhido, esse processo conta com três fases genéricas: a definição, o desenvolvimento e a manutenção [PRE95]. O foco principal da fase de definição concentra-se na especificação dos requisitos, que são propriedades que um sistema deve ter em busca do seu êxito no ambiente onde será utilizado.

É incontestável a importância da definição de requisitos, considerando-se que o sucesso no desenvolvimento depende diretamente da fidelidade com que o software atende às necessidades previstas. Uma definição precisa do que produzir é condição determinante para abonar a satisfação do cliente e do usuário. Porém, uma das maiores dificuldades da engenharia de requisitos reside na complexidade da própria atividade. Almeja-se uma compreensão completa e correta das necessidades, em atividades que participam todos os *stakeholders*<sup>NT</sup>.

Elicitar e modelar os requisitos não são tarefas fáceis, pois os *stakeholders*, normalmente, possuem conhecimentos, experiências, preconceitos e terminologias diferentes. Um dos mais notáveis reflexos dessas diferenças culturais é a dificuldade de comunicação entre eles, que acaba tonando-se um dos principais obstáculos à realização satisfatória da definição dos requisitos. Aspectos de comunicação e a efetiva participação dos envolvidos no processo são eminentemente decisivos para a qualidade e eficácia do resultado e têm sido tratados por metodologias oriundas das ciências sociais.

O estabelecimento de um processo de definição de requisitos é necessário tanto para os desenvolvedores como para os clientes e usuários, objetivando construir gradualmente e interativamente um entendimento compartilhado sobre a especificação. Nesta direção, metodologias incentivam o engajamento prematuro dos usuários no desenvolvimento do software, contando com a sua participação efetiva no sentido de trocar perspectivas e negociar requisitos do sistema. As metodologias que seguem esse paradigma visam reduzir essa lacuna semântica entre os desenvolvedores e os clientes e usuários. Estabelecem métodos que facilitam a descoberta dos objetivos do sistema, definindo técnicas de representação de requisitos que auxiliam a sua abordagem, buscando prover uma conversação eficaz entre os *stakeholders* com um grau de abstração adequado à atividade.

A definição de um método de estruturação e especificação de requisitos, segundo essa abordagem participativa, agrega um rigor conceitual ao software, dado que a qualidade do produto final é intrinsecamente dependente da qualidade da

---

<sup>NT</sup> São todos os envolvidos no universo de informações, que compartilham as informações do domínio. Compreende os especialistas do domínio, os clientes, os usuários, os desenvolvedores e os legisladores.

definição das necessidades. Este trabalho têm por objetivo apresentar um modelo de estruturação de requisitos, visando a especificação segundo uma metodologia que prima pela abordagem de facilitar o tratamento das necessidades. Essa abordagem, voltada à participação do envolvidos, busca cooperar diretamente na compreensão das necessidades, na comunicação entre os *stakeholders*, e no acompanhamento do desenvolvimento dos requisitos.

O presente estudo está inserido no Projeto FILM. O Projeto FILM, Método Fusion Expandido e Adaptado a UML, é um projeto institucional financiado e desenvolvido pela Universidade de Caxias do Sul, cujo objetivo é adaptar os modelos do Método Fusion às notações da *Unified Modeling Language* e expandir o método prevendo a elicitação e modelagem de requisitos.

O Método Fusion [COL96] é um método sistemático de desenvolvimento de software orientado a objetos, que estabelece procedimentos que guiam o processo desde a definição dos requisitos até a implementação do sistema. Foi idealizado buscando prover facilidade no desenvolvimento, manutenção e estruturação do sistema, como também auxiliar no gerenciamento de projetos e permitir suporte efetivo de ferramentas. Integra aspectos positivos de diferentes métodos orientados a objetos: *Método de Booch* [BOO94], *Object Modeling Technique (OMT)* [RUM94], *Object-Oriented Software Engineering (OOSE)* [JAC92] e *Class-Responsability-Collaboration (CRC)*.

O Método Fusion fornece um processo de desenvolvimento bastante conciso, propondo notações completas, simples e bem definidas, e ferramentas de gerenciamento que incluem uma série de verificações, as quais visam assegurar a consistência do processo.

A motivação do Projeto FILM reside no aproveitamento do método sistemático oferecido pelo Fusion para adaptar seus modelos à *Unified Modeling Language* (UML). A UML é uma linguagem de modelagem flexível e adaptável a metodologias de desenvolvimento de software, reconhecida pelo *Object Management Group*<sup>OMG</sup> como padrão exclusivo de notação para modelos orientados a objetos [FOW2000]. Não se caracteriza por propor um método de desenvolvimento, mas um conjunto de notações para modelagem orientada a objetos de forma a flexibilizar a adequação de métodos existentes, como, por exemplo, o Método Fusion.

Salienta-se que o Método Fusion alicerça-se nas abordagens de Booch, da OMT e do OOSE, as quais foram desenvolvidas pelos mesmos autores da UML: Grady Booch, James Rumbaugh e Ivar Jacobson. O Método Fusion reflete um sentimento de sintonia entre o método e a linguagem de modelagem, esta última apresentando-se como uma notação eficaz e padronizada.

No entanto, o método é aplicado a partir de uma descrição de requisitos, cuja metodologia de captura e notação não é abordada no espectro de sua atuação. O Método Fusion parte do princípio de que já exista um documento de requisitos: "vale lembrar que a captura de requisitos será normalmente executada por um usuário, o qual deve fornecer o documento inicial de requisitos; portanto, o Fusion não possui uma fase de captura de requisitos" [COL96].

O presente trabalho está inserido neste contexto. O objetivo é propor um modelo para a representação dos requisitos do software, adequado ao Método Fusion. A

---

<sup>OMG</sup> Object Management Group é uma organização que representa empresas e entidades governamentais e acadêmicas ligadas ao desenvolvimento de software que tem por objetivo estabelecer padrões e especificações de gerenciamento de objetos a fim de prover modelos comuns para o desenvolvimento de aplicações. Disponível em URL: <http://www.omg.org> (1999).

proposta de adaptação do método à notação é assunto de trabalho correlato, conforme diretrizes do Projeto FILM [GAL2000].

Inicialmente, na Seção 2, são definidos alguns conceitos da Engenharia de Requisitos que são trabalhados na proposta. Na Seção 3, são apresentados os objetivos a serem alcançados. Na Seção 4, são citadas pesquisas da área de engenharia de requisitos que contribuíram para a definição e argumentação do modelo proposto. O embasamento da proposta conjuga metodologias orientadas a objetivos com a utilização de cenários, tendo o processo de elicitação e modelagem fundamentado na Teoria da Atividade. Nas Seções 5 e 6, é descrito o modelo de estruturação proposto e a sua correspondência com os modelos do Método Fusion, conforme objetivos do Projeto FILM. Estudos de caso são apresentados na Seção 7, sobre os quais é avaliada a proposta. Por fim, o trabalho é analisado quanto ao aspecto de atendimento do modelo proposto aos objetivos estabelecidos.

## 2 Conceitos da Engenharia de Requisitos

*The IEEE Standard Glossary of Software Engineering Terminology* [IEEE97] define requisito como uma condição ou capacidade necessária para um usuário resolver um problema ou alcançar um objetivo (...).

Desta definição pode-se asserir que os requisitos de software são derivados de necessidades que usuários têm ao resolver algum problema. Situa-se, assim, o domínio do problema. O domínio do problema é uma parte particular do sistema real na qual o usuário está interessado, isto é, a parte relevante para seu problema específico [JAI95].

Há consenso que a especificação de requisitos deve incluir não somente as especificações do domínio do problema, mas também qualquer tipo de informação que descreva o contexto do sistema [ALE99]. Esse contexto é conhecido como universo de informações, que é a realidade circunstanciada pelo conjunto de objetivos definidos pelos que demandam o software, e inclui todas as fontes de informação e todos os *stakeholders* [LEI94].

O domínio do problema é o domínio de atuação do software e inclui todos os elementos que interagem com ele. O universo de informações, entretanto, é mais abrangente. É o contexto do sistema real, no qual encontram-se elementos e ocorrem fenômenos que provêm subsídios ao software e onde sentem-se seus efeitos e benefícios.

Uma analogia com um projeto de uma ponte sobre um rio pode ser considerada para ilustrar os conceitos. O domínio do problema restringe-se às especificações da ponte em si, como o objetivo em tê-la construída, o local e traçado desejado, o desenho arquitetônico, as especificações estruturais e de materiais, as especificações das fundações e do projeto necessário para sua construção. Já o universo de informações compreende todos as informações que influenciam a especificação, como o leito do rio, as margens, o curso d'água, as condições climáticas e geológicas e todos os outros fatores que vão interferir no projeto, e, também, todo o contexto que sofrerá seus efeitos, como a população e mercado beneficiados.

O domínio do problema é, portanto, o ambiente de atuação do software. Neste ambiente principiam as atividades da engenharia de software, a definição das necessidades do software. É tarefa dos engenheiros de requisitos entender o problema, na cultura e linguagem dos usuários, e definir um sistema que atenda às suas necessidades [LEF2000]. Para tal, o engenheiro de requisitos deve descobrir e estabelecer o universo de informações, de onde obtém os recursos na tarefa de elucidação do problema.

Do contexto do problema parte-se para a abordagem da solução. No domínio da solução é enfocada a definição da solução aos problemas dos usuários. Neste espaço, os desenvolvedores do software aplicam seus conhecimentos em busca da especificação do sistema a ser desenvolvido. Envolve as características da solução e os requisitos do sistema (figura 2.1). As características são descrições simples, na linguagem dos usuários, cujo objetivo é a comunicação entre os desenvolvedores e usuários sobre como o sistema trata o problema. Um característica é definida como um serviço que o sistema provê para cumprir uma ou mais necessidades dos *stakeholders* [LEF2000].

Uma vez estabelecido o conjunto de características com a concordância dos *stakeholders*, deve-se definir os requisitos mais específicos que serão necessários impor à solução. Requisitos são propriedades que um sistema de software deve ter em

busca do seu êxito no ambiente onde será utilizado [GOG94]. São especificações sobre o que deve ser desenvolvido e são condicionantes do processo da engenharia do software [SOM97].

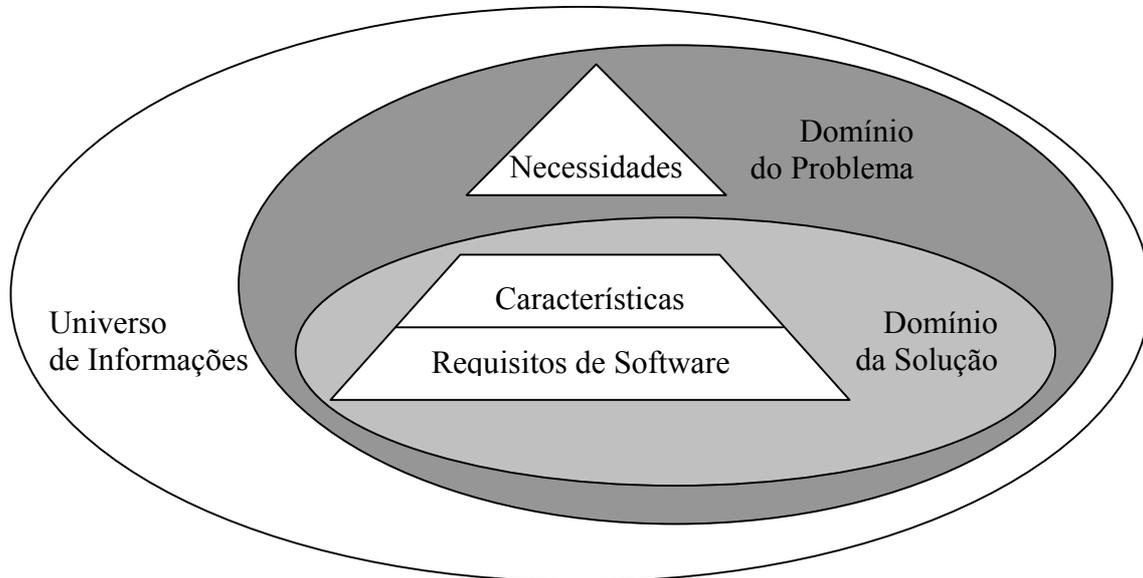


FIGURA 2.1 - Domínios da Engenharia de Requisitos

Os requisitos de software incluem três níveis [WIE99]: os requisitos do negócio, os requisitos do usuário e os requisitos funcionais. Os requisitos do negócio identificam os benefícios primários que o sistema proverá aos clientes. São objetivos de alto nível ou requisições do cliente para o sistema de software.

Os requisitos do usuário são tarefas que os usuários serão habilitados a realizar com o sistema de software. Em geral, são capturados segundo abordagens de casos de uso e descrição de cenários, conforme será apresentado nos capítulos subsequentes.

Os requisitos funcionais definem a funcionalidade que o software deve prover a fim de capacitar os usuários a realizar suas tarefas, satisfazendo os requisitos do negócio. Existem taxonomias que classificam em requisitos funcionais e não funcionais conforme seus objetivos [MAC99]. Basicamente, pode-se definir os requisitos funcionais como especificações do que o software deve estar habilitado a fazer, e os requisitos não funcionais como especificações de o quão bem as funções devem ser realizadas (qualitativos), o custo empregado para criar e manter a funcionalidade e a qualidade (quantitativos), e as restrições impostas aos requisitos ou liberdade de desenvolvimento (restritivos) [GIL97]. Os requisitos de software, portanto, compreendem tanto as necessidades funcionais, que definem o comportamento e as propriedades do sistema, como as necessidades não funcionais, que consistem em requisitos de qualidade e restrições operacionais ou do desenvolvimento do software.

*The IEEE Standard Glossary of Software Engineering Terminology* [IEEE97] define, ainda, requisito como uma representação documentada da condição ou capacidade (necessária para um usuário resolver um problema ou alcançar um objetivo).

Infere-se, segundo essa definição, a essencialidade de documentar as necessidades do usuário. A representação dos requisitos é um conceito chave para a

prática da engenharia do software [WIE99], e visa descrever em modelos conceituais as condições ou capacidades do sistema. O objetivo principal da representação é propiciar uma melhor compreensão do domínio do problema e uma precisa definição do domínio da solução.

A representação dos requisitos é estabelecida em um nível conceitual de modelagem. Um modelo conceitual é uma estrutura cognitiva usada com o propósito de entendimento e comunicação de aspectos físicos e sociais do mundo que nos cerca [LOU95]. Os modelos produzidos, textuais ou em notação gráfica, descrevem os aspectos relevantes do domínio do problema e relata os componentes e comportamento esperado para a solução de software. Uma representação completa e correta é fundamental para que o desenvolvimento do software realmente atenda às necessidades dos clientes e usuários, considerando que a satisfação dos requisitos especificados é uma pré-condição básica para o sucesso de um software [CAL99].

## 2.1 Atividades da Engenharia de Requisitos

A Engenharia de Requisitos é a disciplina que procura sistematizar o processo de definição de requisitos [LEI94]. Aborda um ponto fundamental do desenvolvimento de software: a definição do que produzir. Tem sido identificada como uma fase crucial por tratar de conhecimentos não apenas técnicos, mas também gerenciais, organizacionais, econômicos e sociais [CAS95], e estar intimamente associada a qualidade do software [LEE98].

As atividades da Engenharia de Requisitos vão desde a idéia de desenvolver um sistema de software até a modelagem conceitual do que vai ser construído. Aborda questões organizacionais, como identificação dos objetivos e definição do universo de informação; gerenciais, como determinação de diretrizes e análise de impacto do sistema; econômicas, como análise de custos, previsão e recursos necessários; sociais, como negociação, participação e comunicação entre os envolvidos; e técnicas, como coleta, organização, representação, análise e validação dos requisitos.

A Engenharia de Requisitos é responsável por propor um conjunto de métodos, técnicas e ferramentas que suportam o processo de definição de requisitos. Esse processo compreende duas fases com atividades distintas, porém interrelacionadas [CAS95]: uma fase de aquisição e uma fase de especificação dos requisitos (figura 2.2).

A fase de aquisição tem como objetivo a obtenção do conhecimento do domínio do problema, junto às fontes do universo de informação, e a certificação entre os *stakeholders* de que as necessidades estão especificadas de forma correta e completa. Compreende as atividades de elicitação e validação.

A atividade de elicitação consiste na realização de ações para aquisição do conhecimento do domínio do problema. O objetivo principal é identificar e coletar informações do universo de informações, de forma a prover o mais completo e correto entendimento das necessidades que demandam o software.

A atividade de validação, no contexto da Engenharia de Requisitos, consiste na execução de ações com o objetivo de confirmar o conhecimento adquirido. Procura-se corroborar que o que foi especificado realmente representa o domínio do problema, isto é, retrata as necessidades. Sob esse ponto de vista, a validação pode ser considerada como uma atividade de negociação, na qual os clientes procuram explorar o completo entendimento dos requisitos, de acordo com o que eles querem e necessitam, enquanto que os desenvolvedores procuram a confiança em resolver o problema certo. Também, busca-se assegurar que os requisitos funcionais sejam verificáveis, almejando

garantir a correção da especificação, e que os requisitos não funcionais de qualidade, quantidade e restrições de software estejam contemplados na definição.

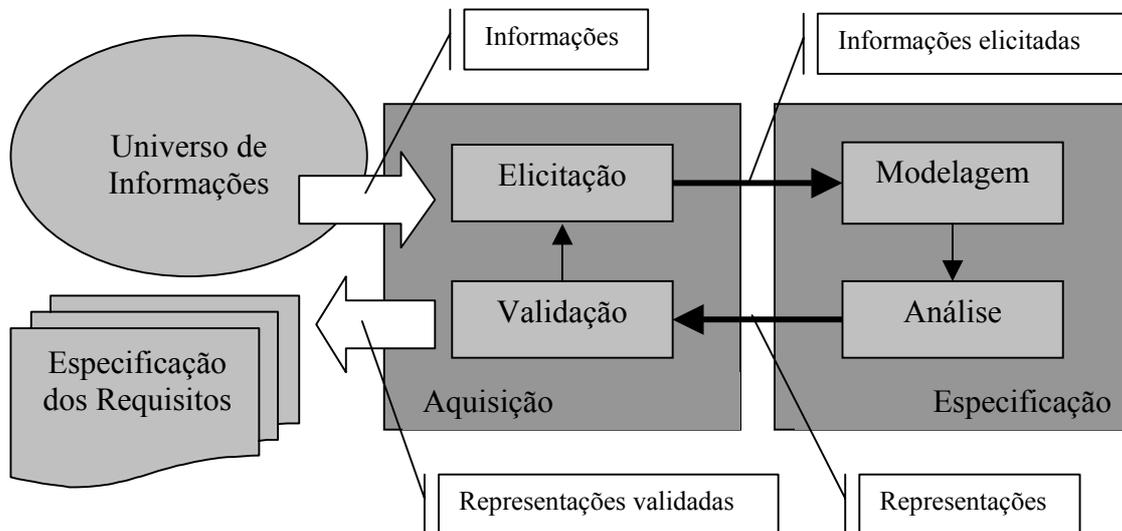


FIGURA 2.2 - Processo da Engenharia de Requisitos

A fase de especificação tem como objetivo descrever conceitualmente o conhecimento elicitado em modelos que expressem o domínio do problema, conforme o interesse dos *stakeholders*, e avaliar a descrição visando a correção, completude e consistência das necessidades. Compreende as atividades de modelagem e análise.

A atividade de modelagem visa a representação dos requisitos em modelos conceituais que descrevem as necessidades. O objetivo principal é representar o domínio problema para propiciar uma melhor compreensão das propriedades desejadas para o software. Os modelos produzidos, textuais ou em notação gráfica, descrevem os requisitos, relatando os componentes e o comportamento do software. A facilidade de leitura e a não ambigüidade são características importantes a serem consideradas nos modelos.

A atividade de análise compreende ações de examinar os modelos buscando detectar e resolver inconsistências e omissões. A meta da análise é analisar se o domínio do problema está representado e entendido coerentemente. Essa atividade antecede a atividade de validação e, ao contrário desta última que conta com a participação dos clientes, é desempenhada pelos engenheiros de requisitos.

Uma propriedade importante para uma análise bem sucedida é o rastreamento, que consiste na ligação da necessidade do domínio do problema ao requisito representado nos modelos [LEI94]. Esse rastreamento deve perdurar durante todo o ciclo de desenvolvimento, facilitando o gerenciamento do processo de produção e a validação do software. Como requisitos são produzidos não apenas na fase de aquisição, mas também durante todas as fases subseqüentes [PIN98], a introdução desta propriedade aos modelos dos requisitos agrega flexibilidade ao desenvolvimento do software e a manutenções futuras.

### 3 Objetivos

O estabelecimento de um modelo de representação de requisitos é necessário tanto para os desenvolvedores como para os usuários, com o objetivo de construir gradual e interativamente o entendimento compartilhado sobre o domínio do problema e da solução [MAT98]. A representação busca descrever conceitualmente os domínios em modelos que expressem as necessidades, conforme o interesse dos *stakeholders*, de forma a prover a mais completa e correta definição do software.

Contudo, a produção de especificações de alta qualidade não é fácil. Problemas de requisitos são custosos e incômodos a quase todos os sistemas e organizações de desenvolvimento de software [SAW99], e cobrem dificuldades desde a definição do universo de informação até o relacionamento entre os *stakeholders*. A questão fundamental é como encontrar as reais necessidades do usuário para a futura implementação do software.

Devido à complexidade dos sistemas atuais, essa tarefa pode ser difícil, haja visto que esses sistemas possuem diversos clientes e usuários, com múltiplos domínios a serem envolvidos. As dificuldades encontradas são críticas. Em um estágio inicial do desenvolvimento, os clientes geralmente não têm uma visão clara sobre o que necessitam do sistema de software e os requisitos não refletem suas reais necessidades. É notório, também, a existências de diversas barreiras que afetam a comunicação entre os envolvidos, como diferenças sociais, culturais, intelectuais e lingüísticas [DEB99]. Existe uma lacuna semântica entre os desenvolvedores e os clientes e usuários do sistema computacional [GON98].

Diante deste panorama, é comum requisitos estarem incompletos ou inconsistentes [ALE99]. Pesquisas têm comprovado que muitos projetos de implementação de software têm falhado por problemas de definição dos requisitos de software [BOE81]. Os custos de manutenção de requisitos relacionados a uma definição imprecisa crescem na ordem de magnitude do progresso do desenvolvimento do software. Estudos mostram que quanto mais avançado no ciclo de desenvolvimento é detectado um requisito errado, mais onerosa é sua correção [BOE81].

Os requisitos de software se modificam, também, ao longo do ciclo de desenvolvimento, mas o impacto da mudança varia conforme o tempo em que ela for introduzida [PRE95]. Uma alteração nos requisitos solicitada em uma fase avançada, também agrega um acréscimo no ônus do desenvolvimento do software na ordem direta do estágio de andamento do processo. Quanto mais tardia ocorrer, maior impacto acarretará, dado o avanço no estabelecimento de uma estrutura e o comprometimento de recursos para o desenvolvimento do software.

Neste contexto, dois fatores são fundamentais na proposta de um modelo de requisitos: a estruturação e representação dos requisitos segundo uma abordagem voltada ao usuário e a usabilidade do modelo, tanto como uma ferramenta de comunicação entre os *stakeholders*, quanto um meio de prover a rastreabilidade dos requisitos ao longo do processo de desenvolvimento.

O modelo conceitual, que decorre do processo de engenharia de requisitos, serve ao propósito de organizar e representar o que é demandado e estabelecer a comunicação entre desenvolvedores, clientes e usuários, visando a compreensão precisa dos requisitos do software. A rastreabilidade é uma propriedade que consiste em estabelecer um elo de ligação entre os requisitos e os produtos do desenvolvimento do software. Esse rastreamento, para frente, é estabelecido durante a fase de definição dos requisitos e perdura durante todo o ciclo de vida. É uma qualidade

que facilita o gerenciamento do processo de produção e a validação do software, contribuindo com a análise e o acompanhamento do desenvolvimento dos requisitos.

Neste trabalho, é apresentada uma proposta de modelo de estruturação e representação de requisitos que objetiva atender a comunicabilidade e a rastreabilidade como qualidades que visam facilitar a definição dos requisitos juntamente com os usuários. O modelo previsto busca complementar o Método Fusion, cuja metodologia não aborda a fase de definição de requisitos, conforme objetivos do Projeto FILM.

Resultado dessas razões, o modelo proposto tem o objetivo fundamental de representar os requisitos funcionais, estruturando-os no contexto dos domínios do problema e da solução. Pretende-se apresentar o modelo como um instrumento que visa atender à atividade de modelagem, na fase de especificação do processo de engenharia de requisitos. Entretanto, a proposta considera relevante o perfeito ajustamento entre a modelagem e a fase de aquisição, com o propósito de facilitar o processo elicitação e validação dos requisitos entre os *stakeholders*. Esse aspecto é tratado no trabalho, prevendo uma metodologia de abordagem de requisitos afinada com o modelo de estruturação dos mesmos.

## 4 Argumentação

Contextualizado o processo de engenharia de requisitos e tendo estabelecidos os objetivos almejados, o modelo de requisitos proposto sustenta-se sob duas fundamentações principais: a abordagem orientada a objetivos, integrada a cenários, e a Teoria da Atividade.

A definição apropriada dos requisitos depende de um preciso conhecimento do ambiente e das razões pelas quais o software é proposto. Os clientes e usuários normalmente expressam esse propósito através de objetivos relacionados às tarefas. Segundo esse ponto de vista as metodologias de engenharia de requisitos orientadas a objetivos apresentam um suporte adequado ao descobrimento desses objetivos.

Porém, a ação humana deve ser analisada dentro de um contexto, para que possa fazer sentido e ser compreendida. Para que se possa obter adequadamente os requisitos de um software, é necessário o entendimento das atividades realizadas em determinadas situações no domínio do problema. Segundo a Teoria da Atividade [KAP97], a atividade é vista como a unidade básica de análise de situações. Assim, adota-se a idéia da atividade como base para a análise do comportamento do sistema, na elicitação e modelagem de requisitos.

### 4.1 Abordagens Orientadas a Objetivos

A necessidade de modelar integralmente o domínio do problema, para análise dos requisitos, é bem reconhecida. Nas primeiras atividades do desenvolvimento de software é importante a compreensão das necessidades apresentadas pelos clientes e usuários. Modelos têm sido desenvolvidos com esse propósito, acompanhado de ferramentas e técnicas da Engenharia de Requisitos que auxiliam no refinamento dos requisitos, a fim de alcançar maior precisão, completude e consistência [ALE98].

No entanto, em um estágio inicial do desenvolvimento, os clientes geralmente não têm uma visão clara sobre o que necessitam do sistema de software [KEN93] e os requisitos não refletem suas reais necessidades [BER94]. Clientes e usuários geralmente expressam suas necessidades como objetivos almejados para a organização, com o desenvolvimento do software.

Assim, o conhecimento da contribuição do desenvolvimento do software para ambiente organizacional é considerado um aspecto fundamental para a determinação dos requisitos. O entendimento das razões para o desenvolvimento do software é importante, não apenas para auxiliar, em primeira instância, a definição do sistema, mas também para facilitar a cooperação com outros sistemas, bem como para a sua evolução.

Muitos modelos de requisitos não se preocupam em responder questionamentos que visam detectar essas razões, os porquês, e acabam por concentrarem-se na funcionalidade do software. A modelagem segundo uma abordagem orientada a objetivos, ao contrário, aborda aspectos mais abstratos em comparação a descrição de dados, funções e comportamento.

Geralmente, clientes e usuários expressam suas necessidades por objetivos, relacionando-os às tarefas que são executadas em busca de atendê-los. Assim, a captura dos requisitos por refinamento dos objetivos contribui com a qualidade de facilitar o entendimento entre os envolvidos no processo de elicitação.

A abordagem por objetivos é uma metodologia eficaz na definição dos porquês do software e identificação dos requisitos. É amplamente utilizada em diferentes propósitos da Engenharia de Requisitos [YUE98], desde a descoberta das razões do desenvolvimento do software [ANT96] até o tratamento de conflitos de pontos de vistas sobre os requisitos [NUS96] [LAM98b]. O argumento desta abordagem é que as razões de um software, para a organização para qual ele é requerido, determinam as diretrizes do desenvolvimento. Questões relacionadas ao *porquê* são adicionadas ao tradicional *o que* no processo de engenharia de requisitos.

Um objetivo é simplesmente uma situação desejada. A especificação de objetivos, que declara os resultados desejados e não a maneira em obtê-los, permite entender as razões dos requisitos e perceber quais envolvidos colaboram na sua obtenção [WIE96].

A abordagem por objetivos é enfocada na participação e colaboração de todos os envolvidos já nas primeiras fases do desenvolvimento, seguindo a filosofia de que nas pessoas encontra-se o poder [NIS97]. Métodos que seguem essa abordagem, tipicamente exigem habilidades em animar e incentivar a colaboração dos envolvidos, e requerem ferramentas com notações informais e gráficas capazes de facilitar a comunicação e o entendimento, em prol da liberdade de criatividade.

O maior problema encontrado na modelagem por objetivos é a complexidade em lidar com o vago conceito de objetivo. Em geral, os clientes e usuários têm dificuldade em expressar os objetivos com um certo grau de abstração adequado a compreensão de todos os *stakeholders* [LAM98].

A fim de suprir essa dificuldade, Potts e outros [POT94] afirmam que os métodos de requisitos orientados a objetivos devem ser complementados com a utilização de cenários. O fundamento desta combinação é que cenários são utilizados como um mecanismo de obter exemplos e ilustrações capazes de auxiliar a operacionalizar os objetivos

Os cenários estão crescentemente sendo reconhecidos como um método eficaz para elicitare, modelar, analisar e validar requisitos de software. São utilizados como meios naturais para a captura do conhecimento pois descrevem episódios concretos e específicos das atividades do sistema, sob o ponto de vista dos clientes e dos usuários.

Um cenário é definido como uma seqüência temporal de interações de eventos entre diferentes agentes em um contexto restrito para obtenção de um propósito explícito [LAM98]. Utilizados no processo de engenharia de requisitos, os cenários são empregados com o propósito conversacional. Facilitam o trabalho de interação entre os *stakeholders* [WEI98], descrevendo as motivações e experiências dos clientes e usuários e caracterizando o fluxo de eventos e atividades do sistema [CAJ98][SUT98b].

A utilização de cenários neste contexto é devida à facilidade em obter-se requisitos com um grau de abstração apropriado à fase de definição, descrevendo de maneira inteligível o comportamento entre os agentes do domínio do problema e contribuindo como uma tipificação dos modelos conceituais.

O modelo de requisitos proposto neste trabalho segue essa abordagem que conjuga a definição dos objetivos, que constitui a abstração com a qual os usuários expressam suas necessidades, à descrição de cenários, com os quais descrevem casos de uso do sistema de uma forma amigável e compreensível aos *stakeholders*. O modelo é baseado, essencialmente, em modelos com abordagem orientada a objetivos, os quais estruturam os requisitos em níveis hierárquicos, através de refinamentos de composição e alternativas.

Em [LAM98] é apresentado um método formal para inferir objetivos e requisitos, baseado em aprendizado sobre cenários. Os objetivos são refinados de maneira *top-down* e abstraídos de forma *bottom-up*. Em paralelo, os cenários são capturados em colaboração com a definição dos objetivos, ilustrando-os e validando-os (figura 4.1).

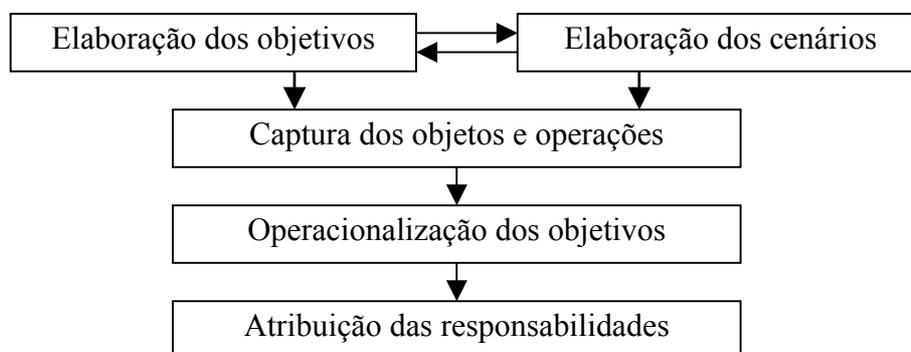


FIGURA 4.1 - Processo de inferência de objetivos [LAM98]

A etapa de elaboração dos objetivos consiste na identificação dos objetivos do sistema e sua definição. A elaboração de cenários é uma atividade que consiste em entender, analisar e descrever o comportamento de um sistema em termos das diferentes formas que se espera utilizá-lo, de maneira a colaborar com a definição dos objetivos.

As etapas seguintes de captura de objetos e operações, operacionalização de objetivos e atribuição de responsabilidades traduzem os objetivos e requisitos definidos em objetos que vão compor o sistema, bem como suas operações e comportamentos. É especificado o domínio da solução e seus limites com o ambiente (domínio do problema), através da identificação dos agentes e atribuição de suas responsabilidades, e também do estabelecimento das pré e pós condições das operações e da identificação dos objetos envolvidos com o atendimento aos requisitos.

O objetivo do processo é utilizar os cenários como uma ferramenta complementar à definição formal dos objetivos e requisitos. Os cenários são qualificados como um meio eficaz para a elicitación dos requisitos, dado o seu aspecto operacional e informal, porém apresentam uma abordagem parcial por tratar o nível de instância do comportamento do sistema. Também, deixam implícitas propriedades, requisitos e objetivos.

Os autores propõem um método formal que, conjugado à elicitación através de cenários, permite uma análise completa dos requisitos, pela inferência dos objetivos do sistema. A proposta oferece, portanto, um método sistemático de declaração de requisitos através de cenários que complementa uma análise formal dos objetivos, em um nível mais abstrato.

Esse aspecto é interessante para o trabalho aqui apresentado. A especificação dos requisitos pela conjugação de abordagens que atuam em níveis de abstração diferentes. Em um nível mais abstrato, dos requisitos de negócios, são abordados os objetivos que são operacionalizados em requisitos do usuário e requisitos funcionais. Neste nível de abstração mais próximo da especificação do software, os cenários apresentam-se como ferramentas eficazes para a definição dos requisitos com a

participação dos clientes e usuários, dadas as características de informalidade e de descrição comportamental a nível de instância.

Em [HAU98] é proposta uma abordagem de definição de um novo sistema a partir da observação do sistema correntemente em uso. Desta observação são definidos os objetivos do novo sistema, e um processo de mudanças que consiste na análise reversa, definição e implementação das mudanças e integração do legado, buscando a reutilização de modelos. O argumento dessa consideração sobre a funcionalidade do sistema existente para a definição de um novo sistema reside nas razões de que o novo sistema provê um alto grau de funcionalidade do sistema corrente e que pode-se aprender muito com histórias de sucesso e casos ocorridos no sistema existente.

A proposta é baseada na análise de objetivos. Procura-se definir um modelo de objetivos para que possa-se ilustrar, explicar, detectar, analisar e resolver interpretações do uso do sistema corrente e comparar, refinar e detalhar um modelo de objetivos do novo sistema. A abordagem proposta busca, basicamente, a captura de observações de uso do sistema e a modelagem do sistema corrente para a posterior definição de um novo modelo de objetivos, derivado dos primeiros. O processo pode ser visualizado (figura 4.2) nas atividades de observação de cenas de uso do sistema existente (RWS - *Real World Scenes*) e captura de exemplos (RWE - *Real World Example*) que vão modelar e validar o estado corrente do sistema (CSM - *Current State Model*). Dos exemplos (RWE) e modelos (CSM) são extraídos fragmentos (RWEF - *Real World Example Fragment*) que são utilizados na definição do modelo de objetivos.

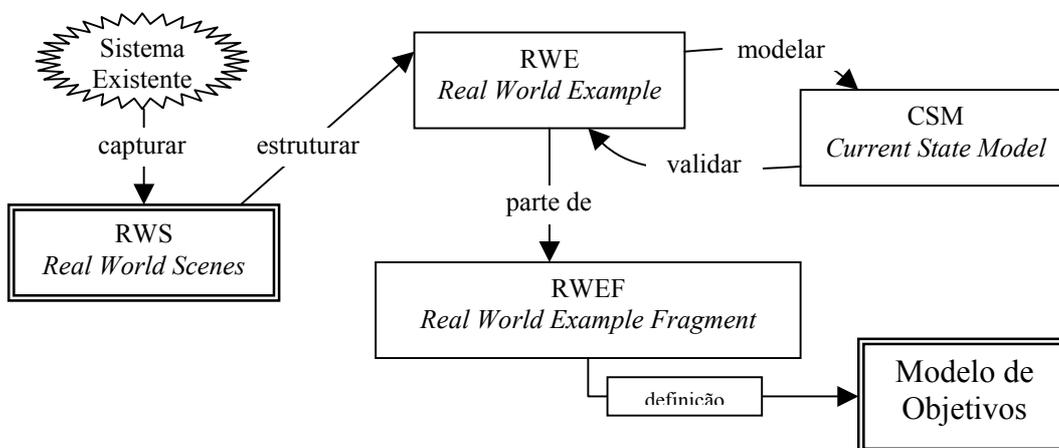


FIGURA 4.2 - Processo de definição de objetivos, com sistema existente [HAU98]

Os objetivos são derivados dos fragmentos de exemplos (RWEF), gerando uma tupla de exemplos e objetivo. Cada objetivo está associado a exemplos que podem ilustrá-lo positiva ou negativamente, isto é, os exemplos podem ilustrar evidências de acordo ou contra o atendimento ao objetivo. Entre os objetivos há um relacionamento hierárquico, representado por refinamentos OU e reduções E (figura 4.3). Os refinamentos OU estabelecem que o atendimento a um objetivo dá-se pelo atendimento a um de seus sub-objetivos. Já as reduções E indicam que o atendimento a todos os sub-objetivos é necessário para atender ao objetivo hierarquicamente superior.

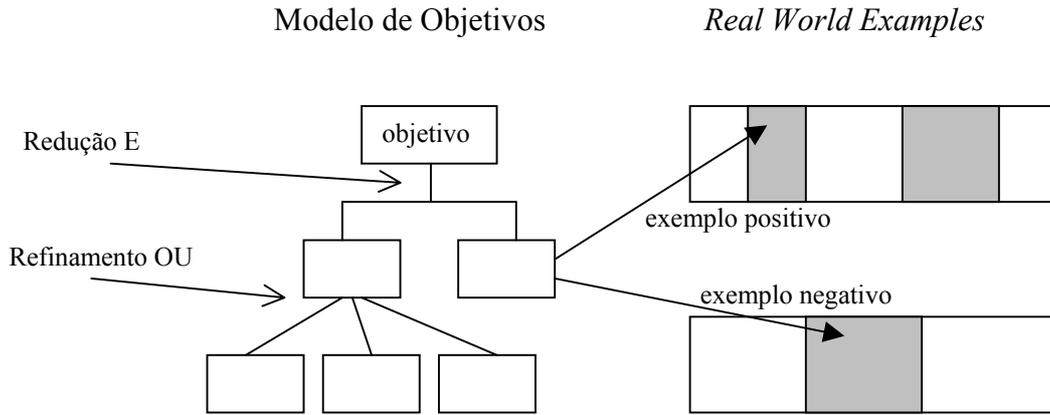


FIGURA 4.3 - Modelo hierárquico de objetivos e fragmentos de exemplos [HAU98]

Segundo os próprios autores, o relacionamento entre o modelo conceitual dos objetivos e os exemplos capturados do uso do sistema, além de facilitar a definição de novos objetivos baseados em objetivos do sistema existente, permite um melhor entendimento do processo de modelagem e facilita o relacionamento entre os *stakeholders* pela representação estruturada dos objetivos e exemplos capturados. Essas características agregam transparência e rastreabilidade ao processo de abstração, considerando os relacionamentos existentes entre os objetivos e as associações destes aos exemplos reais.

Esses aspectos são avaliados como importantes na definição do modelo aqui proposto. A estruturação dos objetivos é considerada uma qualidade importante para o entendimento das necessidades, e a associação a cenários como uma maneira de ilustrar casos específicos e concretos de uso do sistema que facilita a interação entre os *stakeholders*.

A proposta apresentada em [ROL98] oferece basicamente três características visando o auxílio direto às atividades de especificação de requisitos: (1) a associação bidirecional entre objetivos e cenários, na qual para cada objetivo descoberto, um cenário confirma-o, e para cada cenário é explorada a descoberta de objetivos (figura 4.4); (2) o refinamento dos objetivos, o qual ameniza seu vago conceito, especializando e generalizando-os em níveis hierárquicos de definição; e (3) o suporte à metodologia, através da especificação de diretrizes que orientam a aplicação das técnicas em um processo de especificação de requisitos.

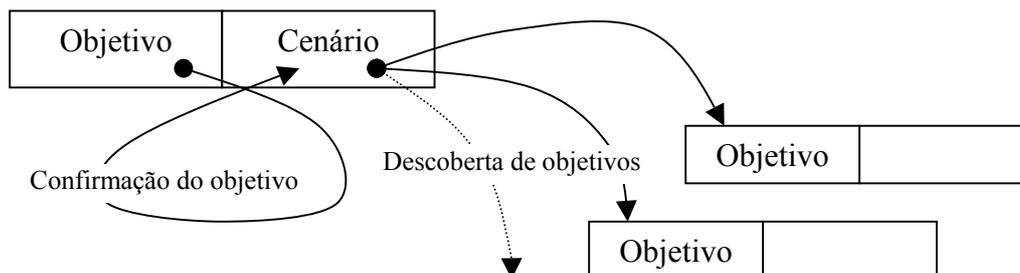


FIGURA 4.4 - Associação bidirecional entre objetivos e cenários [ROL98]

A essência da abordagem é o modelo do requisito (*requirement chunk*). O requisito é definido por um par <objetivo, cenário> que pode ser refinado e composto por relacionamentos entre requisitos (figura 4.5). Neste contexto, um objetivo é algo que algum envolvido espera obter no futuro e um possível comportamento limitado a um conjunto de interações realizadas entre alguns agentes com um propósito.

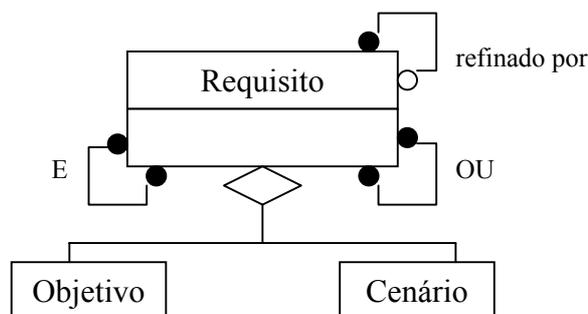


FIGURA 4.5 - *Requirement chunk* [ROL98]

Segundo esse modelo, o objetivo é descrito por um verbo associado a um conjunto de parâmetros que exprimem o alvo, a direção, o caminho e o beneficiário. O alvo designa entidades afetadas pelo objetivo, que pode ser distinguido em *objeto*, que existia antes do objetivo ser alcançado, e *resultado*, que é obtido após o atendimento ao objetivo. A direção identifica a localização dos objetos, e é especializada em *origem* e *destino*. O caminho é expresso pelos *meios*, que designam entidades as quais atuam como instrumentos para o atendimento ao objetivo, e pela *maneira*, que define a forma com que o objetivo é alcançado, podendo ser definida recursivamente. O *beneficiário* é a pessoa, ou grupo, a favor de quem o objetivo é pretendido.

Conforme o modelo, um cenário é representado por um conjunto de ações que ocorre entre agentes de um contexto específico, determinando alteração no estado do sistema. É caracterizado por um *estado inicial* e um *estado final*, que definem, respectivamente, as pré condições para desencadear sua execução e as condições atingidas ao final de sua execução. As *ações* de um cenário podem ser *atômicas*, interações entre dois agentes que afetam algum *objeto* (*agente* ou *recurso*), ou um *fluxo de ações*, que compreende uma composição de ações. Um cenário pode ser classificado em normal ou de exceção. Um *cenário normal* representa uma situação na qual o objetivo é alcançado, enquanto que uma *cenário de exceção* representa uma situação de falha no atendimento ao objetivo.

O modelo apresenta, também, uma hierarquia de requisitos como uma maneira de especificá-los em diferentes níveis de abstração. Abstração é definida como um mecanismo de ocultação de detalhes para focar aspectos essenciais. A associação *refinado por* do modelo de requisitos (figura 4.5) reflete essa ligação hierárquica vertical entre os diversos requisitos representados no modelo. Os relacionamentos de composição e alternativo, representados no modelo de requisitos pelas associações E e OU, respectivamente, auxiliam na estruturação dos objetivos. Estabelecem uma ligação horizontal entre os objetivos. Os relacionamentos de composição (E) refletem uma associação de objetivos que são requeridos para atender completamente uma certa

funcionalidade do sistema. Os relacionamentos alternativos (OU) representam caminhos alternativos para atender a um mesmo objetivo.

Quanto à elicitación dos requisitos, a proposta prevê um conjunto de ações repetidas de descoberta de objetivos e confirmação de cenários, que vão populando, incrementalmente, a hierarquia dos requisitos. Esse processo pode ser visualizado como um fluxo de passos, no qual, cada passo identifica um objetivo e descreve um cenário como uma possível concretização. No progresso dessas ações, a análise dos cenários volta a determinar novos objetivos. O resultado final dessas interações é uma completa definição dos requisitos (figura 4.6).

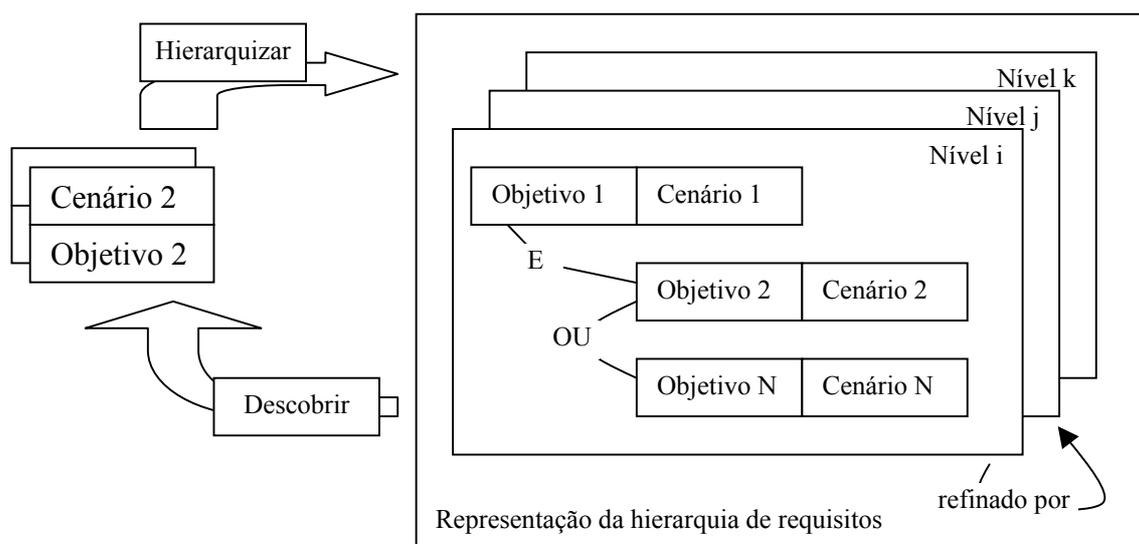


FIGURA 4.6 - Visão geral do processo de descoberta de objetivos [ROL98]

O processo de descoberta de objetivos possui um fluxo de controle baseado em estratégias de refinamento, composição e alternativas aos objetivos. A cada passo uma nova estratégia é dinamicamente utilizada. A maior contribuição desta abordagem é que, dado a natureza *ad hoc* do processo de decomposição, os refinamentos dos requisitos podem ser sistematicamente aplicados em diversos níveis de abstração do sistema.

Esse aspecto de sistematização de refinamento dos requisitos é uma característica importante a ser considerada no modelo aqui proposto. Essa característica permite uma estruturação dos requisitos em diferentes níveis de abstração, com uma abrangência desde o nível de requisitos de negócio até os níveis de requisitos do usuário e funcionais. Também, a associação direta dos requisitos a cenários contribui para o processo de modelagem dos requisitos, por ilustrá-los em casos de uso do sistema.

Outro aspecto importante de contribuição para o modelo proposto é a necessidade de representar os objetivos e cenários por objetos adequadamente determinados. Ambos elementos que compõem os requisitos são caracterizados por um conjunto objetos com uma semântica bem definida. Esse aspecto agrega ao modelo um formalismo que colabora com a estruturação dos requisitos e, principalmente, oferece um suporte adequado à sistematização do processo de especificação dos requisitos e conseqüente desenvolvimento do software.

Sutcliffe e Maiden [SUT98] propõem a utilização da teoria do domínio do conhecimento com o objetivo de definir a semântica e a composição de modelos de aplicações genéricos, no contexto da Engenharia de Software. A abordagem busca definir problemas em classes de conhecimento análogos, isto é, de abstrações comuns.

O objetivo principal é definir um domínio de conhecimento visando sua reutilização. Propõe o reconhecimento e armazenamento de abstrações de problemas correlatos, no mais alto nível do conhecimento, em metamodelos que podem ser selecionados por analogia e instanciados em um processo de especificação de modelos computacionais.

A teoria do domínio é uma teoria híbrida das ciências cognitivas e da Engenharia de Software. Utiliza teorias de representação do conhecimento e memória, aplicadas a conceitos de abstração utilizados na Engenharia de Software. É uma formalização genérica do domínio do conhecimento em classes de aplicações visando a reutilização.

Os autores acreditam que a aplicação da teoria do domínio contribui incisivamente no processo de engenharia de requisitos por delimitar um domínio do conhecimento. Por isso, essa teoria tem sido reconhecida como um fator crítico para a obtenção completa, consistente e acurada dos requisitos.

Essa conclusão é um argumento importante na justificativa do modelo aqui proposto. A delimitação do conhecimento contribui positivamente para o entendimento do domínio do problema. Conforme palavras dos autores do artigo supracitado que "os usuários geralmente expressam os requisitos por objetivos relacionados a tarefas", pode-se inferir que os objetivos do sistema definem domínios de conhecimento que podem ser abordados particularmente na definição e especificação de requisitos.

## 4.2 Teoria da Atividade

A adoção de um processo sistemático auxilia as atividades da Engenharia de Requisitos, em especial, a descoberta dos objetivos do sistema. Porém, a problemática que naturalmente existe no processo de compreensão humana, que está no cerne da elicitación dos requisitos, necessita de uma abordagem que considere efetivamente o contexto em que os usuários exercem suas atividades.

Muitas teorias psicológicas utilizam a ação humana como unidade básica de análise de situações [NAR96]. A ação deve ser analisada dentro de um contexto, para que possa fazer sentido e ser compreendida, como, por exemplo, em uma situação fechada, isolada, definida por um cenário.

A proposta apresentada em [MAR99] sugere a utilização de alguns preceitos da Teoria da Atividade na elicitación dos requisitos de software, considerando-se que os aspectos sociais assumem grande importância nesta atividade.

A Teoria da Atividade é originária da antiga União Soviética, derivada da escola histórico-cultural de psicologia, fundada por Rubinshtein e Vygotsky. A teoria é uma estrutura conceitual (no original, *a conceptual framework*), desenvolvida por Leontjev (aluno de Vygotsky) para o estudo das diferentes formas da prática humana como um processo de desenvolvimento cultural, no qual ambos níveis individual e social são interligados.

Atualmente, essa teoria tem grande aceitação em pesquisas que abordam o entendimento da interação entre o ser humano e o seu ambiente material, em um

contexto social e cultural [KAP95]. Particularmente, neste trabalho, o interesse reside na abordagem da interação entre homem e um sistema computável, no âmbito da atividade.

Segundo a Teoria da Atividade [KAP97], a atividade é o contexto mínimo no qual a ação humana deve ser analisada para que possa fazer sentido e ser compreendida. A atividade é, então, a unidade básica de análise de situações. Esse conceito de atividade implica a existência de um agente que age sobre alguma coisa, de maneira que a atividade media a interação entre sujeitos (agentes) e objetos (coisas).

A Teoria da Atividade possui cinco princípios básicos: (1) orientação a objetos, (2) interiorização e exteriorização, (3) ferramentas de mediação, (4) estrutura hierárquica e (5) desenvolvimento contínuo.

#### (1) *Orientação a Objetos*

A atividade não existe como uma entidade isolada por si só. Toda a atividade é direcionada para algo que objetivamente exista no mundo real, para um objeto. A noção de objeto não é limitada a propriedades físicas de entidades, mas considera, também, propriedades determinadas socialmente e culturalmente, que podem ser transformados no curso de uma atividade. O objeto pode ser algo material, ou algo menos tangível, como um plano ou uma idéia.

#### (2) *Interiorização e Exteriorização*

A atividade possui um lado interno e um lado externo que devem ser analisados em conjunto pois há uma mútua transformação entre eles. A interiorização corresponde ao processo mental, na qual são simuladas ações sem a interação com os objetos reais. Já a exteriorização é a transformação das atividades internas em externas, resultando uma interação com os objetos. A divisão entre atividade interna e externa é meramente artificial, qualquer atividade externa é suportada por processos que são originados internamente ao sujeito e o processo interno aparece de uma maneira ou outra no mundo externo.

#### (3) *Ferramentas de Mediação*

A atividade é mediada por inúmeras ferramentas que são utilizadas na interação entre os sujeitos e os objetos. Essa mediação é feita por artefatos os quais, de um modo geral, definem e incluem objetos materiais (instrumentos, máquinas) ou influências (sinais, linguagem, experiências, leis, métodos, formas de organização) que são criados para controlar o comportamento dos sujeitos perante os objetos. Esses artefatos refletem o conhecimento adquirido, transportando uma cultura e história particular, e são estruturas persistentes que estendem ao longo das atividades através do tempo e do espaço.

Em um nível individual, uma atividade possui, então, três elementos básicos: sujeito, objeto e ferramenta de mediação (figura 4.7). O sujeito é o agente que atua sobre o objeto da atividade. O objeto é o elemento para o qual as ações da atividade estarão direcionadas. O relacionamento recíproco entre o sujeito e o objeto da atividade é sempre mediado por uma, ou mais, ferramentas. Ferramentas possuem um papel de mediação, usadas no processo de transformação do objeto. Nessa transformação algum resultado deve ser obtido. Transformar um objeto em um resultado motiva a existência de uma atividade.

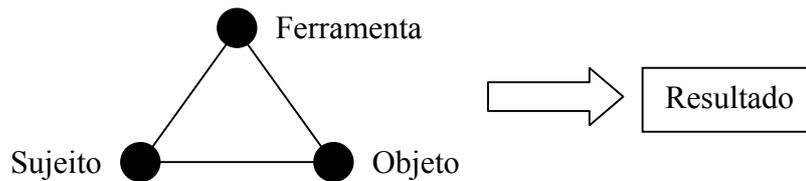


FIGURA 4.7 - Elementos básicos de uma atividade

#### (4) Estrutura Hierárquica

De acordo com a Teoria da Atividade, a interação entre seres humanos e o mundo é organizada em níveis funcionais hierarquicamente subordinados. Há três níveis na estruturação da atividade: atividade, ações e operações (figura 4.8). A atividade é orientada a motivos que, basicamente, refletem a necessidade humana. O motivo é o objeto da atividade como um todo, que a guia por antecipação.

As ações constituem o nível central, e são orientadas a metas, as quais constituem os objetos das ações. Tipicamente, as metas são funcionalmente subordinadas a outras metas e, sucessivamente, até um nível hierárquico superior onde há uma meta não subordinada, o motivo. 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, o qual é chamado de orientação.

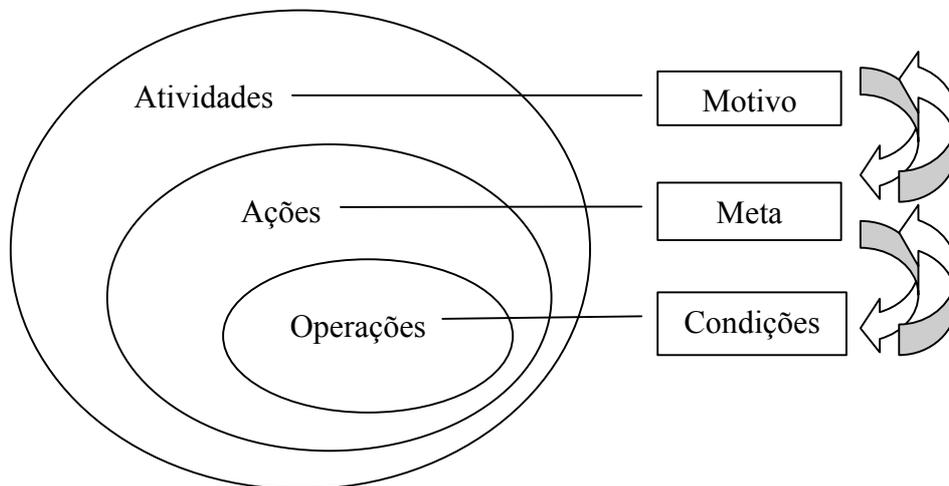


FIGURA 4.8 - Níveis hierárquicos de uma atividade

Cada ação realizada por um ser humano, portanto, não tem somente aspectos intencionais, mas também aspectos operacionais. Em um nível inferior, cruza-se o limite entre os processos conscientes e os automáticos e tem-se o nível das operações. Neste último nível, o ser humano age por condições (não há objetivos) e executam operações respondendo a situações correntes. As operações tornam-se rotineiras e inconscientes com a prática e dependem de condições, sob as quais a ação

está sendo conduzida e que são estabelecidas pelo conhecimento adquirido. Na literatura de interface homem-computador (HCI, *Human Computer Interface*), as atividades são diretamente equivalentes a tarefas e o nível de operações é uma adaptação dos aspectos físicos da interface com o usuário [NAR96].

(5) *Desenvolvimento Contínuo*

A Teoria da Atividade afirma que a interação humana com a realidade deve ser analisada no contexto do desenvolvimento. A atividade por si só é o contexto, no qual são envolvidos sujeitos e ferramentas na transformação de objetos. Todas as práticas são resultados de um certo desenvolvimento histórico, sob certas condições, em contínuo processo de reformulação e desenvolvimento cultural. Quando uma atividade é realizada, há um mecanismo de retorno o qual compara o resultado da atividade com a predição e alguma incongruência promovendo uma situação de aprendizado, isto é, o conhecimento da pessoa é expandido.

A análise das relações sistêmicas existentes no contexto de uma atividade contribui para uma eliciação de requisitos mais criteriosa, pois leva a quem está realizando a tarefa considerar elementos importantes para o conhecimento das necessidades. A estrutura hierárquica e a composição da atividade também contribuem de forma incisiva na organização, estruturação e análise do domínio do problema. São contribuições determinantes da Teoria da Atividade para a argumentação do modelo proposto neste trabalho.

## 5 Modelo de Requisitos

Este trabalho tem o objeto de definir um modelo de estruturação e representação de requisitos com uma abordagem voltada ao usuário. O modelo proposto é definido com o propósito de organizar os requisitos para o desenvolvimento do software, visando atender à atividade de modelagem no processo de engenharia de requisitos e oferecer um suporte adequado à elicitação.

A usabilidade do modelo é uma característica importante para os objetivos propostos, com a qual busca-se estruturar os requisitos visando a rastreabilidade ao longo do ciclo de vida e ser proficiente a comunicação entre os envolvidos.

É premissa que os clientes e usuários, normalmente, expressam suas necessidade através de objetivos almejados e que, para a obtenção adequada dos requisitos, é necessário o perfeito entendimento das atividades realizadas em situações que expressam o domínio do problema. Segue o pressuposto que a definição apropriada dos requisitos depende de um preciso conhecimento do ambiente e das razões pelas quais o software é proposto e que a ação humana deve ser analisada dentro de um contexto, para que possa fazer sentido e ser compreendida.

Segundo essa base de raciocínio, o uso conjugado de abordagens da Engenharia de Requisitos orientadas a objetivos com alguns preceitos da Teoria da Atividade apresentam-se como uma proposição adequada à especificação dos requisitos de software.

Sustentado sobre essas argumentações básicas, o modelo proposto é definido conceitualmente com base nos princípios de orientação a objetos, ferramentas de mediação e estrutura hierárquica da atividade, segundo a Teoria da Atividade. A atividade é uma forma de agir de um sujeito direcionada a um objeto, a qual é decomposta em ações, e as ações em operações. Enquanto uma atividade é orientada por um motivo, as ações são orientadas a metas, e as operações executadas segundo condições.

### 5.1 Os Objetivos

O princípio da orientação a objetos induz que toda a atividade é motivada a um objeto. Há uma analogia bem aproximada que pode-se fazer entre esse princípio da atividade e as abordagens orientadas a objetivos. O argumento dessas últimas reflete que as razões de um software determinam as diretrizes do desenvolvimento, constituindo, portanto, o motor orientador para o atendimento a uma necessidade. Essas razões, ou objetivos, são, conceitualmente, análogas aos motivos de uma atividade. Constituem o resultado almejado que condiciona as ações. Então, pode-se tratar um objetivo como uma atividade direcionada a um objeto, ou seja, visa alcançar algum resultado.

Essa análise permite deduzir que um objetivo pode ser perfeitamente definido como uma atividade. Desta conclusão, decorre que, considerando que a Teoria da Atividade prega que uma atividade é definida por um conjunto de elementos básicos, um objetivo, também, pode ser completamente explicado por uma composição desses elementos básicos: sujeito, objeto e ferramenta.

Cada objetivo é, portanto, modelado por uma composição desses elementos (figura 5.1). O resultado é o motivo, ou o objetivo propriamente dito. O

objetivo descreve a necessidade em si, e pertence ao domínio do problema. No modelo, um objetivo é representado pela informação elicitada, que, segundo as abordagens orientadas a objetivos, responde a um questionamento *por quê*.

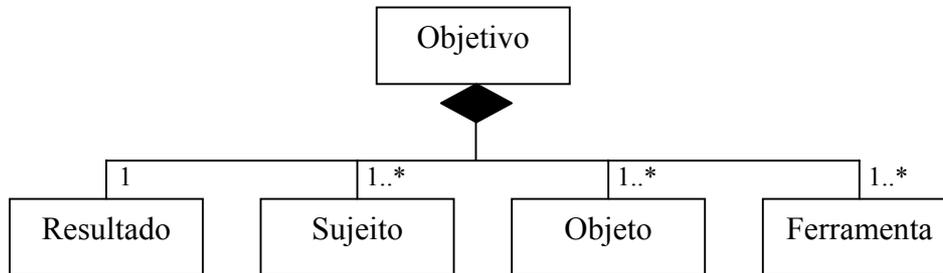


FIGURA 5.1 - Definição de um objetivo

Um exemplo pode ilustrar essa definição. Uma empresa de saneamento possui, em sua estrutura organizacional, um setor responsável pelos serviços de setorização da rede distribuidora de água. O setor, chamado de controle operacional, é composto por equipes, controladas por um chefe, que realizam esses serviços. A setorização é uma obra de engenharia para otimização da rede de abastecimento em uma área delimitada, através de planejamento técnico no qual são analisados pressão e vazão da rede e perfil dos consumidores atendidos.

Há, neste contexto resumidamente apresentado, a necessidade de ter-se uma aplicação com o propósito de manter informações quantitativas dos serviços realizados diariamente e os materiais utilizados pelas equipes, a fim de permitir ao gerente do setor de controle operacional planejar mais precisamente os custos das obras.

Elicitadas as informações, pode-se definir um objetivo para o sistema em questão: gerenciamento de informações para o planejamento das obras de setorização. Essa é a necessidade do cliente. Reflete uma diretriz básica que deve ser seguida no desenvolvimento do software, respondendo ao questionamento *por quê*. Por que desenvolver o software? Para que o gerente do setor disponha de informações que permitam o planejamento preciso das obras de setorização. Isso delimita um atividade do sistema, isto é, um contexto do domínio do problema que envolve o setor de controle operacional.

Essa necessidade, gerenciamento de informações para o planejamento das obras, é o objetivo, cuja descrição é modelada pelo resultado. A informação elicitada compõe o modelo, instanciando um objetivo, definido pelo seu componente resultado. Desta informação infere-se os outros objetos que compõem o objetivo: objeto, sujeito e ferramenta.

Essa inferência, nesta etapa do processo de engenharia de requisitos, define abstrações adequadas aos clientes. Considerando a fase de conhecimento e entendimento do domínio do problema, trata-se de um nível de representação em acordo com a visão que os clientes têm do sistema. Um glossário de definição de termos especializados do domínio é necessário a fim de atenuar o problema de diferenças de conceituação e entendimento entre os envolvidos, equalizando o conhecimento e facilitando o aspecto de comunicação.

### 5.1.1 Objeto

Objeto é o elemento para o qual as ações estão direcionadas, e pode ser algo material ou abstrato, conforme o princípio da orientação a objetos. Representa o elemento que pode ser produzido, transformado ou consumido no curso da atividade, e que a motiva. Isto é, descreve alguma coisa para qual o objetivo existe. No processo de elicitação, o objeto responde a um questionamento de *para o que* esse objetivo faz-se necessário ao sistema. Ou seja, determina-se o objeto analisando o resultado esperado no atendimento ao objetivo definido, no sentido de elucidar-se para qual "coisa" tal objetivo foi definido.

No exemplo do setor de controle operacional, questiona-se para o que é direcionado o gerenciamento de informações para o planejamento das obras de setorização. Existe algo, material ou não (idéia ou plano), para o qual a atividade de planejamento de obras de setorização é realizada? Sim, para a determinação do custo da obra. Esse é o motivo da atividade, é o objeto para o qual o objetivo é definido. O gerente do setor necessita, para planejar uma obra de setorização, estabelecer o seu custo em termos de serviços e materiais empregados. Então, deduz-se pela informação elicitada, que o objeto do objetivo é o custo da obra. Esse objeto representa a coisa que será criada e transformada durante a execução desta atividade específica.

Esclarece-se que o objeto de um objetivo pode não ser único. Pode haver objetivos que estão definidos por mais de um objeto. No exemplo, considerando que o setor de controle operacional não atua somente na setorização, mas também é responsável pela manutenção da rede de água, poder-se-ia definir um objetivo de gerenciamento de informações para planejamento de atuação do setor. Este objetivo seria composto de dois objetos, o custo das obras de setorização e o custo de médio das manutenções realizadas na rede. Neste caso, para a representação do objetivo haveria a necessidade de instanciação de dois objetos.

### 5.1.2 Sujeito

Segundo o modelo, o sujeito participa, também, da composição do objetivo. O sujeito é o agente que atua sobre o objeto. Representa o elemento ativo da atividade, o qual age, exteriorizando uma atividade interna, segundo o princípio da exteriorização da atividade. No modelo, o sujeito é definido como um agente responsável pela criação, transformação ou destruição do objeto. Isto é, descreve alguém ou algum ente, presente no domínio do problema, que tem participação ativa sobre o objeto em questão. No processo de elicitação, responde a um questionamento de *quem* atua em prol do objetivo definido. Ou seja, determina-se o sujeito analisando quem são os envolvidos na atuação sobre o objeto.

No exemplo do setor de controle operacional, os sujeitos são identificados pelo questionamento de quem atua no gerenciamento de informações para o planejamento das obras de setorização. São definidos dois sujeitos, o chefe da equipe que é o responsável pela geração das informações dos serviços realizados diariamente e do material empregado, e o gerente do setor que é quem usufrui dessas informações a fim de executar o planejamento das obras. Ambos sujeitos agem sobre o objeto, o custo da obra. O gerente cria-o, baseado em informações do chefes das equipes, que são os responsáveis pela transformação do objeto, através do provimento de informações sobre os serviços e materiais empregados no cotidiano de sua atuação.

### 5.1.3 Ferramenta

Pelo princípio das ferramentas de mediação da Teoria da Atividade, a ferramenta também participa da composição do objetivo. As ferramentas são utilizadas na interação entre sujeitos e objetos. Representam os artefatos materiais ou de influência que são utilizados pelos sujeitos na criação, transformação e destruição de objetos. Descrevem os recursos, que podem ser desde instrumentos até métodos, ou qualquer conhecimento ou forma de ação utilizados no contexto específico da atividade em questão. Essa utilização das ferramentas sobre objetos é disciplinada pela obtenção do resultado esperado, ou seja, para o atendimento ao objetivo.

Para a determinação das ferramentas questiona-se *quais* são os recursos necessários para os sujeitos interagirem com os objetos. No exemplo em tela, analisa-se quais são os recursos utilizados pelo gerente e chefes de equipes para a elaboração do custo das obras de setorização. Em resposta a essa pergunta, descobre-se que são necessárias ordens de serviços e uma planilha contendo os dados quantitativos sobre a atuação das equipes. As ordens de serviços são documentos preenchidos por chefes de equipes com as informações quantitativas dos serviços realizados e materiais utilizados diariamente pela equipe na realização dos trabalhos em obras de setorização. A planilha é utilizada pelo gerente para consolidar as informações quantitativas das equipes em uma obra específica de setorização. Dessa planilha pode-se obter dados cumulativos sobre o rendimento das equipes e a provisão de materiais, os quais serão utilizados para atender ao objetivo de gerenciamento de informações para o planejamento das obras de setorização.

Esses são os instrumentos utilizados pelos sujeitos (gerente e chefes de equipes) para a manipulação dos objetos, isto é, para a criação e manutenção dos custos das obras. As ordens de serviços e a planilha são utilizadas para transformar o custo da obra; cada dado informado na ordem de serviço e atualizado na planilha reflete uma alteração no custo da obra.

Compõe-se, assim, o objetivo definido (figura 5.2): gerente e chefes de equipes (sujeitos), utilizando ordens de serviços e planilhas de dados (ferramentas), na obtenção e manutenção de custo da obra (objeto), com o objetivo de gerenciar as informações para o planejamento de obras de setorização (resultado).

A elicitação das informações sobre o objetivo é realizada com a participação dos envolvidos. O emprego desses princípios da atividade visa facilitar essa interação, com a argumentação que a atividade é o contexto mínimo no qual a ação humana deve ser observada para que possa fazer sentido e ser compreendida. A atividade é, portanto, considerada uma unidade básica natural para a análise de situações. O objetivo, em analogia com a atividade, define um contexto específico e bem determinado para o exame do domínio do problema.

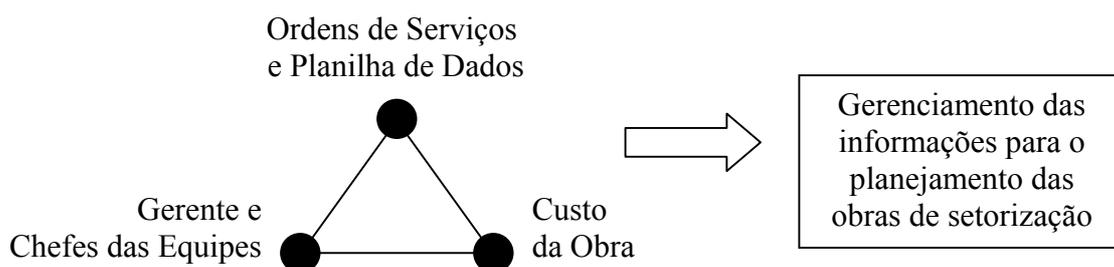


FIGURA 5.2 - Composição do objetivo do exemplo do setor de controle operacional

O processo de elicitação deve ser concomitante ao modelo de objetivo, o qual define a estruturação das informações. Considerando o exemplo apresentado, o objetivo é elicitado em uma estrutura bem definida que facilita a modelagem segundo a proposta (figura 5.3). A estrutura das informações prevê a definição do objetivo e a descrição dos objetos que o compõem. São sugeridas questões que tem o propósito de descobrir, ante ao objetivo, os objetos resultado, sujeitos, objetos e ferramentas.

<b>Objetivo</b>	Manter informações quantitativas dos serviços realizados diariamente e os materiais utilizados pelas equipes a fim de permitir ao gerente do setor de controle operacional planejar os custos das obras.
<b>Resultado</b>	<i>Qual o resultado esperado? Por que há a necessidade?</i> gerenciamento de informações para o planejamento das obras de setorização
<b>Objeto</b>	<i>Para o que esse objetivo faz-se necessário ao sistema?</i> custo da obra
<b>Sujeito</b>	<i>Quem atua em prol desse objetivo?</i> gerente chefes de equipes
<b>Ferramenta</b>	<i>Quais são os recursos necessários para os sujeitos interajam com o objeto?</i> ordens de serviços planilha de dados

FIGURA 5.3 - Estrutura para definição de objetivo, no processo de elicitação

#### 5.1.4 Glossário de Definição de Termos

O glossário de definição de termos é construído em conjunto com todos os envolvidos, a fim de obter-se uma definição precisa dos elementos percebidos do domínio do problema. No exemplo, os objetos que compõem o objetivo são definidos durante a elicitação, constituindo o início do glossário de termos que deve perdurar durante todo o processo de desenvolvimento (figura 5.4).

## 5.2 A Estrutura Hierárquica

Outro princípio da Teoria da Atividade importante a ser considerado no modelo é o princípio da estrutura hierárquica da atividade. Segundo esse princípio, a atividade humana é organizada em níveis funcionais hierarquicamente subordinados. Em um nível mais alto, ou mais abstrato, está a atividade, que reflete a necessidade humana, orientada por algum motivo. Para a integral conclusão da atividade é necessário a realização de ações, as quais são dirigidas por metas. Neste patamar, as ações derivam outras ações e assim, sucessivamente, compondo uma estrutura hierárquica na qual a funcionalidade é planejada. No nível mais inferior, ou mais operacional, estão as operações, as quais são executadas automaticamente em resposta a uma configuração de condições.

<b>Termo</b>	<b>Definição</b>
setor de controle operacional	setor encarregado dos serviços de setorização
setorização	obra de engenharia para a otimização da rede de abastecimento em uma área delimitada, através de planejamento técnico no qual são analisadas pressão e vazão da rede e perfil dos consumidores atendidos
gerente	responsável pela gerência do setor de controle operacional; quem planeja as setorizações, com base nos custos das obras
chefe de equipe	responsável pelo controle de uma equipe que executa as obras de setorização; quem informa os serviços realizados e os materiais utilizados nas obras diariamente
custo da obra	informação da quantidade de serviço necessário e de material empregado para a realização de uma obra de setorização
ordem de serviço	documentos preenchidos pelos chefes de equipes com as informações quantitativas dos serviços realizados e materiais utilizados diariamente pela equipe na realização dos trabalhos em obras de setorização
planilha de dados	documento com dados quantitativos consolidados sobre o rendimento dos serviços realizados pelas equipes e a provisão de materiais em uma obra de setorização, utilizado pelo gerente para o planejamento do custo de outras obras de setorização

FIGURA 5.4 - Glossário de termos do objetivo do setor de controle operacional

Este princípio está fortemente ligado a algumas abordagens orientadas a objetivos, as quais propõem a estruturação dos requisitos de forma hierárquica. Nestas abordagens, a hierarquia dos objetivos e requisitos modela as informações elicitadas em diferentes níveis de abstração do sistema. Os modelos, portanto, possuem uma abrangência de representação desde o nível de requisitos de negócio, onde tem-se uma visão mais abrangente e próxima do domínio do problema, até os níveis de requisitos do usuário e funcionais, onde a visão do sistema reflete a funcionalidade requerida ao software.

Pode-se inferir, então, um relacionamento consistente entre o princípio da estrutura hierárquica da atividade e a representação hierárquica dos requisitos proposta pelas abordagens orientadas a objetivos. Neste relacionamento encontra-se argumentação para propor um estrutura hierárquica ao modelo de requisitos, com o objetivo de estruturação e melhor entendimento do que deseja-se desenvolver. Essa estrutura permite a modelagem dos requisitos em diferentes níveis de abstração, o que vem a contribuir diretamente com o grau de usabilidade do modelo. Lembrando que a usabilidade, neste contexto, é uma característica pretendida com o intuito de facilitar a comunicação entre os envolvidos sobre os requisitos.

Em apreciação realizada nesse sentido, o modelo de requisitos possibilita a composição hierárquica dos requisitos (figura 5.5). Essa estruturação é dada pela associação entre objetivo e requisitos e refinamentos entre os requisitos. O objetivo define a atividade e os requisitos definem as ações.

Os requisitos são detalhados, através dos refinamentos, abordando níveis de abstração diferentes da visão dos clientes e usuários sobre o domínio do problema. Em um nível logo abaixo da definição do objetivo estão os requisitos dos usuários, que derivam requisitos funcionais. Estes últimos definem a funcionalidade que o software deve prover a fim de capacitar os usuários a realizar suas ações.

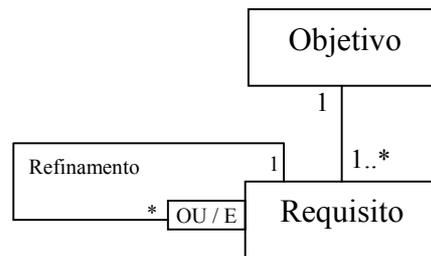


FIGURA 5.5 - Estrutura hierárquica dos requisitos

Entre o objetivo e os requisitos existe um limite que separa o ambiente do software. Essa demarcação é útil na definição de domínios de análise, assim como é, por exemplo, o conhecido modelo referência de redes OSI<sup>ISO</sup>, que define o protocolo e o escopo de atuação das camadas de comunicação entre sistemas [GUN2000]. Na Engenharia de Requisitos, essa delimitação é utilizada na classificação dos requisitos, e tem grande utilidade na abordagem do tratamento dos requisitos do software.

O ambiente, cuja abrangência é demarcada pelo domínio do problema, representa o conhecimento do sistema e a área de atuação do software. Esse escopo é definido pelos clientes, os quais requerem o desenvolvimento do software com o propósito de atender aos objetivos definidos para a organização. Neste escopo estão os requisitos de negócio, que, no modelo, são representados pelos objetivos.

O domínio do software enquadra o domínio da solução. Neste escopo trabalha-se com os usuários na definição das características e requisitos requeridos para o software. São representadas as visões abstratas que os usuários têm do sistema a ser desenvolvido e são especificadas a funcionalidade e o comportamento que o software deve apresentar, como, também, as informações que estão sob seu domínio. Constitui o nível dos requisitos de usuário e requisitos funcionais que, no modelo, são representados pelos requisitos propriamente ditos.

No escopo dos requisitos de usuário e funcionais, estes podem estar associados através de refinamentos qualificados, expressando composições e alternativas de funções a serem disponibilizadas pelo software. As composições de requisitos, representadas pelo qualificador E, denotam um conjunto de ações que devem ser realizadas a fim de atender ao objetivo, ou a uma ação hierarquicamente superior. Isto significa que para que o objetivo ou requisito de mais alto nível seja contemplado completamente é necessário que todos os seus refinados sejam executados. As alternativas de requisitos, representadas pelo qualificador OU, denotam um conjunto de alternativas de ações que podem ser realizadas para atender ao objetivo ou a uma ação de mais alto nível hierárquico. Isso significa que para que o objetivo ou requisito seja contemplado é necessário que algum de seus refinados seja executado.

Ressalta-se que a modelagem sempre ocorre em um contexto de uma atividade e suas ações decorrentes. Ou seja, parte-se de um objetivo específico, do qual deriva-se os requisitos necessários para o seu atendimento. Os requisitos modelados a partir dos objetivos representam uma visão mais abstrata do sistema, enquanto que os refinamentos buscam visões de abstrações de componentes do software. Esse mecanismo é utilizado buscando uma definição precisa e interativa dos requisitos,

<sup>ISO</sup> *Open System Interconnection* é um modelo de referência de rede de computadores, desenvolvido pela *International Standards Organization*, para a padronização dos protocolos utilizados nas diversas camadas de comunicação de sistemas abertos [TAN97].

contando com uma maior participação de todos os envolvidos, com diferentes visões do domínio, desde gerencial até operacional.

Voltando ao caso do setor de controle operacional, pode-se ilustrar a estruturação hierárquica dos requisitos das ações necessárias para o atendimento ao objetivo de gerenciamento das informações para o planejamento das obras de setorização.

Elicitadas as informações com os clientes e usuários, deduz-se que é necessário a realização de ações de coleta de quantitativos diários sobre o rendimento das equipes em suas atuações e de consolidação das informações sobre a execução de uma obra de setorização. Essas ações são metas a serem atingidas em prol da satisfação do objetivo que é o gerenciamento das informações para o planejamento das obras de setorização. Atenta-se que ambas as ações devem ser realizadas para o completo atendimento ao objetivo, ou seja, é necessário fazer a coleta de quantitativos diários para, então, consolidar esses dados em informações sobre a execução da obra. É um caso de composição de funcionalidade, representada pelo qualificador E entre os requisitos que instanciam as ações.

Para a realização da ação de coleta de quantitativos diários, a emissão de ordem de serviço para a equipe e a informação dos dados da ordem de serviço sobre a atuação diária da equipe em uma obra são executadas. Neste escopo, está-se detalhando uma ação, de nível abstrato mais alto, próximo à visão dos usuários, em outras duas ações de abstração mais próxima à especificação do software. Trata-se de níveis diferentes de requisitos do software.

Também, ocorre isso no detalhamento da consolidação das informações sobre a execução de uma obra de setorização em acumulação dos serviços realizados e materiais utilizados em uma planilha de dados e cálculo do custo total de serviços e materiais empregados na obra. Transpõe-se o limite entre os requisitos de usuário e os requisitos funcionais.

Essa abordagem hierárquica do modelo permite estruturar os requisitos em níveis verticais de abstração. Desde a definição dos requisitos de negócio, representados nos objetivos, até a especificação dos requisitos do usuário e funcionais, representados pelos requisitos propriamente ditos. A estruturação dos requisitos é traduzida em facilidade nas atividades de elicitação e modelagem, dada a abordagem segmentada. Ou seja, o tratamento de parte do domínio do problema desde o objetivo, em um grau de abstração apropriado a motivação do software, até a representação do requisito como uma funcionalidade requerida ao software.

A estruturação dos requisitos derivados do objetivo abordado nessa ilustração é construída pela composição das ações que representam esses requisitos (figura 5.6). A ação de emissão de ordem de serviço deve ser realizada juntamente com a de informação dos dados diários da equipe para que seja completa a ação de coleta de quantitativos diários. Os requisitos modelados por essas ações são, portanto, associados por um refinamento de composição, representados pelo qualificador E. Da mesma forma, são estruturados os requisitos de acumulação de serviços e de materiais e o cálculo do custo total, compondo a consolidação das informações de uma obra, pelo qualificador E.

Em um nível hierárquico mais abstrato, também ocorre uma composição representada pelas ações de coleta quantitativos e de consolidação das informações, necessária à contemplação do objetivo de gerenciamento das informações para o planejamento das obras de setorização.

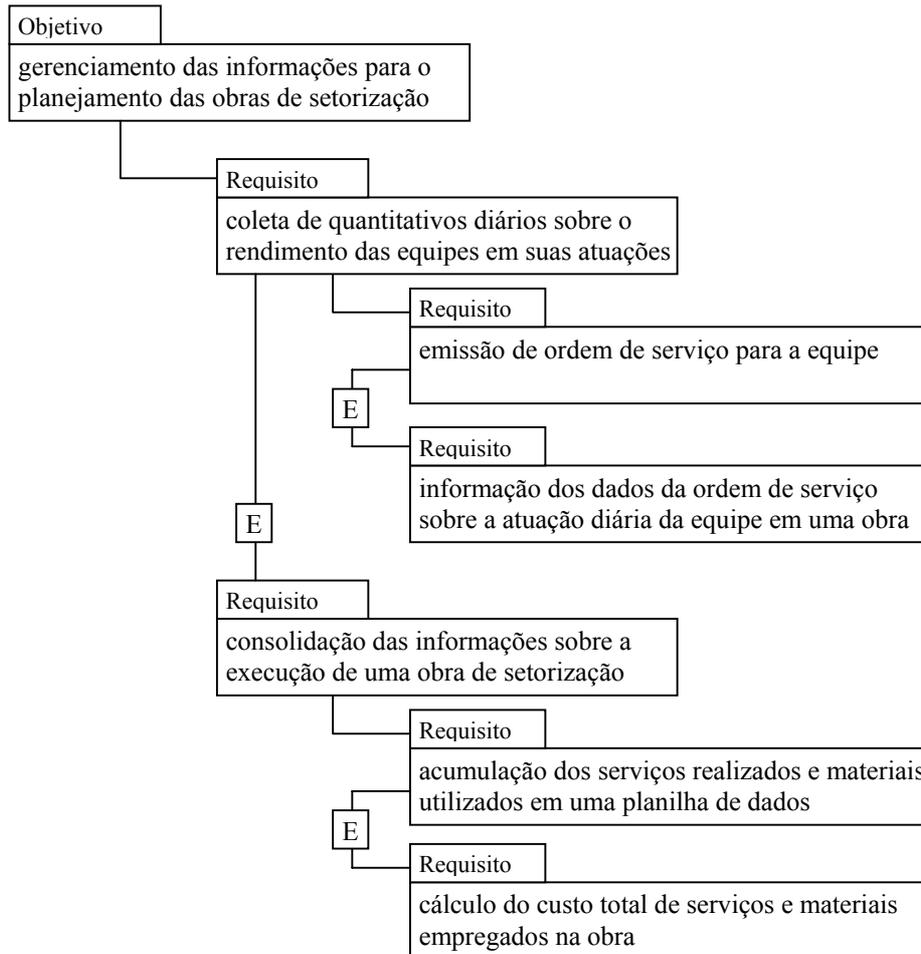


FIGURA 5.6 - Hierarquia dos requisitos do setor de controle operacional

Para exemplificar um caso de refinamento de alternativas de requisitos, pode-se sugerir que o requisito de cálculo de custo total de serviços e materiais empregados na obra possa ser realizado pela execução de uma ação de somatório das horas realmente gastas em cada serviço e da quantidade de material empregado, após a conclusão total da obra, ou pela ação de estimar o tempo total de cada serviço e a quantidade de material com base em dados diários obtidos da planilha de dados da obra em andamento. Nesse caso, os requisitos são estruturados hierarquicamente através de associações com qualificador OU, representando que o requisito de cálculo de custo total pode ser atendido pelo atendimento ao requisito de somatório final, após a conclusão, ou pelo atendimento ao requisito de estimativa no andamento da obra (figura 5.7).

### 5.3 Os Requisitos

No modelo, os requisitos representam os níveis de requisitos de usuário e de requisitos funcionais. Um exemplo desses diferentes níveis pode ser observado no caso abordado (figura 5.7). Em um nível de requisitos do usuário, há a representação da necessidade do usuário estar habilitado a realizar o cálculo do custo total de serviços e dos materiais empregados na obra. Deste requisito, deriva-se um requisito de nível

funcional, o somatório final do tempo gasto em serviços e quantidade de materiais empregados na obra. Este último, representando uma funcionalidade requerida para o software para capacitar o usuário a realizar sua tarefa.

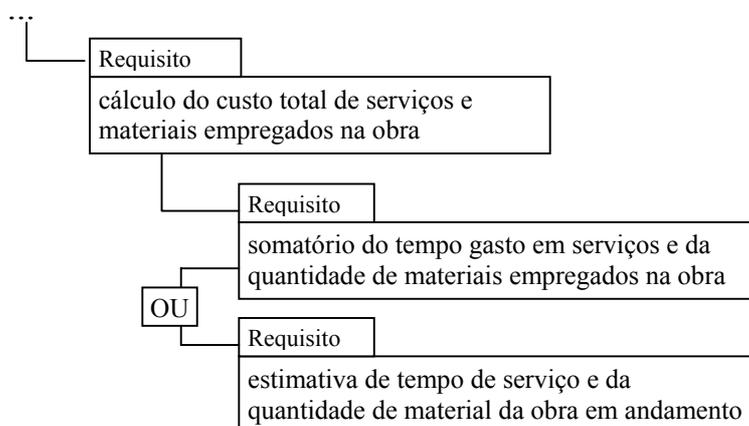


FIGURA 5.7 - Exemplo de hierarquia de requisitos alternativos

Os requisitos do usuário estabelecem um elo de ligação entre os requisitos do negócio, modelados pelos objetivos, e os requisitos funcionais. Estes últimos são os requisitos dos quais é elaborada a solução de software e constituem, portanto, a interface entre a definição do que desenvolver, para satisfazer os usuários, e os modelos que representam os objetos, a funcionalidade e o comportamento do software.

Segundo essa premissa, é desejável que a representação dos requisitos facilite tanto a identificação da origem dos objetivos que os determinou, como a construção de uma solução de software válida. Busca-se estabelecer uma correspondência acessível entre os objetos que compõem os requisitos e os elementos que representarão os modelos do software propriamente dito.

Nesta etapa do processo há uma identificação bem definida das necessidades, delimitada por um objetivo e estruturada por uma hierarquia de requisitos que representam diferentes abstrações do software, de um nível mais organizacional a um nível mais funcional. O propósito, agora, é definir precisamente os objetos dos requisitos a fim de que possam ser traduzidos em uma solução de software.

Considerando que neste ponto tem-se representada a visão que os clientes e usuários têm do que deve ser desenvolvido, presume-se que essa visão reflita aspectos de utilização do software. Essa análise é feita com base na presunção que as pessoas normalmente expressam os desejos através objetivos relacionados a tarefas necessárias para atendê-los, conforme já tratado na argumentação. Ressalta-se que essa argumentação professou a abordagem de modelagem por objetivos, com embasamento racional na Teoria da Atividade.

Assim sendo, o que encontra-se modelado até então é uma representação abstrata do comportamento desejado para o software. Esse comportamento é refletido por casos de uso, que tornar-se-ão subsídios para a modelagem dos aspectos de projeto do software requerido.

Situando o modelo em uma metodologia de desenvolvimento de software orientada a objetos, conforme os objetivos definidos para o trabalho em questão, é

natural que os requisitos sejam representados em uma notação apropriada a esse paradigma, no caso, a UML. Por isso, decidiu-se pela representação dos requisitos através de elementos que venham a facilitar a sua transformação em diagramas de caso de uso (*use cases*). Muito embora, essa representação de requisitos não somente tenha elementos de casos de uso mas, também, expõe outros objetos úteis para a atividade de modelagem do software.

Os casos de uso são utilizados para descrever os requisitos visíveis de um sistema [SCH98]. Descrevem um conjunto de ações realizadas para produzir um resultado observável a um usuário (alguém ou algo que interage com o sistema), e expressam a compreensão comum entre os envolvidos no processo de engenharia de requisitos. A modelagem por casos de uso visa representar a funcionalidade do sistema (o que o sistema deve fazer?), cuja estratégia pode ser caracterizada por acrescentar uma continuação à pergunta: para qual usuário? [JAC99].

As abordagens de casos de uso estão crescentemente atraindo a atenção na Engenharia de Requisitos porque a sua concepção centrada no ponto de vista do usuário é importante nas atividades que envolvem a modelagem de requisitos. A estruturação de casos de uso por diagramas *use case*, aliada à aplicação de abordagens com enfoque orientado a objetivos, facilita a compreensão do comportamento do sistema e a validação dos requisitos [LEE99].

### 5.3.1 Ação

Fundamentado nessa argumentação, o requisito é, então, representado por uma composição de ação, agente, recurso, produto e anotação (figura 5.8). A ação é o objeto da descrição do requisito em si, tal como é elicitada a informação entre os envolvidos. É o ponto de partida para a definição dos outros objetos que compõem o requisito. A denominação de ação indica que o requisito, no modelo, representa o nível de ação, conforme estabelecido pelo princípio da hierarquia da atividade.

A ação é um objeto que possui dois atributos: o nome e a descrição. O nome é o identificador da ação, como ela é reconhecida dentre os casos de uso do sistema. A descrição é, propriamente, a informação elicitada; tipicamente, uma narrativa como os envolvidos descrevem a ação requerida ao software.

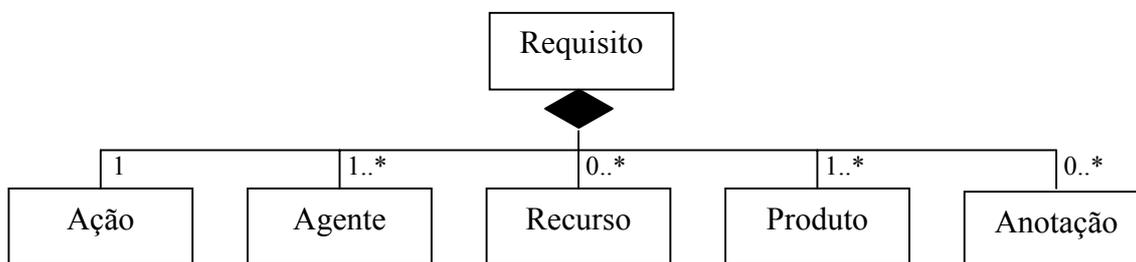


FIGURA 5.8 - Definição de um requisito

### 5.3.2 Agente

O agente é o solicitante da ação; quem age. É o ente que emite um estímulo requerendo a execução de alguma ação ao sistema. Cabe lembrar que a ação, segundo a Teoria da Atividade, é uma exteriorização que resulta em interação com os

objetos, de forma planejada, e, portanto, ocorre pela vontade de alguém. Esse alguém, no modelo, é abstraído pelo agente.

A revelação do agente é obtida da derivação do sujeito do objetivo. Na representação do objetivo são determinados os sujeitos que motivam a atividade. Como a atividade provê as ações, segundo o princípio hierárquico, os sujeitos do objetivo determinam os agentes dos requisitos. Assim, os agentes ou são instâncias de sujeitos do objetivo ou são subordinados aos mesmos. No caso de subordinação, entende-se que os agentes agem em prol de um objetivo, determinado por um sujeito.

Há uma multiplicidade de um ou mais agentes na composição estrutural do requisito. Isso indica que pode haver mais de um agente que pode solicitar a execução da mesma ação. O refinamento da ação pode representar agentes que atuam em sub-ações específicas.

### *5.3.3 Produto*

O princípio da orientação a objeto, da Teoria da Atividade, rege que a atividade é direcionada a um objeto, cuja transformação é movida a um motivo. Esse conceito é aplicado no nível das ações, que são orientadas a metas determinadas em proveito ao motivo da atividade. O resultado esperado da transformação do objeto da atividade pode ser segmentado em resultados obtidos pela execução das ações.

No contexto das ações, representadas por requisitos no modelo, busca-se a produção de um resultado no progresso da obtenção do objeto da atividade, representada pelo objetivo. Tem-se, então, o produto como elemento de representação do requisito. O produto é, portanto, um objeto produzido por uma ação realizada na criação, transformação ou destruição do objeto definido no objetivo. Desta maneira, os produtos do requisito são composições dos objetos do objetivo.

Entretanto, como os objetivos tratam níveis de abstração diferentes dos requisitos, os objetos são observados no domínio do problema, enquanto que os produtos são abstrações para o domínio da solução, passíveis de serem representados em modelos do software. Evidencia-se, aqui, o limite entre o domínio do problema, modelado por objetivos definidos em motivação à criação, transformação ou destruição de objetos do mundo real, e o domínio da solução, a realidade modelada por requisitos que prevêm a funcionalidade necessária para que o software atenda às metas desejadas.

Os produtos representam a capacidade produtiva de cada requisito definido para o software. No modelo, compõem o requisito com multiplicidade de um ou mais pois, assim como os objetos do objetivo, os produtos produzidos pelo atendimento ao requisito podem não ser únicos.

### *5.3.4 Recurso*

Na modelagem do objetivo, tem-se representado as ferramentas que são os recursos utilizados pelos sujeitos na criação, transformação e destruição dos objetos. No nível de requisitos, essas ferramentas são representadas por recursos. Os recursos são elementos de informação que compõem as ferramentas. Nota-se que, enquanto as ferramentas são representações de artefatos materiais ou de conhecimento presentes no mundo real, os recursos são abstrações desses materiais e conhecimentos utilizados no domínio da solução e representados por informações geridas pelo software.

A interpretação dada a um recurso, na representação do requisito, é de um objeto que corresponde à interface entre o agente e o software, ou a informação que o agente envia para os objetos do domínio do software para que possam executar

alguma ação em prol da produção de algum produto. Cabe ressaltar que, no processo de engenharia de requisitos, essa informação é definida sob o ponto de vista que os envolvidos têm do sistema; é uma abstração do usuário sobre o software.

Em um processo de modelagem, conforme a hierarquia do modelo, os recursos do requisito são composições das ferramentas das atividades. Representam uma visão das ferramentas em um nível abstrato adequado à modelagem dos requisitos do usuário e funcional. E, por apresentar o aspecto de composição, cada requisito identificado prevê uma composição estrutural com multiplicidade zero a muitos. Ou seja, pode não haver a necessidade de recurso para a execução da ação ou pode haver um ou mais recursos necessários para que a ação seja executada. O caso de não haver recursos representa a solicitação de execução de uma ação por parte do agente, sem a necessidade do mesmo informar algo, apenas emitindo o estímulo. A produção dar-se-á pelas informações presentes no domínio do software.

### *5.3.5 Anotação*

O modelo de requisito apresentado estrutura e representa requisitos funcionais. As características não funcionais do software não são tratadas nesta proposta. Porém, no processo de elicitação dos requisitos, essas características qualitativas, quantitativas e restritivas são capturadas e representadas por objetos de anotações. As anotações são, portanto, descrições de requisitos não funcionais que compõem um requisito modelado.

Conforme os objetivos estabelecidos, de o modelo prover a rastreabilidade para frente, e com o intuito de acompanhamento do desenvolvimento dos requisitos e validação do software, os apontamentos instanciados por anotações podem ser avaliados. Embora não haja previsão no processo de modelagem de algum procedimento sistemático, a verificação do atendimento aos requisitos não funcionais pode ser acompanhada, vinculada ao atendimento dos requisitos funcionais, porém dependendo da habilidade dos engenheiros de software.

### *5.3.6 Um Exemplo*

Para ilustrar a modelagem de requisitos, considera-se o requisito de coleta de quantitativos diários sobre o rendimento das equipes em suas atuações, do caso do setor de controle operacional (figura 5.9). Esse requisito aponta para a realização da ação de coletar a quantidade de serviços realizados e a quantidade de materiais empregados pela equipe, durante um dia, na realização de uma obra de setorização. Essa descrição, capturada na atividade de elicitação entre os envolvidos, é a própria descrição da ação que compõe o requisito. O nome da ação pode ser dado por "coleta de informações diárias", e deve ser a identificação desse requisito durante o desenvolvimento do software.

Identificada a ação, parte-se para a elucidação dos outros objetos que compõem o requisito. Essa tarefa é realizada em conjunto com os envolvidos, analisando e discutindo a ação específica. Trabalha-se sobre esse escopo, o que facilita a compreensão e entendimento mútuo.

O agente é o usuário para qual o requisito deve ser desenvolvido; o solicitante que faz com que a ação ocorra. Nesse exemplo, o agente é o chefe da equipe, pessoa responsável pela coleta das informações diárias da atuação de sua equipe. Analisando-se a definição do objetivo, nota-se que o agente está instanciado como sujeito. A interpretação desse fato é que o chefe da equipe participa da criação do objeto

custo da obra e, também, atua na sua produção através das informações prestadas que colaboram para a formação do custo da obra. O chefe da equipe informa em qual obra a equipe dele está trabalhando, todos os serviços que foram prestados no dia, com a quantidade de horas trabalhadas pelos funcionários em cada um deles, e todos os materiais utilizados, com o seu gasto em unidades.

Todavia, o modelo prevê a ocorrência de agentes não instanciados como sujeito. Um exemplo poderia ser dado no caso de haver um fornecedor materiais que providenciasse um catálogo com a descrição técnica de materiais, por solicitação do chefe da equipe. Neste caso, o fornecedor seria um colaborador, subordinado ao chefe da equipe, pois somente teria participação sobre o objeto (do objetivo) via chefe da equipe.

<b>Ação</b>	<b>Nome: coleta de informações diárias</b>
	<i>Descrição:</i> coletar a quantidade de serviços realizados e a quantidade de materiais empregados pela equipe, durante um dia, na realização de uma obra de setorização
<b>Agente</b>	chefe da equipe
<b>Produto</b>	identificação da obra de setorização identificação da equipe atuante e da data serviços executados e a quantidade de horas diárias de trabalho materiais empregados e o quantitativo diário de utilização
<b>Recurso</b>	ordem de serviço

FIGURA 5.9 - Representação do requisito "coleta de informações diárias"

O produto do requisito é o resultado da ação executada que vai criar, transformar ou destruir o objeto, e que direciona o objetivo. No caso ilustrado, o requisito de coleta de informações diárias prevê a funcionalidade para a composição do custo da obra. Nesta composição deve-se produzir a identificação da obra, a equipe que atuou, a enumeração dos serviços executados e a quantidade de horas diárias de trabalho necessárias para tal, e a listagem dos materiais necessários com os quantitativos diários empregados na realização da obra. Esse é o produto gerado, após o chefe da equipe coletar a quantidade de serviços realizados e a quantidade de materiais empregados pela equipe identificada, durante um dia, na realização de uma obra de setorização.

O produto do requisito vai compor o objeto do objetivo. Isto significa que, para ter-se o custo da obra é necessário ter-se a identificação da obra de setorização em questão, as equipes que atuaram, em quantos dias, os serviços executados e a quantidade de horas diárias de trabalho, e os materiais empregados e o quantitativo diário de utilização dos mesmos.

Para o agente fazer gerar o produto é necessário a utilização de recursos. O recurso utilizado pelo chefe da equipe para a produção da identificação da obra de setorização, da sua equipe, e da relação de serviços e materiais com seus quantitativos diários é a ordem de serviço. Neste nível de definição de requisitos, a ordem de serviço representa a informação que abstrai a ordem de serviço instanciada como ferramenta do objetivo.

A ordem de serviço do objetivo é o documento, em papel, utilizado no gerenciamento de informações para o planejamento das obras de setorização. É uma

instância do domínio do problema. Constitui a fonte de informação da qual obtém-se as informações necessárias para representar a ordem de serviço no domínio da solução. Esta ordem de serviço é a abstração requerida para o software daquela instância de ordem de serviço real. Representa um objeto que deve ser desenvolvido, em software, e que compõe o requisito modelado. Neste exemplo, o termo ordem de serviço identifica o mesmo objeto em níveis de abstração diferentes, do objetivo do sistema e do requisito de software.

Conforme a estrutura hierárquica dos requisitos do exemplo em questão (figura 5.6), para realizar-se a ação de coleta de quantitativos diários sobre o rendimento das equipes em suas atuações é necessário a realização de outras duas ações: a emissão de ordem de serviço e a informação dos dados diários da equipe. Cada uma dessas ações delimita a abordagem do sistema entre os envolvidos, o que vem a facilitar o processo de elicitação e modelagem dos requisitos pois trata um escopo menor, mais detalhado, da ação sobrejacente.

Na ação "emissão de ordens de serviço", tem-se a descrição de emitir ordem de serviço para o chefe da equipe coletar as informações dos quantitativos diários da atuação de sua equipe. Os outros objetos que compõem o requisito são o agente chefe da equipe, os produtos listas de serviços e materiais, e o recurso ordem de serviço (figura 5.10).

<b>Ação</b>	<i>Nome:</i> <b>emissão de ordem de serviço</b>
	<i>Descrição:</i> emitir ordem de serviço para o chefe da equipe coletar as informações dos quantitativos diários da atuação de sua equipe
<b>Agente</b>	chefe da equipe
<b>Produto</b>	lista de serviços lista de materiais
<b>Recurso</b>	ordem de serviço (padrão para preenchimento)
<b>Anotação</b>	impressão da ordem de serviço em uma folha de tamanho A4

FIGURA 5.10 - Representação do requisito "emissão de ordem de serviço"

O chefe da equipe é quem solicita que a ordem de serviço seja emitida e é, portanto, o responsável pela ocorrência da ação. É o mesmo que ocorre na ação sobrejacente.

Os produtos constituem uma enumeração de serviços possíveis de serem realizados pela equipe e uma lista de possíveis materiais empregados. São utilizados como indicação para o preenchimento dos quantitativos na ação conseqüente. Este produto é uma composição do produto da ação da qual essa é detalhada e é utilizado na formação do objeto do objetivo, ou seja, compõe o custo da obra.

O recurso, também, entra na composição do recurso da ação mais abrangente, a ordem de serviço. Observa-se que a ordem de serviço é o recurso utilizado para a coleta das informações diárias das equipes. Porém, nesta ação, o agente recebe uma ordem de serviço padrão (com os serviços e materiais enumerados, além de campo para preenchimento da identificação da equipe, data e obra na qual atuou) para que seja preenchida ao final da jornada. Este é o recurso utilizado para a interação entre agente e software, que representa parte da informação da abstração da ordem de serviços. Ou

seja, a informação padronizada que, após complementada com os dados identificadores e quantitativos, compõem o recurso ordem de serviço como um todo.

A anotação também compõe o requisito, indicando alguma característica não funcional que deva ser atendida. Um exemplo de tal poderia ser a restrição da ordem de serviço ser impressa em apenas uma folha no tamanho A4. Essa poderia ser uma imposição ao desenvolvimento do software, definida pelos envolvidos para execução da ação. No caso, viria a facilitar o manuseio do documento que deve ser preenchido durante o trabalho em campo.

A outra ação que complementa a emissão de ordem de serviço é denominada informação diária da equipe, cuja descrição é informar a obra na qual a equipe trabalhou, a quantidade de horas utilizadas em cada serviço e a quantidade de materiais empregados no dia de trabalho. Compõem este requisito o agente chefe da equipe, os produtos data, identificação da obra e da equipe e as quantidades de serviços e materiais, e o recurso ordem de serviço (figura 5.11).

<b>Ação</b>	<i>Nome:</i> <b>informação diária da equipe</b> <i>Descrição:</i> informar a obra na qual a equipe trabalhou, a quantidade de horas utilizadas em cada serviço e a quantidade de materiais empregados, no dia de trabalho
<b>Agente</b>	chefe da equipe funcionário
<b>Produto</b>	data identificação da obra identificação da equipe quantidade de serviços quantidade de materiais
<b>Recurso</b>	ordem de serviço (preenchida)

FIGURA 5.11 - Representação do requisito "informação diária da equipe"

O chefe da equipe é quem presta as informações diárias sobre a atuação da equipe e responsável pela realização da ação. Poder-se-ia ter outro agente colaborador, diferente do que ocorre na ação sobrejacente, o funcionário. Neste caso, o funcionário, subordinado à solicitação do chefe da equipe, daria alguma informação sobre o trabalho diário da equipe, por exemplo, o total de horas que ele mesmo trabalhou na realização de um serviço e a quantidade de materiais utilizados.

O produto é o resultado da ação. Constitui as informações prestadas pelo chefe da equipe, as quais servirão para compor o custo da obra. Lembrando que o custo da obra é o objetivo final (objeto) para o qual toda a atividade é motivada e, conseqüentemente, ações são realizadas em busca de seu atendimento. No caso específico da ação em questão, são as informações que vão complementar o produto da outra ação de mesmo nível hierárquico, que contribuirão na formação do custo da obra.

Compõem, então, o produto: a data, a identificação da obra de setorização, a identificação da equipe, e os valores de quantidade de serviços realizados e quantidade de materiais utilizados. A data será utilizada para o cálculo de dias necessários na conclusão de uma obra. A identificação da equipe servirá para a avaliação da produtividade. Os quantitativos estarão associados à identificação de cada

serviço e a cada material, que se complementarão na informação quantitativa da atuação da equipe no dia corrente.

O recurso, novamente, é a ordem de serviço. A ordem de serviço, neste escopo (das duas ações de mesmo nível hierárquico), é o instrumento que transporta a informação entre o agente e o sistema, na produção do produto e, conseqüentemente, na formação ou transformação do objeto do objetivo. Ressaltando-se, aqui, o aspecto dinâmico do objeto, ou melhor, a constante alteração do custo da obra conforme as ações vão sendo realizadas e novas informações da atuação das equipes vão sendo noticiadas diariamente.

## 5.4 Os Cenários

Em um processo de desenvolvimento participativo com uma abordagem voltada ao usuário, a instrumentação dos objetivos e requisitos é um fator importante para a validação do entendimento entre os envolvidos. Diante desta necessidade, os cenários são artefatos utilizados na Engenharia de Requisitos com propósitos diversos, desde a abstração de modelos conceituais até a identificação de casos excepcionais de comportamento do software [LAM98].

No contexto deste trabalho, os cenários são utilizados com um propósito conversacional, colaborando com a atividade de validação dos requisitos e análise do comportamento do software requerido. Os cenários têm sido utilizados com eficácia na comunicação entre os envolvidos [WEI98], descrevendo as motivações e experiências de clientes e de usuários, em exemplos concretos de situações e comportamento do sistema.

Ponderando sobre a Teoria da Atividade, os cenários descrevem operações executadas em um situação particular. Pelo princípio da hierarquia, os cenários cruzam o limite entre as ações intencionais e as operações condicionadas por respostas a situações correntes. Nos cenários estão representadas essas operações, que são executadas segundo um conjunto configurado de condições e descrevem os aspectos físicos da interface do usuário com o sistema.

Cenário é definido como uma seqüência temporal de interações de eventos entre diferentes agentes em um contexto restrito para obtenção de um propósito explícito [LAM98]. Os eventos são traduzidos em operações, e o propósito indica a meta da ação, que está modelada nos requisitos. O contexto do cenário descreve uma ação particular com início e fim bem determinados, enfocando conhecimento específico e ignorando outros irrelevantes. Esta característica facilita a interação entre os envolvidos, permitindo a decomposição do requisito, reduzindo a sua complexidade e contribuindo na identificação e tratamentos de situações normais ou de exceção do comportamento do sistema. Atenta-se para o fato que um requisito pode estar associado a vários cenários, cada qual ilustrando uma situação diferente de comportamento.

No modelo apresentado, os cenários são instanciados associados aos requisitos (figura 5.12). No processo de elicitação e modelagem, pode-se ter vários cenários operacionalizando um ou vários requisitos. Significa que as operações representadas no cenário podem contribuir para exemplificar um requisito, ou mais de um, dentro da estrutura hierárquica. Destacam-se, aqui, os cenários positivos e negativos, descrevendo, respectivamente, situações da execução que concluem a ação pretendida e atendem o requisito, ou situações de exceção, nas quais as ações não são completadas. Cabe ressaltar, ainda, que essas situações de exceção são comportamentos previstos de ocorrerem no sistema modelado, e não ocorrências anormais e imprevistas.

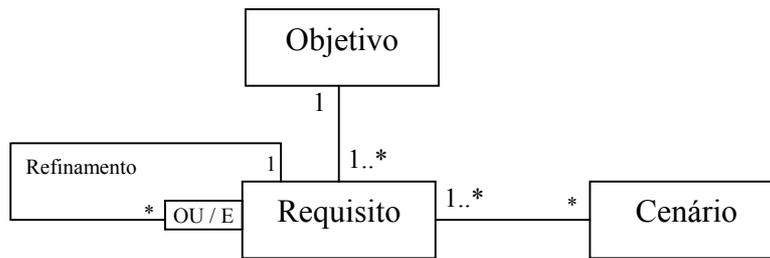


FIGURA 5.12 - Hierarquia dos requisitos com cenários

Os cenários são ferramentas visuais com um grau de informalidade e abstração apropriado à fase de definição dos requisitos. Descrevem de maneira inteligível o comportamento entre os agentes e contribuem como uma tipificação do modelo de requisitos. Essas características agregam aos cenários a qualidade de usabilidade, no sentido de apresentar-se como uma ferramenta que facilita a comunicação entre os envolvidos e a validação do que está sendo modelado.

A elaboração de cenários é um processo que consiste em observar uma particular e fragmentada instância de uma ação. O propósito é entender e representar o comportamento dos agentes que participam da ação, identificando as interações com o sistema e validando os objetos elicitados e modelados como requisitos (agente, produto, recurso).

Um cenário pode ser representado por um diagrama de interação [BOO94], que é uma generalização dos diagramas de eventos [RUM94] e dos diagramas de interações [JAC92]. Os diagramas de eventos compõem-se de linhas verticais que representam a linha do tempo de cada uma das instâncias dos agentes que participam do cenário, com os eventos sendo representados por setas horizontais. O tempo progride de cima, onde fica a representação do agente, para baixo, determinando a ordem de ocorrência dos eventos (figura 5.13). Essa representação gráfica constitui uma maneira amigável, de fácil compreensão e usabilidade na interação entre os envolvidos no processo definição dos requisitos do software.

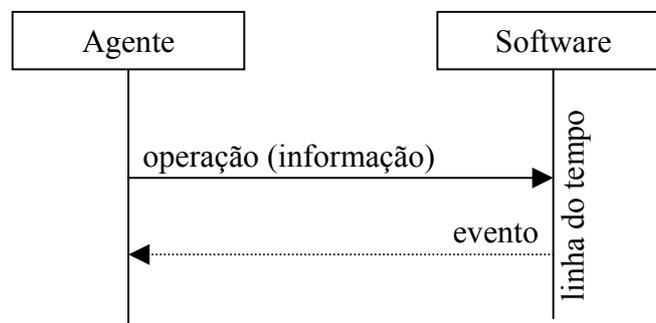


FIGURA 5.13 - Representação de um cenário

Entre os agentes há, pelo menos, um que representa o software. Pode haver vários agentes de software, representando diferentes componentes, na visão dos clientes e usuários. É importante ressaltar que os cenários, como ferramentas de

engenharia de requisitos, são modelados com elementos que representam uma visão abstrata do software, em um processo que conta com a participação de todos os envolvidos na atividade de definição dos requisitos.

As setas horizontais representam os eventos que ocorrem entre os agentes, sendo que a base da seta representa a sua origem. Os eventos que chegam aos agentes do software indicam a execução de uma operação. Essa concepção está intimamente ligada aos conceitos da orientação a objetos, na qual um evento é uma solicitação de um objeto cliente, neste caso, um agente, requerendo a execução de uma operação, provida por um objeto servidor. O objeto servidor é algum objeto que compõe a solução de software que, neste momento, é representado abstratamente pelo próprio software.

Segundo o Método Fusion [COL96], os eventos que partem de agentes externos para o software são chamados de operações, enquanto que os que partem dos objetos do software são chamados de eventos e representam os resultados da execução das operações. Os eventos resultado das operações são representados por setas pontilhadas. Essa semântica, utilizada no Fusion, é mantida em virtude da motivação do modelo em atender a uma demanda de requisitos para o método em questão.

Associando os cenários aos requisitos do modelos, tem-se que os agentes são representações diretas dos agentes do requisito, à exceção dos agentes do software que somente ocorrem nos cenários. Os recursos dos requisitos são representados nos cenários pelas informações dos parâmetros de operações e nos eventos resultado das operações. O significado é modelar a informação que o agente deve prestar ao software como subsídio para a execução da operação, indicado através dos parâmetros.

Os eventos são os efeitos desejados do software e representam os meios pelos quais são apresentados os produtos. É uma visão da criação e transformação do objeto do objetivo, instanciado por um recurso. Cada evento resultado das operações, constitui o produto das ações realizadas que, por sua vez, cria e transforma o objeto final, fruto do objetivo e motivador do desenvolvimento do software. O recurso é o meio que permite a visualização desse objetivo.

Acompanhando o caso do setor de controle operacional, pode-se desenvolver um cenário associado ao requisitos de coleta de informações diárias, e seus derivados: emissão de ordem de serviço e informação diária da equipe (figura 5.14).

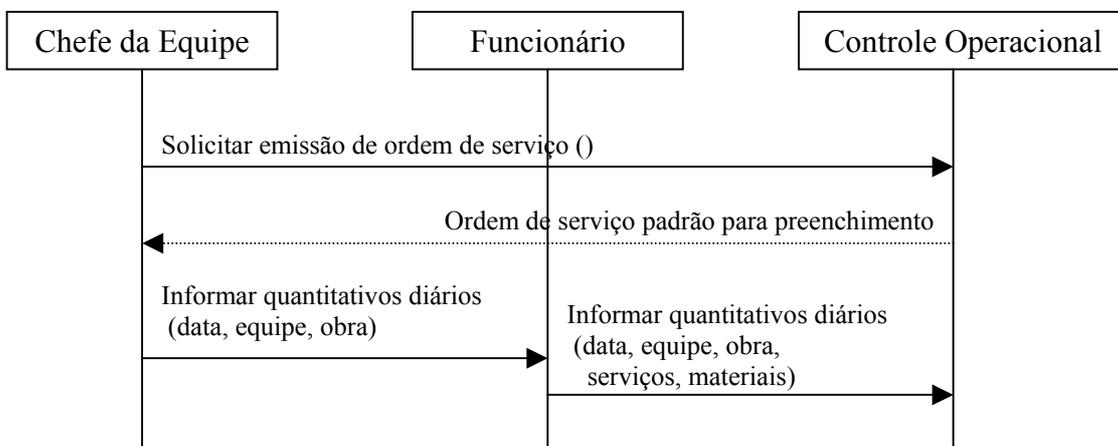


FIGURA 5.14 - Cenário do requisito "coleta de informações diárias"

O software é representado pelo agente controle operacional. A interação dos outros agentes com ele estabelece o limite entre os domínios do problema e do software, compondo a interface homem-computador.

Os agentes chefe da equipe e funcionário participam do cenário como atores, do mundo real, os quais requerem a execução de alguma ação. São os atores representados nos requisitos associados ao cenário. O chefe da equipe solicita a emissão de uma ordem de serviço padrão para ser preenchida com informações sobre as atividades diárias. Ao final da jornada, ele presta informações de data, identificação da equipe e da obra na ação de informação diária da equipe. O funcionário, neste cenário, é um colaborador do chefe da equipe nesta última ação, informando a quantidade de serviços e de materiais utilizados durante o dia no trabalho em uma obra.

O cenário inicia com a solicitação, pelo chefe da equipe, de emissão de ordem de serviço padrão para o preenchimento. Essa operação não contém parâmetros, é apenas uma requisição ao software de impressão do documento. O software responde com o evento de emissão da ordem de serviço para o chefe da equipe. A ordem de serviço é o recurso (do requisito de emissão de ordem de serviço) que contém o produto gerado pela ação, a lista de serviços e materiais disponíveis para o trabalho diário. A ordem de serviço emitida contém apenas uma relação de serviços possíveis de serem executados pelas equipes e uma relação dos materiais que podem ser utilizados na execução dos serviços. Observa-se que o recurso ordem de serviço é a instância do produto, a materialização que expressa a composição do objeto do objetivo: o custo da obra. Isto significa que os serviços e os materiais vão compor o custo da obra que, nesta instância comportamental do sistema, estão representados no objeto ordem de serviço.

Após o dia de trabalho da equipe, as informações desta atividade devem ser transcritas na ordem de serviço. Essa ação, de informação diária da equipe, é representada no cenário pela ação de informar quantitativos diários, realizada em dois momentos. Em um primeiro momento, o chefe da equipe informa a data, identifica a equipe e a obra na qual trabalha. Em um segundo momento, o funcionário da equipe informa os serviços executados e a quantidade de materiais empregados. Para essa ação, ambas operações utilizam o recurso ordem de serviço que, antes com apenas a lista de serviços e de materiais, é agora preenchida com as identificações e os quantitativos de serviços e de materiais.

A ordem de serviço preenchida é o recurso utilizado pelos agentes para gerar e gerir os produtos dos requisitos. Nas operações de informar quantitativos diários, os parâmetros representam as informações constantes no preenchimento da ordem de serviço. A ordem de serviço é, portanto, o recurso utilizado nesta operação específica para gerir o produto da ação de informação diária da equipe. Nota-se, nos parâmetros dessas operações, os produtos da ação associada. Em outras palavras, o produto é a informação que compõe o custo da obra e o recurso é o meio utilizado para transportar essa informação entre os agentes do sistema.

Cabe ressaltar que os produtos das ações de emissão de ordem de serviço e informação diária da equipe compõem os produtos da ação hierarquicamente superior, a coleta de informações diárias. Esses produtos tramitam entre os agentes do sistema através da ordem de serviço, que é o recurso modelado nas três ações em questão.

É importante observar, também, que os produtos representam as transformações do objeto custo da obra, que é composto pela relação de serviços executados e de materiais empregados, com seus respectivos quantitativos, de todas as equipes que trabalham na obra, ao longo de um período indicado pelas datas. Em um primeiro momento, instancia-se somente os serviços e materiais que compõem o custo

da obra. No momento seguinte, agrega-se quantitativos a esses insumos, que são acumulados por todas as equipes durante os dias em que realizaram-se trabalhos na obra, gerando, finalmente, o seu custo. A acumulação das informações e cálculo do custo da obra é especificado no requisitos de consolidação das informações sobre a execução de uma obra de setorização (figura 5.6).

Tem-se, aqui, a modelagem de uma atividade nos diversos níveis hierárquicos, desde a especificação dos objetivos, até a ilustração das operações, instanciadas através de cenários. Essa modelagem abstrai diferentes visões dos requisitos, em um segmento do domínio do problema bem determinado pela atividade, suas ações e operações. Essa metodologia facilita o processo de elicitação e modelagem, dada a argumentação básica da Teoria de Atividade: a atividade é a unidade básica de análise de situações, o contexto mínimo no qual a ação humana deve ser analisada para que possa fazer sentido e ser compreendida.

## 6 Rastreabilidade

A rastreabilidade é uma qualidade que diz respeito ao acompanhamento do ciclo de desenvolvimento do software, com a finalidade de facilitar sua validação. O objetivo é estabelecer um elo de ligação entre os objetivos e requisitos e os componentes de software produzidos.

O modelo apresentado busca prover a rastreabilidade para frente (*forward*), assim denominada por prover uma ligação entre os modelos, desde objetivos e requisitos aos modelos de elaboração, no sentido do ciclo de desenvolvimento do software. Esse elo estrutural vertical, do ponto de vista do ciclo de desenvolvimento, tem papel importante na validação do software. Em um nível mais alto tem-se os requisitos do negócio e dos usuários, que são associados às operações que compõem a solução do software. Cada operação implementada por algum componente executável pode ser validada conforme atendimento ou não ao requisito que a originou, através de uma análise da base para o topo (*bottom-up*) sobre essa estrutura vertical.

A concepção do modelo de requisitos fundamentada na atividade como unidade básica de análise do comportamento do sistema é uma aspecto que contribui diretamente para a validação. A validação é uma atividade que busca avaliar se o que está sendo desenvolvido é o que foi definido (“estamos produzindo o produto correto?” [PRE95]). Somente uma avaliação comparativa entre o que foi especificado como requisito e o que foi produzido em software pode dirimir essa questão.

A abordagem por atividades permite estruturar as necessidades desejadas pelos clientes e usuários, de maneira a refletir suas visões do que é esperado que seja realizado pelo software. Essa estruturação dos requisitos segundo os princípios da atividade contribui de maneira decisiva ao permitir a comparação entre o que foi definido com o que foi produzido. Permite associar a abstração dos clientes e usuários sobre os objetivos e requisitos com os componentes executáveis do software e, assim, validar estes últimos, sob o ponto de vista dos próprios clientes e usuários. O argumento é que a modelagem é realizada sob um domínio específico de atividade, definido pelo objetivo, que vai sendo detalhado em modelos de definição e implementação do software, o que facilita a abordagem e a inter-relação entre a funcionalidade do software e o sistema real como um todo.

A proposta é, então, definir vínculos conceituais que associam cada objeto do modelo de requisitos com os elementos que compõem os modelos de análise e de projeto do software. A partir dos modelos de análise e projeto, o software é implementado conforme uma metodologia determinada.

Este trabalho está inserido no contexto do Projeto FILM, o qual enfoca o Método Fusion. O Fusion é um método de desenvolvimento de software sistemático no que tange à consistência e amarração entre os modelos propostos. Essa qualidade é fundamental para guiar e acompanhar o processo de desenvolvimento, da etapa de análise à implementação do software. O modelo de requisitos apresentado propõe-se a estruturar e representar os requisitos em complementação ao Método Fusion, já que este último não aborda essa etapa no espectro de sua atuação. A sistematização do Método Fusion e a estruturação do modelo de requisitos proposto são aliados no provimento da qualidade de rastreabilidade. É objetivo estabelecer o elo de ligação entre os requisitos e os modelos de análise e projeto do Fusion visando orientar completamente o desenvolvimento do software, desde a definição dos objetivos pretendidos até a produção do software executável.

## 6.1 Casos de Uso

A associação hierárquica entre os requisitos, conforme apresentado no capítulo anterior, mantém a ligação entre os objetivos e os requisitos que especificam a funcionalidade requerida para o software. Os cenários ilustram esses requisitos em operações cujas seqüências temporais de execução ilustram uma instância comportamental do software.

Os requisitos e cenários representam o comportamento do software, sob o ponto de vista dos usuários. Pode-se, então, estabelecer um vínculo direto entre o modelo de requisitos e casos de uso (*use cases*), justificado pela abordagem da modelagem do sistema a partir da visão dos cliente e usuários que, no escopo da fase de análise, corresponde à modelagem estática do comportamento do software.

Considerando que os diagramas de casos de uso são ferramentas utilizadas para a modelagem estática do comportamento do software, a geração de diagramas de caso de uso a partir do modelo de requisitos proposto é importante para estabelecer a correlação direta entre a fase de definição requisitos e a fase de análise de outros métodos de desenvolvimento de software que utilizam notações da UML.

Tendo-se o modelo de requisitos estruturado, o objetivo é transformar os requisitos em casos de uso. Caso de uso é uma seqüência de ações que um ator (uma entidade externa ao software) realiza em um sistema buscando um objetivo particular. Diagramas de casos de uso são representações dos atores e os casos de uso, e são desenhados integrados a cenários [SCH98][ROS99]. Enquanto um cenário é uma seqüência de passos que descreve uma interação entre usuários e sistema, um diagrama de caso de uso representa um conjunto de cenários agrupados por um objetivo de usuário comum [FOW2000].

Segundo a concepção do modelo de requisitos, cada diagrama de caso de uso é derivado de um requisito de usuário, que representa uma ação a ser executada pelo software. Assim, para o desenho dos diagramas de caso de uso, deve-se partir da identificação de requisitos de usuário representados no modelo. Os requisitos derivados do selecionado, representados na hierarquia do modelo, e os cenários associados contribuem para o desenho do diagrama de caso de uso.

Identificado um requisito de usuário, utiliza-se toda a estrutura hierárquica do modelo que dele deriva para construir o diagrama de casos de uso. Todos os requisitos desta subestrutura geram casos de uso no diagrama, pois correspondem à funcionalidade que deve ser provida pelo software a fim de capacitar os usuários executarem as ações. O nome da ação, de cada requisito, determina o nome do caso de uso.

A organização dos casos de uso é especificada conforme os qualificadores que os associam no modelo. Os qualificadores E especificam relacionamentos de inclusão (*include*) entre os casos, e os qualificadores OU especificam relacionamentos de extensão (*extend*). Em ambos os relacionamentos, o caso de uso base é determinado pelo requisito ascendente na hierarquia do modelo. Os refinamentos dos requisitos, sem qualificador, representam detalhamentos de casos de uso.

Os relacionamentos de inclusão entre casos de uso significam que o caso de uso base incorpora explicitamente o comportamento do outro caso de uso [BOO2000]. O caso de uso incluído é instanciado como parte do que o inclui, e é utilizado como delegado na definição do comportamento do caso de uso base. Essa inclusão reflete a

acepção do qualificador E entre os requisitos do modelo, o qual representa uma complementação de ações a fim contemplar a ação (ou objetivo) subjacente.

Um exemplo de diagrama de casos de uso, com relacionamento de inclusão, pode ser obtido pela modelagem dos casos de uso do requisito "coleta de informações diárias", do caso Setor de Controle Operacional (figura 5.9). A estrutura hierárquica do modelo de requisitos do caso em questão (figura 5.6), define outros dois requisitos derivados desse abordado, associados através de qualificador E: "emissão de ordem de serviço" (figura 5.10) e "informação diária da equipe" (figura 5.11).

Cada requisito determina um caso de uso, identificado pelo nome da ação. O relacionamento entre eles é definido por inclusões; incluindo os casos de uso derivados no caso de uso base, que corresponde ao requisito hierarquicamente superior no modelo. O relacionamento de inclusão é representado por uma dependência, estereotipada como *include* (figura 6.1). O significado desse relacionamento indica que o comportamento representado pelas ações "emissão de ordem de serviço" e "informação diária da equipe" complementam, e são necessárias para a execução da "coleta de informações diárias".

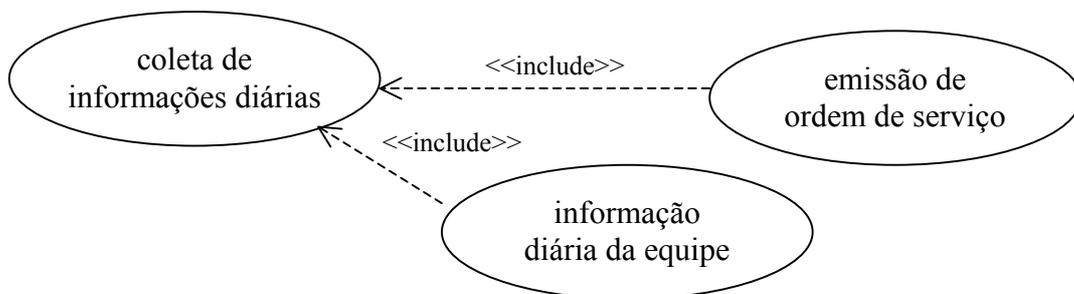


FIGURA 6.1 - Casos de uso do requisito "coleta de informações diárias"

Os relacionamentos de extensão entre casos de uso significam que o caso de uso base incorpora implicitamente o comportamento do outro caso de uso [BOO2000]. O caso de uso base, sob certas condições, tem seu comportamento estendido pelo comportamento de outro caso de uso, que pode ser considerado um comportamento opcional. Essa extensão reflete a acepção do qualificador OU entre os requisitos do modelo, o qual representa uma alternativa de ações a fim contemplar a ação (ou objetivo) subjacente.

Um exemplo de diagrama de casos de uso, com relacionamento de extensão, pode ser obtido pela modelagem dos casos de uso do requisito "cálculo do custo total de serviços e materiais empregados na obra", aqui nomeado de "cálculo de custo total da obra". A estrutura hierárquica do modelo de requisitos do caso em questão (figura 5.7) define outros dois requisitos derivados, associados através de qualificador OU e nomeados de "somatório final" e de "estimativa parcial".

O relacionamento entre esses casos de uso é definido por extensões, estendendo o caso de uso base aos casos de uso derivados, que correspondem aos requisitos hierarquicamente inferior no modelo. O relacionamento de extensão é representado por uma dependência, estereotipada como *extend* (figura 6.2). O significado desse relacionamento indica que o comportamento representado pela ação "cálculo de custo total da obra" pode ser, opcionalmente, representado pelas ações "somatório final" ou "estimativa parcial".

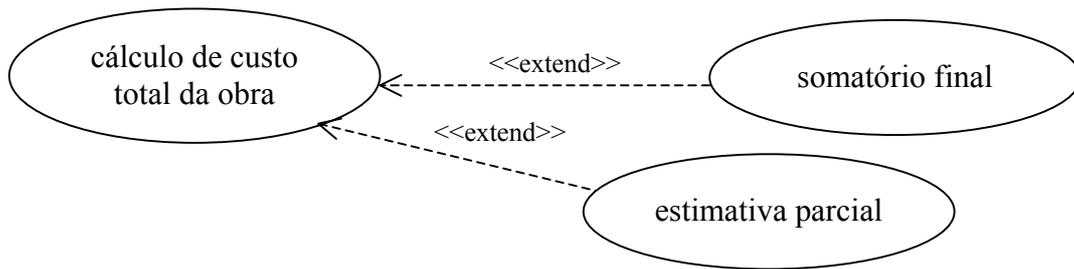


FIGURA 6.2 - Casos de uso do requisito "cálculo de custo total da obra"

Os agentes dos requisitos têm tradução direta para atores de casos de uso. Um ator representa um papel que os usuários ou entidades desempenham quando interagem com os casos de uso. Similarmente, o agente do modelo de requisitos é o solicitante que requer a execução de alguma ação no software e, portanto, interage de alguma maneira com ele. Na especificação do diagrama de casos de uso, os atores são associados a casos de uso, conforme os agentes são definidos nos requisitos. Para cada agente definido em um requisito, há um ator associado ao caso de uso gerado pelo requisito.

Complementando o caso de uso do requisito "coleta de informações diárias", os agentes são identificados conforme a modelagem dos requisitos (figuras 5.9, 5.10 e 5.11). Nestes modelos são definidos os agentes "chefe de equipe" e "funcionário". O agente "chefe de equipe" é instanciado em todos os requisitos dessa subestrutura, portanto, no diagrama de casos de uso é associado somente ao caso de uso de mais alto nível hierárquico. Essa simplificação é considerada em função dos relacionamentos existentes entre o caso de uso base e os seus derivados, obtidos pela associação hierárquica entre eles no modelo de requisitos. A própria organização dos casos de uso reflete essa interação do ator com todos os casos de usos relacionados com o caso de uso base. Já o agente "funcionário" é associado somente ao caso de uso cujo requisito correspondente o modela (figura 6.3).

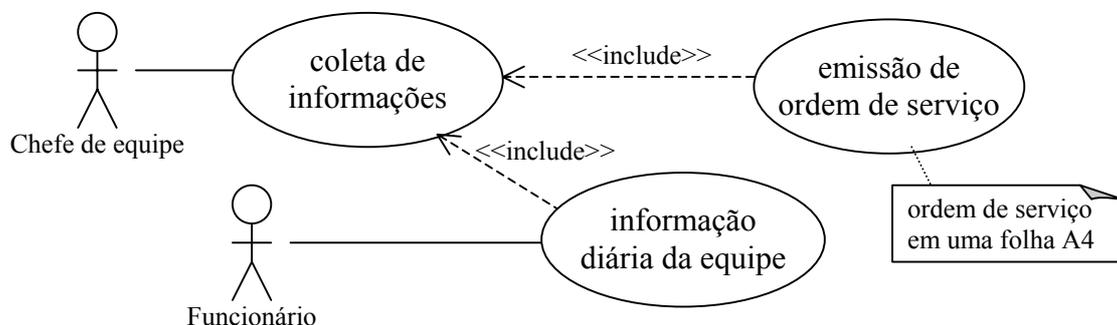


FIGURA 6.3 - Diagrama de casos de uso do requisito "coleta de informações diárias"

Os requisitos não funcionais, instanciados por anotações do modelo de requisitos, são representados em anotações, segundo notação da UML. Essas anotações, que são retratadas nos diagramas de casos de uso (figura 6.3), tem o propósito de

apontar a necessidade do atendimento de algum requisito de qualidade ou restrição de desenvolvimento do software.

Os diagramas de casos de uso derivados do modelo de requisitos devem ser complementados conforme o processo de desenvolvimento do software e as notações previstas na UML. A proposta apresentada propõe-se a determinar uma ligação básica entre modelos a fim de manter-se o vínculo dos requisitos definidos e com o diagrama de casos de uso. O objetivo é prover a rastreabilidade dos requisitos, estabelecendo uma correspondência entre o modelo de requisitos proposto e um modelo expresso em UML, amplamente utilizado para descrever os requisitos visíveis de um sistema.

## **6.2 Modelos do Método Fusion**

O Método Fusion estabelece um processo de desenvolvimento de software bastante sistemático, com procedimentos que guiam esse processo nas fases de análise, projeto e implementação. Essa sistematização colabora decisivamente com a rastreabilidade do software, conjugando os objetivos propostos neste trabalho e no Projeto FILM. O objetivo é dispor o modelo de requisitos como complemento ao Método Fusion, integrando-o ao processo, já que o método não prevê a modelagem dos requisitos.

O Método Fusion prevê uma completa dependência entre os modelos da análise, projeto e implementação (figura 6.4), segundo um processo de desenvolvimento provido de etapas de verificação e consistência, característica essa que contribui com a rastreabilidade. Os modelos definidos para a fase de análise são amarrados aos modelos de projeto, gerando os programas na fase de implementação. O documento de requisitos é o genitor do processo. Segundo os objetivos definidos, o modelo de requisitos deve tomar lugar do documento de requisitos e estabelecer uma dependência com os modelos da fase de análise.

Os modelos de análise do Método Fusion são o modelo de objetos e o modelo de interfaces. O propósito do modelo de objetos é capturar os conceitos do domínio do problema, bem como os relacionamentos existentes entre eles. O modelo de objetos descreve a estrutura do sistema, enquanto que o modelo de interfaces tem a finalidade de exprimir o seu comportamento. O modelo de interfaces é representado pelos modelo ciclo-de-vida e modelo de operações. O modelo ciclo-de-vida descreve o comportamento visto de uma perspectiva ampla, mostrando como o sistema se comunica com o seu ambiente desde o momento de sua criação até o seu término, enquanto que o modelo de operações descreve o comportamento individual de uma operação do sistema, definindo seus efeitos em termos de mudanças de estado e eventos gerados.

### *6.2.1 Modelo de Objetos*

O modelo de objetos é utilizados para fazer a modelagem da visão estática de um sistema, representando um conjunto de classes, atributos e seus relacionamentos. Uma classe é uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica, e implementa uma ou mais interfaces. As classes podem incluir abstrações que são parte do domínio do problema, como também representar elementos de software [BOO2000].

O modelo de requisitos colabora na definição das classes. Os recursos indicam fontes de informação que contém elementos que definem as classes. Os agentes são atores representados por classes e delimitam os limites do software, indicando as interfaces e as operações das classes. Os produtos também colaboram diretamente com a definição dos atributos das classes, pois definem composições do objeto final para qual o software é motivado.

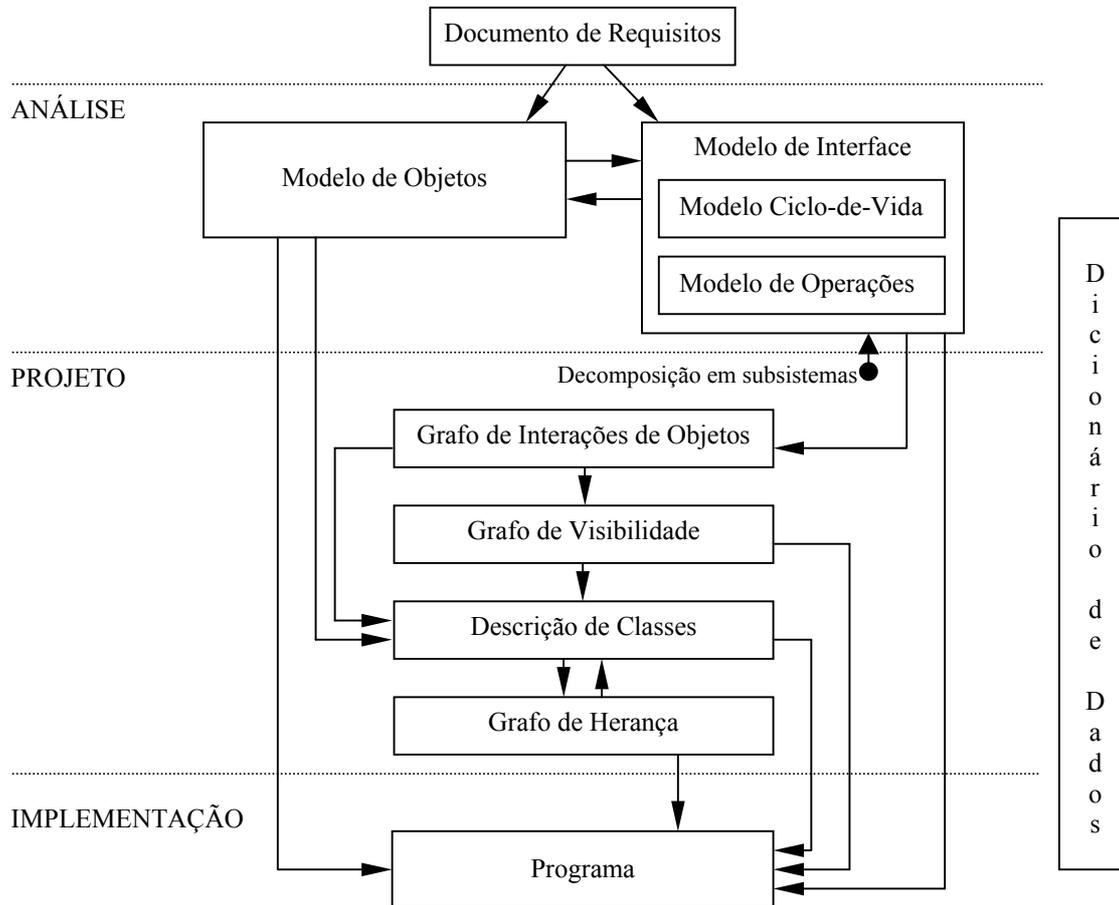


FIGURA 6.4 - O Método Fusion [COL96]

Para exemplificar esse inter-relacionamento entre o modelo de requisitos e o modelo de objetos, recorrem-se aos requisitos "emissão de ordem de serviço" (figura 6.5) e "informação diária da equipe" (figura 6.6), definidos no caso do setor de controle operacional. Nestes requisitos, tem-se o recurso "ordem de serviço". Esta ordem de serviço é uma abstração do documento utilizado para a realização das ações e constitui uma fonte de informação da qual pode-se obter a semântica e as características (atributos, operações e relacionamentos) dos objetos que vão determinar uma possível classe ordem de serviço. A definição completa das classes é resultado da etapa de análise (também pode haver classes de projeto), e as atividades neste sentido fazem parte do escopo desta etapa.

Os agentes são definidos como atores no modelo de objetos. O "chefe da equipe" e "funcionário" são classes que representam agentes externos ao software. Os relacionamentos dessas classes com outras classes do diagrama estabelecem a interface

do software, e delimitam as fronteiras do sistema. Os cenários também contribuem diretamente na definição dessas interfaces, haja visto que são associados aos requisitos e representam a interação dos agentes externos com o software.

<b>Ação</b>	<i>Nome: emissão de ordem de serviço</i> <i>Descrição: emitir ordem de serviço para o chefe da equipe coletar as informações dos quantitativos diários da atuação de sua equipe</i>
<b>Agente</b>	chefe da equipe
<b>Produto</b>	lista de serviços lista de materiais
<b>Recurso</b>	ordem de serviço (padrão para preenchimento)
<b>Anotação</b>	impressão da ordem de serviço em uma folha de tamanho A4

<b>Ação</b>	<i>Nome: informação diária da equipe</i> <i>Descrição: informar a obra na qual a equipe trabalhou, a quantidade de horas utilizadas em cada serviço e a quantidade de materiais empregados, no dia de trabalho</i>
<b>Agente</b>	chefe da equipe funcionário
<b>Produto</b>	data identificação da obra identificação da equipe quantidade de serviços quantidade de materiais
<b>Recurso</b>	ordem de serviço (preenchida)

FIGURA 6.6 - Reprodução do requisito "informação diária da equipe"

Os produtos, nesse requisito, são composições da ordem de serviço. Esses objetos do modelo de requisitos indicam atributos necessários para a definição das classes de objetos. A data, a identificação da obra, a identificação da equipe, a lista e quantidade de serviços, e a lista e quantidade de materiais geram atributos que esclarecem as classes do diagrama de classes. São informações que devem ser representadas a fim de que se chegue ao objetivo, isto é, a produção do objeto final para o qual o software é requerido.

O modelo de objetos gerado para o caso do setor de controle operacional (figura 6.7) demonstra essa associação entre os objetos do modelo de requisitos e os elementos do modelo de objetos. Os agentes "chefe da equipe" e "funcionário" são representados no modelo de objetos e delimitam os limites do sistema. São agentes que vão executar ações no sistema.

A "ordem de serviço", identificada como um recurso nos requisitos "emissão de ordem de serviço" e "informação diária da equipe" gerou uma classe *OrdemServiço* no modelo de objetos. É uma abstração no domínio da solução das

ferramentas definidas no objetivo e, portanto, modelada no software. Os produtos "data", "identificação da obra", "identificação da equipe", "lista de serviços", "lista de materiais", "quantidade de serviços" e "quantidade de materiais" vão compor o "custo da obra", objeto para qual o objetivo que originou o requisito faz-se necessário. Esses produtos são, portanto, representados no modelo estático, pois abstraem, também, elementos de informação do domínio da solução. No exemplo, os produtos estão relacionados a *OrdemServiço*, com atributos (*data*), objetos agregados (*ServiçoExecutado*, *MaterialEmpregdo*, *HorasTrabalhadas*) ou objetos associados (*Obra*, *Equipe*, *Serviço*, *Material*), razão pela qual o recurso é um objeto que corresponde à interface entre agentes e software para a execução de uma ação visando a produção de algum produto, que representa uma composição do objeto do objetivo.

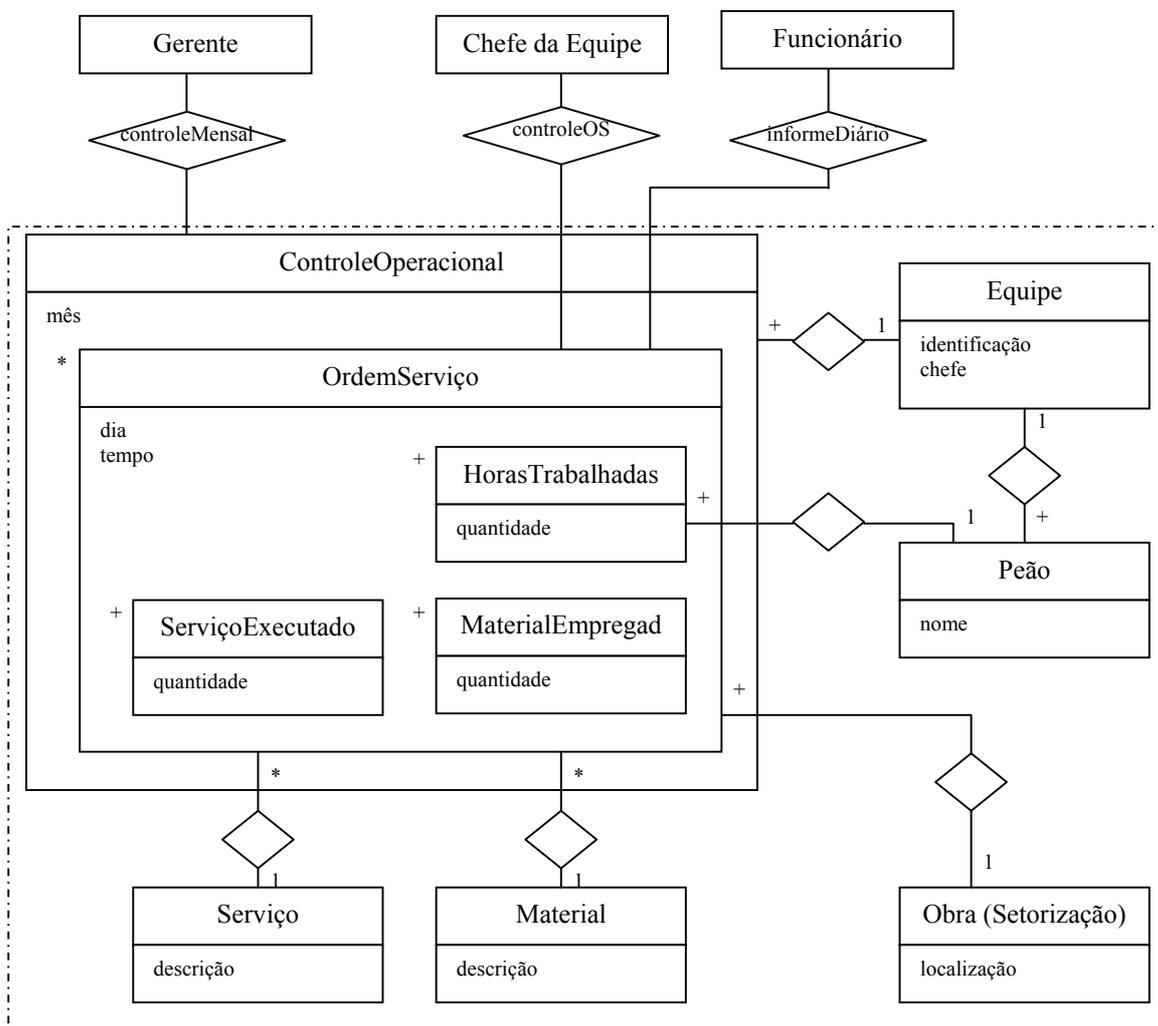


FIGURA 6.7 - Modelo de objetos do setor de controle operacional

Os agentes, recursos e produtos são objetos do modelo de requisitos que colaboram na geração do modelo estático do sistema. No caso do Método Fusion, contribuem para a confecção do modelo de objetos. Entretanto, como o modelo de requisitos, estruturado com base nas atividades, ações e operações, especificam o comportamento necessário do software sob o ponto de vista dos usuários, pode-se

estabelecer um vínculo mais afinado com os modelos de interface, assim como com os diagramas de caso de uso. O modelo de interface do Método Fusion é composto pelo modelo ciclo-de-vida e o modelo de operações.

### 6.2.2 Modelo Ciclo-de-vida

O modelo ciclo-de-vida tem a função de descrever o comportamento do sistema de uma perspectiva mais ampla, mostrando como o sistema comunica-se com o seu ambiente desde o momento de sua criação até o seu término. Uma expressão ciclo-de-vida define a seqüência permissível de interações das quais um sistema pode participar dentro de seu período de existência. A sua representação é através de expressões textuais que seguem uma sintaxe e uma semântica bem definidas, com um alfabeto, operadores e precedência, e substituições de expressões [COL96].

A descrição da expressão ciclo-de-vida parte da definição dos objetivos. A hierarquia do modelo de requisitos contribui com as substituições na descrição do ciclo-de-vida. Requisitos associados por qualificadores E denotam ocorrências seqüenciais, representadas na expressão pelo operador ponto (.). Os requisitos associados por OU denotam ocorrências opcionais, representado pelo operador barra (|). Os cenários associados aos requisitos colaboram na definição dos eventos de saída, representados pelos retornos das operações do modelo ciclo-de-vida, precedidos pelo símbolo #.

Um exemplo pode ser obtido a partir da composição hierárquica do modelo de requisitos do caso do setor de controle operacional (figura 6.8). O objetivo no topo da hierarquia - gerenciamento das informações para o planejamento das obras de setorização - define a expressão ciclo-de-vida *GerenciamentoCustoObra*, da qual são derivadas as substituições conforme a composição hierárquica dos requisitos. Pode haver mais de um objetivo para o mesmo sistema, que devem ser representados na expressão ciclo-de-vida. A determinação dos operadores que ligam os objetivos, bem como o número de ocorrências e opcionalidade de cada objetivo, faz parte da etapa de análise do método.

O modelo ciclo-de-vida do objetivo *GerenciamentoCustoObra* (figura 6.9) deriva diretamente da estrutura hierárquica de seus requisitos (figura 6.8). Em um primeiro nível há dois requisitos associados pelo qualificador E: "coleta de informações diárias" e "consolidação das informações". Cada um desses requisitos, nomeados no modelo pelas expressões *ColetaInformaçõesDiárias* e *ConsolidaçãoInformações*, é representado associado pelo operador de seqüência e pode ser substituído por novas expressões que revelam um detalhamento conforme a hierarquia do modelo.

O requisito "coleta informações diárias" (*ColetaInformaçõesDiárias*) é detalhado pela seqüência de *EmissãoOrdemServiço* e *InformaçãoDiáriaEquipe*, as quais correspondem aos requisitos, associados pelo qualificador E, "emissão de ordem de serviço" e "informação diária da equipe", respectivamente.

O requisito "consolidação das informações" (*ConsolidaçãoInformações*) é detalhado pela seqüência de *AcumulaçãoQuantitativos* e *CálculoCustoTotalObra*, as quais correspondem aos requisitos, associados pelo qualificador E, "acumulação dos serviços e materiais" e "cálculo do custo total da obra", respectivamente.

O requisito "cálculo de custo total da obra" (*CálculoCustoTotalObra*) é substituído por outros dois requisitos associados pelo qualificador OU - "somatório final dos serviços e materiais" (*SomatórioFinal*) ou "estimativa parcial de serviço e material" (*EstimativaParcial*) - os quais são expressos associados pelo operador de ocorrências opcionais.

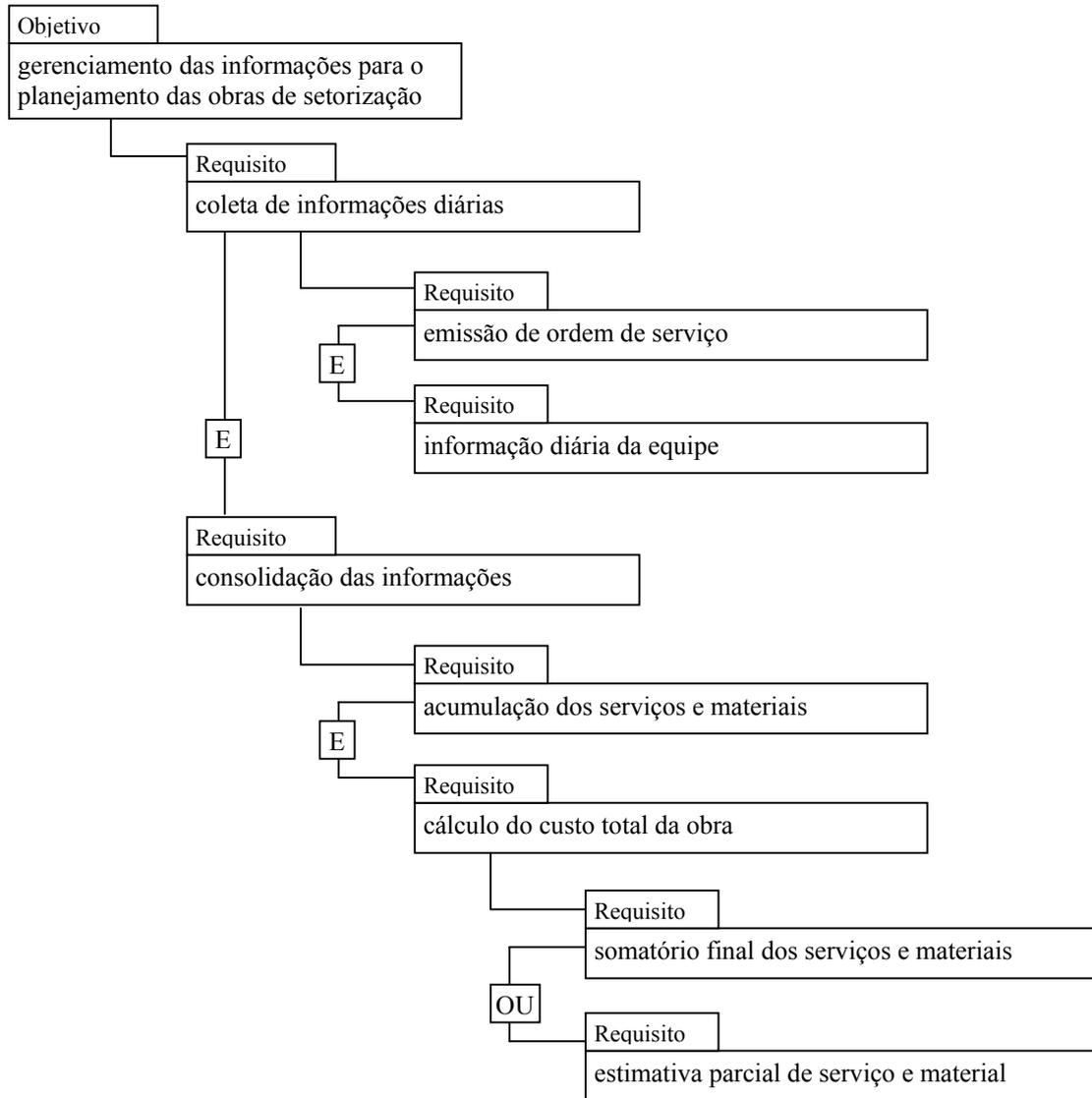


FIGURA 6.8 - Estrutura hierárquica dos requisitos do setor de controle operacional

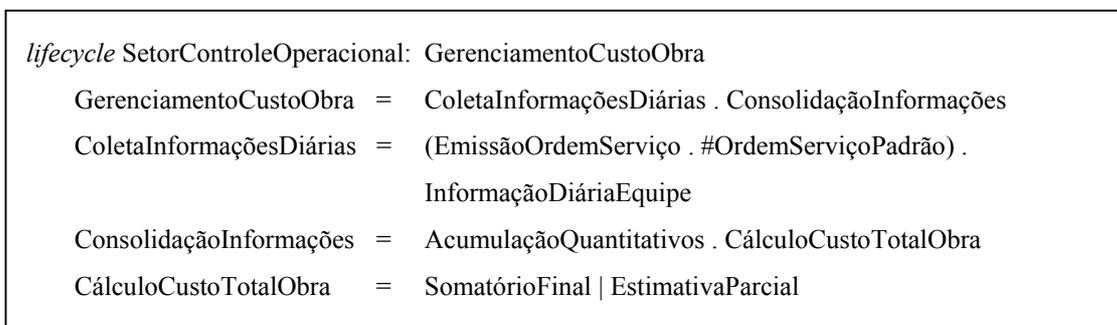


FIGURA 6.9 - Modelo ciclo-de-vida do setor de controle operacional

Os eventos de saída (precedidos pelo símbolo #) são obtidos dos cenários associados aos requisitos em questão. Para tal, deve-se analisar as operações que ilustram os requisitos e observar os eventos que delas derivam, isto é, que retornam dos agentes do software para os agentes dos requisitos. Um exemplo é a "ordem de serviço padrão para preenchimento", evento de saída da operação que descreve o requisito "emissão de ordem de serviço" (*EmissãoOrdemServiço*), nomeada pela expressão *OrdemServiçoPadrão* (figura 6.9).

Os cenários dão ênfase à ordenação temporal das mensagens. Mostram um conjunto de objetos e as interações realizadas entre eles ao longo de um período de tempo, indicado pelos envios e recebimentos de mensagens. Os cenários, no Método Fusion, são técnicas utilizadas para auxiliar a definição da interface do sistema. Eles são derivados diretamente dos cenários do modelo de requisitos, considerando que a forma e a semântica de ambos são similares, com exceção do nível de abstração que tratam os objetos.

### 6.2.3 Modelo de Operações

As expressões do modelo ciclo-de-vida que não são detalhadas no próprio modelo são descritas como uma operação no modelo de operações, conforme processo do Método Fusion. No modelo apresentado (figura 6.9), por exemplo, as expressões *EmissãoOrdemServiço* e *InformaçãoDiáriaEquipe* devem ser descritas como operações do modelo de operações.

O modelo de operações tem a função de descrever o comportamento de uma operação individual do sistema, definindo seus efeitos em termos de mudança de estado e eventos gerados. Para cada operação é determinado um esquema por textos estruturados com uma sintaxe bem definida, nos quais os itens que definem a operação são organizados em cláusulas [COL96].

Cada operação é descrita pelas cláusulas *operation*, *description*, *reads*, *changes*, *sends*, *assumes* e *result*. A cláusula *operation* aponta um nome que será o identificador único da operação do software. A cláusula *description* apresenta uma descrição informal e concisa da operação. A cláusula *reads* introduz uma lista de itens que constituem parâmetros (precedidos pela palavra *supplied*), objetos, atributos e relacionamentos que podem ser acessados pela operação. A cláusula *changes* introduz uma lista de itens que constituem objetos, atributos e relacionamentos que podem ser criados (precedidos pela palavra *new*) ou alterados pela operação. A cláusula *sends* apresenta uma lista de agentes e os eventos que a operação lhes envia. A cláusula *assumes* define pré-condições para a execução da operação e a cláusula *result* define as pós-condições após a execução da operação.

O modelo de operações, segundo a abordagem proposta, é derivado dos cenários e requisitos associados. Para cada conjunto de cenários associados aos mesmos requisitos é gerado um esquema de uma operação. Esse esquema descreve a funcionalidade que a operação deve prover. Sugere-se que o nome da operação no esquema seja o mesmo nome indicado no modelo ciclo-de-vida da operação inicial do cenário, dado que a operação é uma solicitação de um agente para a realização de alguma ação no software. Assim, pode-se definir as operações identificadas por *EmissãoOrdemServiço* (figura 6.11), correspondente ao requisito "emissão de ordem de serviço" (figura 6.5), e *InformaçãoDiáriaEquipe* (figura 6.12), correspondente ao requisito "informação diária da equipe" (figura 6.6). Essas operações ocorrem em seqüência, detalhando a ação *ColetaInformaçõesDiárias* (figura 6.9), conforme descrito no cenário do requisito "coleta informações diárias" (figura 6.10).

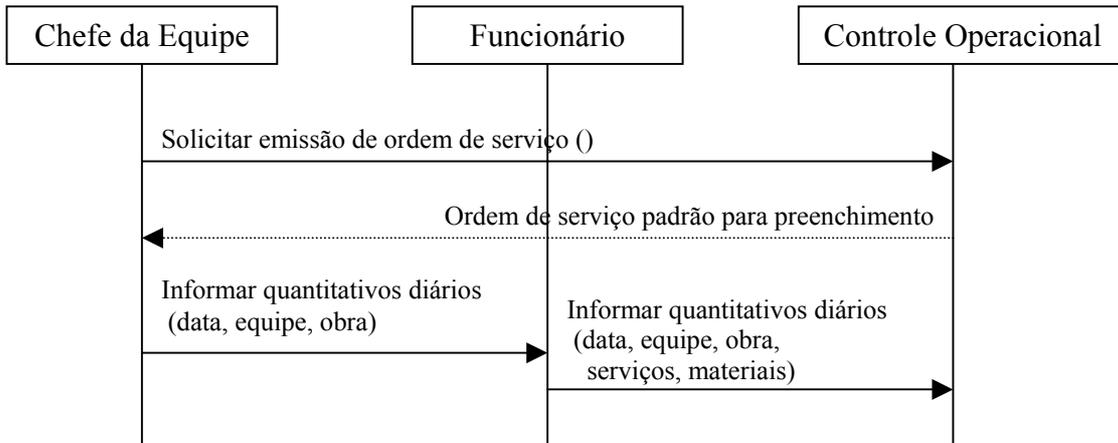


FIGURA 6.10 - Reprodução do cenário do requisito "coleta de informações diárias"

A descrição da operação, cláusula *description*, é um texto que advém da descrição dos requisitos associados ao cenário. A descrição do modelo de requisitos, ou de um resumo da mesma, facilita a narração do que a operação deve prover e, conseqüentemente, contribui para a sua validação. Isso deve-se pelo fato que a descrição dos requisitos é obtida durante a elicitacão dos usuários e expõe a visão que estes têm da funcionalidade desejada para o software.

Os parâmetros das operações são representados no modelo de operações na cláusula *reads*. Como esses parâmetros são fornecidos pelos agentes para que a operação seja realizada, eles são precedidos pela palavra reservada *supplied*, conforme estabelecido no método. No exemplo da operação *InformaçãoDiáriaEquipe* (figura 6.12), são passados os parâmetros "data", "equipe", "obra", "serviços" e "materiais", representados no cenário (figura 6.10) e introduzidos na cláusula *reads*. Também são introduzidos, na cláusula *reads*, os objetos acessados pela operação, por exemplo, *Serviço* e *Material* na operação *EmissãoOrdemServiço* (figura 6.11), os quais representam a "lista de serviços" e "lista de materiais", definidos como produto do requisito "emissão de ordem de serviço" (figura 6.5).

Já a cláusula *changes* introduz os objetos criados ou alterados pela operação. Na operação *EmissãoOrdemServiço* (figura 6.11), uma nova instância de ordem de serviço é criada, a qual é entregue ao chefe da equipe, conforme descrito no requisito e representado no cenário. Na operação *InformaçãoDiáriaEquipe*, os dados da ordem de serviço são atualizados, conforme definido no requisito, tendo, portanto, o objeto *OrdemServiço* expresso na cláusula *changes* da operação (figura 6.12).

Os eventos dos cenários representam as mensagens que são enviadas aos agentes. Correspondem ao texto descrito na cláusula *sends* do modelo de operações. Nesta cláusula é descrito o agente que recebe a mensagem seguido da mensagem em si. O agente está representado no cenário, enquanto que a mensagem é o evento que parte do agente de software para o agente externo, em resposta à operação solicitada. No exemplo da operação *EmissãoOrdemServiço* (figura 6.11), o agente "chefe da equipe" recebe a ordem de serviço padrão para preenchimento, que contém a lista de serviços e lista de materiais e é representada na cláusula *sends* por *OSpadrão*.

As cláusulas *assumes* e *results* definem pré e pós-condições da execução da operação e são tratadas durante a fase de análise, conforme o Método Fusion.

<i>Operation:</i>	EmissãoOrdemServiço
<i>Description:</i>	Emitir ordem de serviço para o chefe da equipe coletar as informações dos quantitativos diários da atuação de sua equipe.
<i>Reads:</i>	Equipe, Serviço, Material
<i>Changes:</i>	new OrdemServiço
<i>Sends:</i>	Chefe da Equipe: {OSpadrão}
<i>Assumes:</i>	
<i>Result:</i>	Chefe da Equipe recebe OS padrão para preenchimento, com identificação da equipe, lista de serviços e lista de materiais.

FIGURA 6.11 - Esquema da operação *EmissãoOrdemServiço*

<i>Operation:</i>	InformaçãoDiáriaEquipe
<i>Description:</i>	Informar a obra na qual a equipe trabalhou, a quantidade de horas utilizadas em cada serviço e a quantidade de materiais empregados, no dia de trabalho.
<i>Reads:</i>	supplied data, equipe, obra, serviço, materiais Equipe, Obra, Serviço, Material
<i>Changes:</i>	OrdemServiço
<i>Sends:</i>	
<i>Assumes:</i>	OrdemServiço já instanciada (emitida)
<i>Result:</i>	OrdemServiço é atualizada com os dados da jornada e é fechada.

FIGURA 6.12 - Esquema da operação *InformaçãoDiáriaEquipe*

#### 6.2.4 Usabilidade

A ligação do modelo de requisitos com os modelos de objetos e de interface do Fusion, permite estabelecer um vínculo entre o documento de requisitos e a fase de análise, complementando uma lacuna do método, conforme objetivo deste trabalho. A amarração entre os modelos da fase de análise e os da fase de projeto, estabelecida na sistematização do Fusion, associa os componentes de software projetados a suas definições. Assim, tem-se uma estrutura de software que permite o rastreamento dos requisitos definidos no modelo ao longo do processo de desenvolvimento.

Partindo de um objetivo, navegando pela estrutura dos seus requisitos derivados e pelas ligações com o modelo de interfaces do Fusion, é possível identificar quais as operações atendem a necessidade especificada. Seguindo no processo Fusion e acompanhando a dependência entre os seus modelos, pode-se determinar quais componentes de software implementam a funcionalidade prevista. Esse vínculo entre objetivo e software contribui sumariamente para a atividade de validação, haja visto que os objetivos e requisitos são descritos segundo uma abordagem voltada aos usuários e, portanto, expressam suas expectativas e visões do sistema, facilitando a percepção entre *o que foi definido* e *o que o software está provendo*.

A contribuição do modelo de estruturação de requisitos, neste aspecto, reside na qualidade de usabilidade. A usabilidade diz respeito à rastreabilidade dos requisitos no processo de desenvolvimento do software, culminando com o vínculo entre o requisito a operação implementada. Neste contexto, o modelo é uma ferramenta utilizada na atividade de validação do software em um esforço conjunto entre desenvolvedores, clientes e usuários.

### 6.3 Ferramenta de Suporte ao Modelo

O modelo de estruturação de requisitos é fundamentado sobre um modelo de objetos que representa a estrutura de objetivos, requisitos, cenários e suas associações (figura 6.13). A partir desse modelo e considerando a abordagem de estabelecimento dos requisitos com base nos princípios da atividade, é possível definir requisitos para o desenvolvimento de uma ferramenta que dê suporte a estruturação dos requisitos conforme a proposta apresentada.

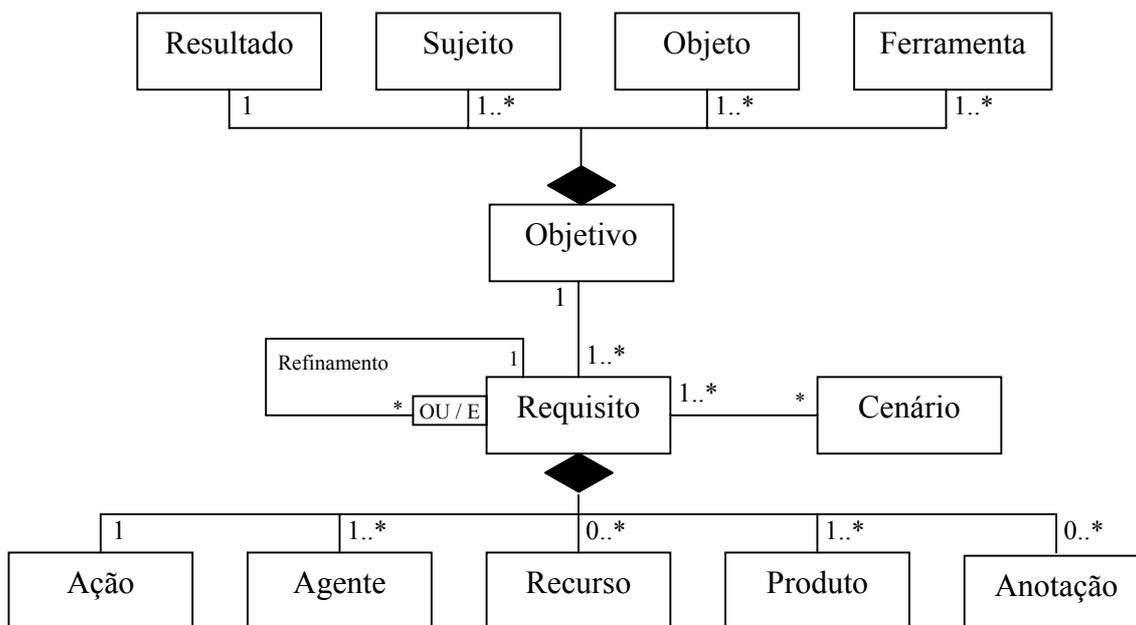


FIGURA 6.13 - Modelo de estruturação de requisitos

Pode-se determinar como objetivo a geração de diagramas de casos de uso (*use case*), com o intuito de manter o vínculo dos requisitos com um modelo de definição de comportamento do software provido pela UML. Cabe observar que um dos objetivos do Projeto FILM é adaptar o método às notações UML. Também, já foi comentado que os diagramas de caso de uso são amplamente utilizados para descrever os requisitos visíveis de um sistema em metodologias de desenvolvimento orientada a objetos que utilizam os diagramas da UML.

Neste contexto, pode-se definir dois objetivos para o projeto de uma ferramenta que suporte o modelo proposto: "especificação do modelo de estruturação de requisitos" (figura 6.14) e "geração dos diagramas de casos de uso" (figura 6.15).

<b>Objetivo</b>	<i>Especificar o modelo de estruturação de requisitos tal que organize e represente os objetivos, requisitos e cenários os quais definem as necessidades do sistema.</i>
<b>Resultado</b>	<i>Qual o resultado esperado? Por que há a necessidade?</i> especificação do modelo de estruturação de requisitos
<b>Objeto</b>	<i>Para o que esse objetivo faz-se necessário ao sistema?</i> modelo de estruturação de requisitos
<b>Sujeito</b>	<i>Quem atua em prol desse objetivo?</i> engenheiro de requisitos
<b>Ferramenta</b>	<i>Quais são os recursos necessários para os sujeitos interagirem com o objeto?</i> conhecimento do universo de informação (informações elicitadas)

FIGURA 6.14 - Objetivo de especificação do modelo de estruturação de requisitos

<b>Objetivo</b>	<i>A partir do modelo de estruturação de requisitos, gerar os diagramas de casos de uso que descrevem a estrutura comportamental do sistema.</i>
<b>Resultado</b>	<i>Qual o resultado esperado? Por que há a necessidade?</i> geração dos diagramas de casos de uso
<b>Objeto</b>	<i>Para o que esse objetivo faz-se necessário ao sistema?</i> diagramas de caso de uso
<b>Sujeito</b>	<i>Quem atua em prol desse objetivo?</i> engenheiro de requisitos
<b>Ferramenta</b>	<i>Quais são os recursos necessários para os sujeitos interagirem com o objeto?</i> modelo de estruturação de requisitos conhecimento do domínio do problema

FIGURA 6.15 - Objetivo de geração dos diagramas de caso de uso

Para cada objetivo são estruturados os requisitos que dele derivam (figura 6.16). Do objetivo de "especificação do modelo de estruturação de requisitos", derivam-se os requisitos: "definir objetivo" (figura 6.17) e "definir estrutura do objetivo" (figura 6.18), "definir requisito" (figura 6.19) e "definir cenário" (figura 6.20), "definir glossário de termos" (figura 6.21). Do objetivo de "geração dos diagramas de caso de uso", derivam-se os requisitos: "definir casos de uso" (figura 6.22) e "definir relacionamento entre casos de uso" (figura 6.23) e "definir atores associados aos casos de uso" (figura 6.24).

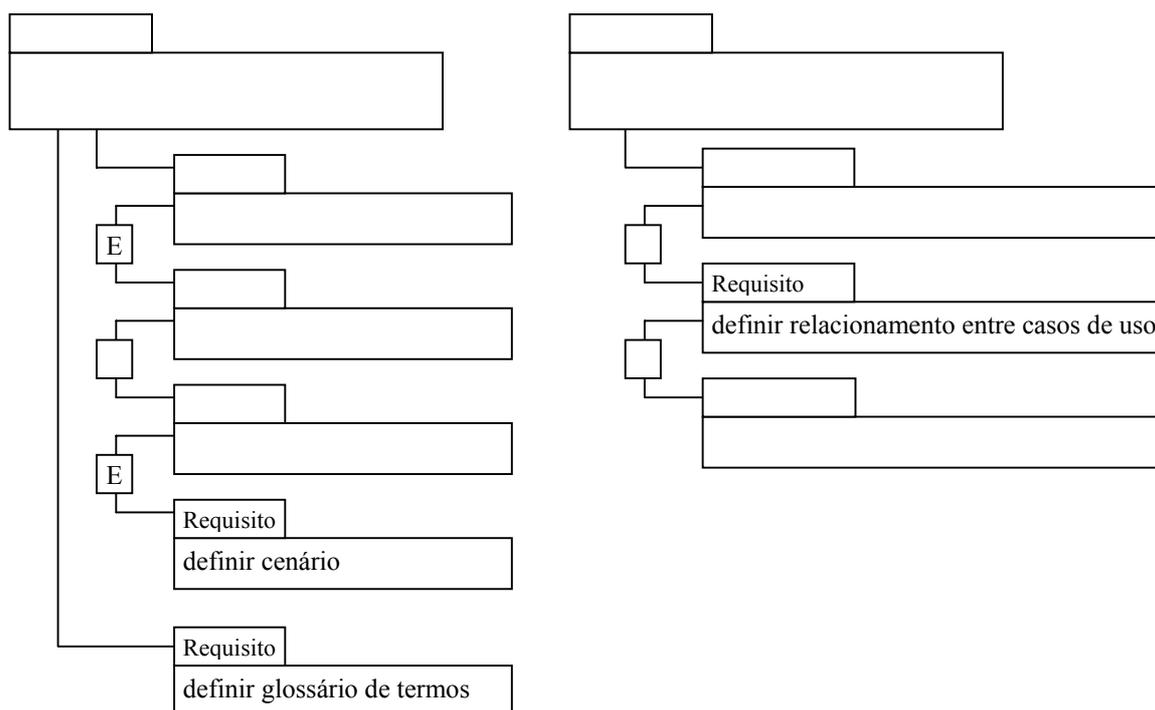


FIGURA 6.16 - Estrutura hierárquica dos requisitos da ferramenta de suporte

<b>Ação</b>	<i>Nome: definir objetivo</i> <i>Descrição: definir um objetivo do sistema (atividade), especificando resultado, objeto sujeito e ferramenta</i>
<b>Agente</b>	engenheiro de requisitos
<b>Produto</b>	modelo de definição do objetivo
<b>Recurso</b>	informação elicitada
<b>Anotação</b>	conforme definido no modelo de estruturação de requisitos

FIGURA 6.17 - Requisito "definir objetivo"

<b>Ação</b>	<i>Nome: definir estrutura do objetivo</i> <i>Descrição: definir a estrutura hierárquica do objetivo, estabelecendo os requisitos que dele derivam e suas associações</i>
<b>Agente</b>	engenheiro de requisitos
<b>Produto</b>	estrutura hierárquica do objetivo
<b>Recurso</b>	informação elicitada
<b>Anotação</b>	conforme definido no modelo de estruturação de requisitos

FIGURA 6.18 - Requisito "definir estrutura do objetivo"

<b>Ação</b>	<i>Nome:</i> <b>definir requisito</b>
	<i>Descrição:</i> definir um requisito do sistema (ação), especificando ação, agente, produto, recurso e anotação
<b>Agente</b>	engenheiro de requisitos
<b>Produto</b>	modelo de definição do requisito
<b>Recurso</b>	informação elicitada estrutura hierárquica do objetivo
<b>Anotação</b>	conforme definido no modelo de estruturação de requisitos

FIGURA 6.19 - Requisito "definir requisito"

<b>Ação</b>	<i>Nome:</i> <b>definir cenário</b>
	<i>Descrição:</i> definir um cenário de operações do sistema, associado a um ou vários requisitos da estrutura hierárquica
<b>Agente</b>	engenheiro de requisitos
<b>Produto</b>	cenário
<b>Recurso</b>	informação elicitada estrutura hierárquica do objetivo
<b>Anotação</b>	conforme definido no modelo de estruturação de requisitos

FIGURA 6.20 - Requisito "definir cenário"

<b>Ação</b>	<i>Nome:</i> <b>definir glossário de termos</b>
	<i>Descrição:</i> montar um glossário com os termos elicitados do universo de informação, definindo-os segundo conhecimento dos envolvidos
<b>Agente</b>	engenheiro de requisitos
<b>Produto</b>	glossário de termos
<b>Recurso</b>	informação elicitada modelos de definição de objetivos e requisitos
<b>Anotação</b>	associar cada termo aos objetos correspondentes do modelo de estruturação de requisitos

FIGURA 6.21 - Requisito "definir glossário de termos"

<b>Ação</b>	<i>Nome:</i> <b>definir casos de uso</b>
	<i>Descrição:</i> enumerar todos os possíveis casos de uso derivados dos requisitos do modelo de estruturação e definir quais participarão dos diagramas de casos de uso
<b>Agente</b>	engenheiro de requisitos
<b>Produto</b>	casos de uso
<b>Recurso</b>	modelo de estruturação de requisitos (modelos definição do requisito)
<b>Anotação</b>	denominar o caso de uso com o nome do requisito

FIGURA 6.22 - Requisito "definir casos de uso"

<b>Ação</b>	<b>Nome: definir relacionamento entre casos de uso</b>
	<i>Descrição:</i> relacionar os casos de uso, por dependências <i>extend</i> ou <i>include</i> , considerando a associação dos requisitos dos quais são derivados no modelo de estruturação de requisitos
<b>Agente</b>	engenheiro de requisitos
<b>Produto</b>	casos de uso relacionados
<b>Recurso</b>	modelo de estruturação de requisitos (estrutura hierárquica dos objetivos)
<b>Anotação</b>	

FIGURA 6.23 - Requisito "definir relacionamento entre casos de uso"

<b>Ação</b>	<b>Nome: definir atores associados aos casos de uso</b>
	<i>Descrição:</i> definir os atores e associá-los aos casos de uso, considerando os modelos de definição do requisito, nos quais estão identificados os agentes que participam da ação
<b>Agente</b>	engenheiro de requisitos
<b>Produto</b>	diagrama de casos de uso
<b>Recurso</b>	modelo de estruturação de requisitos (modelo de definição do requisito)
<b>Anotação</b>	

FIGURA 6.24 - Requisito "definir atores associados aos casos de uso"

Um macro cenário pode ser desenhado para facilitar o entendimento da seqüência de ações de modelagem, conforme abordagem proposta (figura 6.25). Em um primeiro momento definem-se os objetivos do sistema. Para cada objetivo é definida sua estrutura e especificados os requisitos e cenários. Após a estruturação do modelo de requisitos, podem ser gerados os diagrama de casos de uso.

O desenvolvimento da ferramenta de suporte à estruturação dos requisitos, conforme modelo proposto, é um atividade definida na proposta do Projeto FILM, o qual encontra-se em andamento, no Departamento de Informática da Universidade de Caxias do Sul.

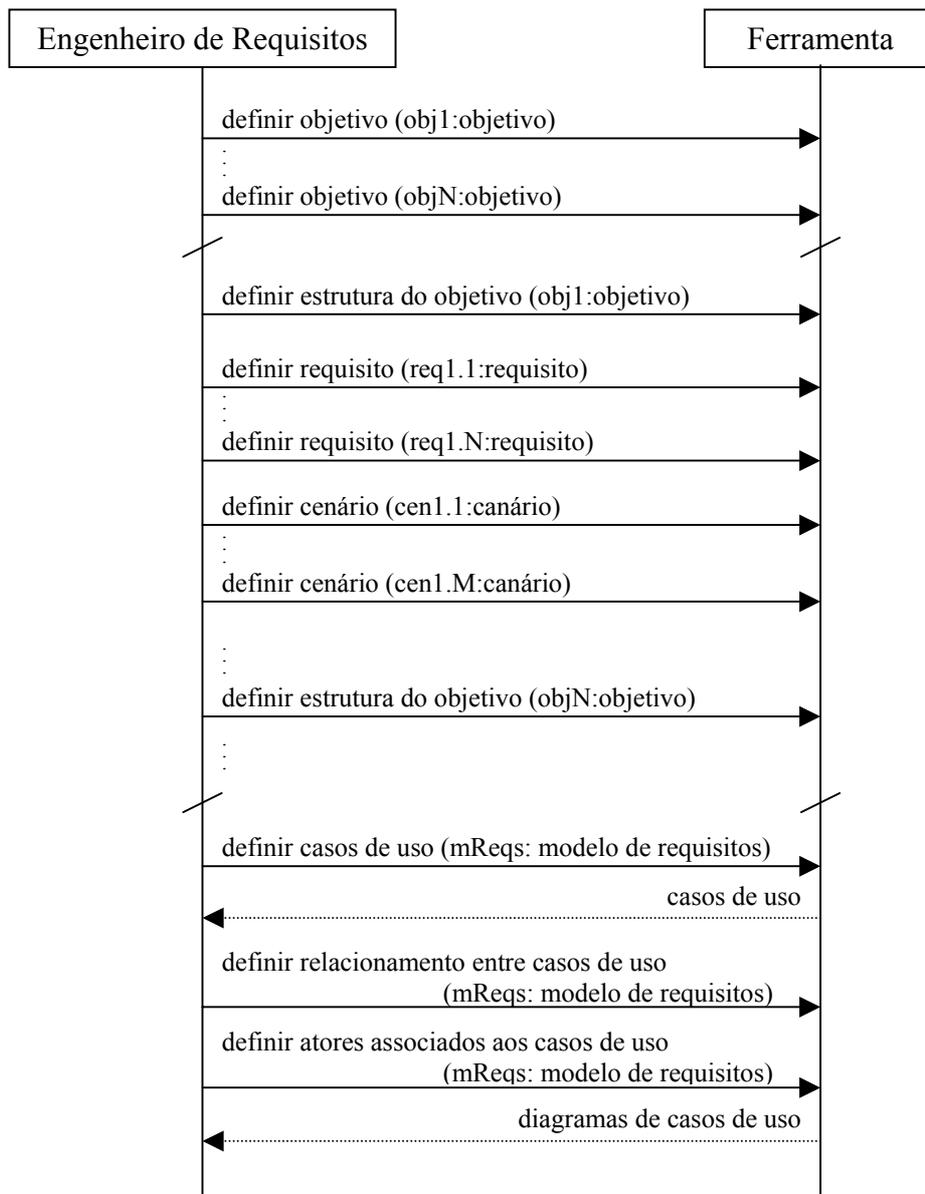


FIGURA 6.25 - Macro cenário de ações da ferramenta de suporte ao modelo proposto

## 7 Estudos de Casos

O modelo de requisitos proposto foi aplicado a três estudos de casos. Os estudos de casos foram escolhidos considerando a viabilidade de realização durante o período proposto para o desenvolvimento desta dissertação de mestrado, ou seja, o decorrer do ano de 2000. O porte do sistema de informação a ser desenvolvido e a técnica de elicitação empregada foram considerados visando adequar o estudo ao tempo proposto. A escolha das técnicas de elicitação dependeu diretamente do progresso no desenvolvimento deste trabalho, apresentando alguma peculiaridade em cada caso, desde a aplicação de entrevista livre, não estruturada, até a utilização de reuniões estruturadas de acordo com a abordagem do modelo.

Um processo de coleta bem sucedido inclui agregar maneiras eficientes para encorajar a efetiva participação e colaboração dos envolvidos. O conhecimento da estrutura e do comportamento do domínio do problema só será efetivo e satisfatório se houver o envolvimento de todos os *stakeholders*. Pesquisas realizadas revelam que a participação dos usuários no processo de desenvolvimento colaboram diretamente na satisfação do software [HER96].

Diversas abordagens para elicitação de requisitos têm sido apresentadas com o intuito de incentivar a participação e melhorar a comunicação entre os *stakeholders*. A seguir tem-se a apresentação de algumas técnicas de elicitação, estudadas com esse objetivo e possíveis de serem aplicadas (embora nem todas foram utilizadas), seguida de uma breve análise sobre sua aplicabilidade ao modelo proposto.

### 7.1 Métodos de Elicitação de Requisitos

A usabilidade do modelo, característica almejada pela proposta, diz respeito à facilidade com que os requisitos serão representados e organizados em uma abordagem voltada ao usuário. A comunicação e o entendimento entre os *stakeholders* é uma premissa importante para o atendimento a esse objetivo, assim como a utilização de um método de elicitação que conte com a efetiva participação dos envolvidos na definição dos requisitos.

A elicitação envolve todo um conjunto de ações, que ocorrem no universo de informações, visando capturar e registrar as informações que subsidiarão o entendimento do domínio do problema e a conseqüente especificação dos requisitos. A captura é um processo de descoberta no qual procura-se obter o máximo de informações para o conhecimento do objeto em questão. Essa atividade requer uma habilidade em trabalhar com especialistas humanos e com o conhecimento tácito, que é trivial para quem conhece a informação mas não é trivial para quem procura obtê-la, de forma que o conhecimento tácito dificilmente é lembrado e, portanto, não é transmitido [GOG94].

Não é uma tarefa fácil obter esse conhecimento, pois os envolvidos têm experiências, conhecimentos, preconceitos e terminologias diferentes. Algumas técnicas das ciências sociais, como psicologia e sociologia, têm sido estudadas e utilizadas nesta atividade, as quais envolvem fatores comportamentais e de relacionamento humano [BEJ98] [DEB99].

Entre as técnicas mais comuns para a aquisição do conhecimento estão: análise de documentos, entrevistas, reuniões e observações. Essas técnicas foram aplicadas no estudos de casos definidos, considerando-se a facilidade de sua aplicação em função do tempo disponível para o estudo e do porte dos casos versados. Outras

técnicas, como etnografia e *inquiry cycle model*, também são apresentadas como adequadas ao modelo proposto, contudo não foram aplicadas nos estudos de casos por serem técnicas convenientes a sistemas mais críticos e requererem maior tempo para suas aplicações.

A análise de documentos é uma técnica usualmente aplicada na qual explora-se o conhecimento escrito encontrado no universo de informações. Na modelagem dos requisitos, segundo o modelo proposto, essa técnica é muito útil para a definição dos objetos que compõem o modelo. A análise dos documentos permite um contato com o vocabulário utilizado no domínio do problema e auxilia a construção do glossário de termos especializados, que tem por objetivo definir os objetos e equalizar o conhecimento dos *stakeholders*.

### 7.1.1 Técnicas de Entrevistas

Entrevista é uma técnica de interação entre entrevistado (especialista do conhecimento) e entrevistador (engenheiro de requisitos) buscando revelar conceitos, objetos e a organização do domínio do problema, além de buscar soluções ou projeções de soluções que comporão o domínio da solução.

As entrevistas mais usuais são as tutoriais, informais e estruturadas [OLI96]. Nas entrevistas tutoriais o entrevistado fica no comando, praticamente lecionando sobre um determinado assunto. Nas entrevistas informais ou não estruturadas, o entrevistador age espontaneamente, perguntando ao entrevistado sem obedecer a nenhuma organização. Esse tipo de entrevista oferece flexibilidade ao entrevistador e, normalmente, é utilizado no início do processo de elicitação. Já as entrevistas estruturadas são preparadas pelo entrevistador, que define previamente o andamento do procedimento de aquisição de conhecimento. Um fator importante a ser considerado nas entrevistas é o registro das informações coletadas, que pode ser realizado através de anotações ou gravações de áudio ou vídeo. O material produzido deve ser organizado e serve como base para a preparação da próxima entrevista.

A entrevista é uma técnica muito adequada para a modelagem dos requisitos segundo o modelo proposto. Essa técnica permite o contato direto entre os envolvidos e possibilita a avaliação imediata das informações elicítadas. O modelo dá suporte a essa prática de elicitação, permitindo a organização das informações no momento da entrevista.

Em um momento inicial é sugerido que se realize entrevistas informais a fim de que informações gerais sobre o domínio do problema seja trocadas entre os envolvidos. Essas informações preliminares permitem a detecção das atividades realizadas no sistema. As atividades, que definem os objetivos, são as diretrizes básicas para a organização dos requisitos. A partir da identificação dos objetivos, as entrevistas podem ser estruturadas conforme os próprios objetivos. Para cada objetivo definido, as entrevistas são preparadas a fim de conduzir à elicitação e estruturação dos requisitos, até o desenho dos cenários, os quais permitem uma representação mais concreta das operações previstas para o software, em um nível de abstração adequado a validação entre os envolvidos no processo.

### 7.1.2 Técnicas de Reuniões

Reunião é uma técnica que prevê a participação coletiva dos envolvidos para discutir questões do domínio do problema. Esta prática permite uma interação mais natural entre os participantes e dispor de múltiplas visões sobre a questão abordada.

*Participatory Design* e *Joint Application Design* são metodologias de reuniões que enfatizam a participação coletiva na elicitação de requisitos. São práticas bem conhecidas de negociação de requisitos, que promovem a cooperação, entendimento e formação de equipes de trabalho entre os envolvidos no universo de informações.

Essas abordagens oferecem um ambiente apropriado para os desenvolvedores e usuários trabalharem em equipe, com o objetivo de compartilhar informações e idéias sobre os domínios do problema e da solução. Este processo auxilia a comunicação entre os envolvidos, que empenham-se em identificar as necessidades, refinar os requisitos, tomar decisões conjuntas, explorar possíveis soluções e selecionar alternativas apropriadas.

A técnica *Joint Application Design* (JAD) baseia-se em sessões estruturadas e disciplinadas, onde os envolvidos reúnem-se para desenvolver juntos o sistema de software. Em linhas gerais, essas sessões possuem uma agenda detalhada, recursos visuais para auxiliar a exposição de idéias, um moderador e um relator que registra as especificações de acordo comum entre os membros do grupo. O produto final é um documento que contém definições do software [DEB99].

*Participatory Design* (PD) é uma abordagem que concentra-se mais fortemente no envolvimento dos usuários, em relação ao *Joint Application Design*, por facilitar o processo de aprendizado entre desenvolvedores e usuários através de experiências conjuntas em situações de trabalho simuladas. Em linhas gerais, os usuários são introduzidos no ambiente dos desenvolvedores, conhecendo possibilidades técnicas e, da mesma maneira, os desenvolvedores colaboram com os usuários em suas tarefas. Ocorre um aprendizado mútuo que vem a contribuir no processo de definição dos requisitos [HER96].

*JAD* representa um movimento em direção a práticas mais colaborativas procurando viabilizar objetivos, enquanto que *PD* representa um movimento em direção a práticas mais técnicas procurando, também, viabilizar objetivos [CAR93]. Ambas metodologias são bem conhecidas por acentuar um alto grau de envolvimento dos usuários como imperioso para o desenvolvimento de um bom projeto de software. Como resultado, os desenvolvedores aumentam seus conhecimentos sobre o domínio do problema e os usuários tornam-se mais envolvidos no processo de desenvolvimento.

Todas essas técnicas de reuniões também são muito adequadas para a modelagem dos requisitos segundo o modelo proposto. Além de permitir o contato e o envolvimento direto entre desenvolvedores e usuários, possibilitam a avaliação imediata das informações elicitadas e a resolução de conflitos e pontos de vistas diversos sobre as necessidades e requisitos. É um trabalho conjunto e participativo que visa uma definição consensual sobre a solução mais apropriada.

O modelo dá suporte a essa prática de elicitação, considerando que as reuniões são organizadas com uma pauta de assuntos bem definidos. Sugere-se que a pauta seja estabelecida por atividades, visando buscar a conformidade com a abordagem proposta pelo modelo, o qual estrutura os requisitos segundo os princípios estabelecidos na Teoria da Atividade. Tanto as reuniões *JAD* quanto *PD* são determinadas por objetivos que planificam o andamento dos trabalhos, perfeitamente adequadas a estruturação dos requisitos por objetivos.

### 7.1.3 Técnicas de Observações

Observação é uma técnica na qual o engenheiro de requisitos procura ter uma posição passiva no domínio do problema, observando seus elementos e

comportamentos. Esta estratégia visa obter um entendimento inicial sobre o contexto em estudo.

As observações consistem em observar alguém no momento da realização de suas tarefas rotineiras no ambiente real. O observador procura familiarizar-se com o domínio do problema e elicitare as informações necessárias para o seu entendimento. A aquisição desse conhecimento pode ocorrer com interrupção ou não das atividades do observado. Na observação simples, o observador acompanha o raciocínio sem interromper o processo, enquanto que na análise por interrupção, o observador pode suspender o processo a fim de esclarecer o raciocínio das atividades e operações realizadas [OLI96].

A técnica de observação é útil na elaboração do modelo por apresentar a vantagem de ser simples e de baixo custo para um conhecimento essencial do comportamento dos elementos no domínio do problema. Colabora na percepção inicial e definição das atividades, assim como no esclarecimento sobre as ações executadas.

A utilização dessa técnica permite o reconhecimento das atividades realizadas sob o domínio do problema, base para o desenvolvimento da abordagem e essencial para o estabelecimento da estruturação dos requisitos segundo a proposta do modelo. Também, a observação possibilita a identificação dos objetos que compõem a descrição dos objetivos, requisitos e cenários, bem como colabora na definição do glossário de termos. O modelo suporta, portanto, a elicitação de informações por observações, como uma técnica auxiliar na definição dos requisitos.

#### 7.1.4 Abordagem Etnográfica

Outra técnica que pode ser aplicada de forma complementar com o intento de permitir uma visão mais completa e ajustada do domínio do problema é a etnografia. Etnografia é a coleta direta, e o mais minuciosa possível, dos fenômenos observados, por uma impregnação duradoura e contínua e um processo que se realiza por aproximações sucessivas [LAP88]. É originária da antropologia, a partir de observações de práticas das sociedades. No contexto da Engenharia de Requisitos, é uma técnica que busca revelar informações sobre a estrutura, organização e práticas do domínio do problema [SOM93].

Em um estudo de caso do projeto *Editorial Board* [SIM97], foram aplicadas técnicas tradicionais, como reuniões, entrevistas e análises de documentos, e técnicas etnográficas, além de observações e análises de gravações de vídeo, durante o projeto conceitual. Daquela aplicação, pode-se concluir que as observações dos usuários envolvidos em suas atividades corriqueiras são necessárias para o estabelecimento de uma compreensão mútua do processo, no auxílio do entendimento da prática corrente e no desenvolvimento de uma visão geral dos objetivos do projeto. A utilização da etnografia pode, também, revelar múltiplos pontos de vista dos usuários sobre as atividades correntes e sobre os objetivos do projeto. Esses diferentes pontos de vista podem ser harmônicos ou problemáticos no sentido das possibilidades de integração em um sistema coerente, sendo que os conflitos devem ser analisados e apresentadas diferentes soluções e suas conseqüências, visando a busca de um consenso. A etnografia, segundo a conclusão do projeto realizado, pode levar à obtenção de requisitos fechados à prática corrente, o conhecimento tácito.

A abordagem etnográfica aplicada em conjunto com outras técnicas de elicitação possibilita um entendimento completo das atividades realizadas e a minimização do conhecimento tácito, dada a grande integração ao ambiente. Trabalhos realizados utilizando-se essa abordagem têm comprovado a eficácia na obtenção dos

requisitos pela participação do engenheiro de requisitos no ambiente de trabalho [DEB99].

### 7.1.5 Inquiry Cycle Model

O *Inquiry Cycle Model* é uma estrutura formal utilizada para descrever discussões sobre requisitos. Consiste em três atividades (figura 7.1) que se repetem sob uma estratégia de controle, nas quais os envolvidos propõem e discutem seus requisitos e refinam-nos quando aprovados [POT94].

Na fase de *expressão* são apresentadas informações relacionadas ao domínio do problema, com o objetivo de analisá-las cooperativamente. Na fase de *discussão* são debatidos os comentários e anotações individuais sobre a análise da expressão. Na fase de *compromisso* são tomadas as decisões resultantes das discussões, acordadas as terminologias e comprometida a procura pelas informações ausentes. Cada ciclo completo pelas três fases finaliza um estágio de investigação. O progresso no cumprimento dos estágios segue uma estratégia que visa a aquisição completa e o refinamento dos requisitos capturados.

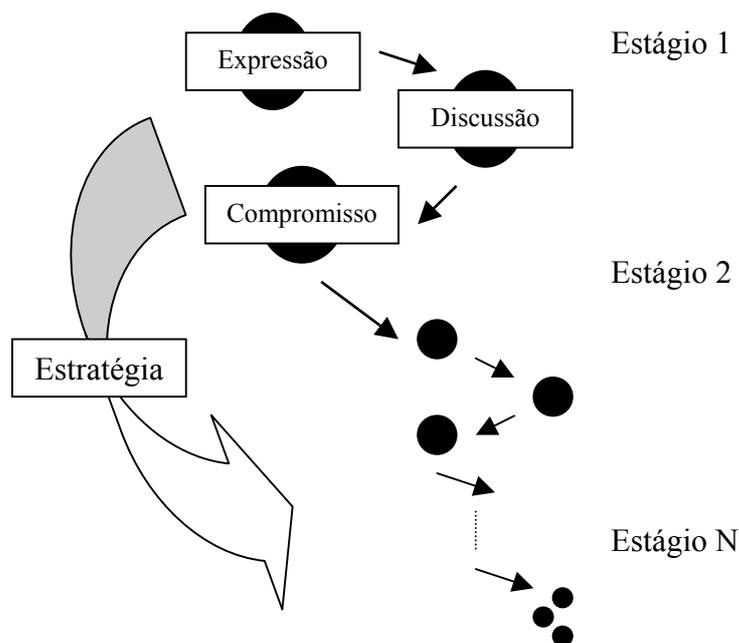


FIGURA 7.1 - Fases do *Inquiry Cycle Model*

O *Inquiry Cycle Model* pode ser instanciado por diversas técnicas e métodos, dependendo da aplicação. É uma técnica adequada a ser aplicada visando a estruturação dos requisitos segundo o modelo proposto. O fato do modelo propor uma hierarquia de requisitos que parte de um objetivo até a especificação de operações de software é complacente à técnica, que estabelece uma estratégia de refinamento dos requisitos elicitados.

## 7.2 Os Estudos de Casos

Os três casos selecionados para este estudo foram desenvolvidos utilizando-se, basicamente, as técnicas de análise de documentos, entrevistas, reuniões e observações para a elicitação dos requisitos. Em cada caso a abordagem de aplicação das técnicas de maneira adequada a proposta foi progressiva, no sentido que partiu-se de um caso utilizado para corroborar o modelo pré definido, a um caso em que as técnicas foram aplicadas de forma sintonizada a estruturação dos requisitos segundo o modelo.

### 7.2.1 Caso: Controle de Serviços Operacionais

No caso do *Controle de Serviços Operacionais* (utilizado parcialmente como ilustração na apresentação do modelo), foram aplicadas as técnicas de análise de documentos e entrevistas livres, não estruturadas.

Em um primeiro contato com a gerência do setor de controle operacional advieram algumas necessidades de informatização das atividades do setor. Uma entrevista inicial foi realizada para elaborar uma narrativa sobre as necessidades preliminares de desenvolvimento do software. Foi uma entrevista livre, não estruturada, na qual a gerência relatou algumas atividades do escopo do sistema requerido, e da qual resultou um documento com a narrativa básica das necessidades.

Outras reuniões, também não estruturadas, e coleta de documentos ocorreram contando com a participação da gerência do setor de controle operacional. Nessas entrevistas e análise dos documentos coletados, buscou-se definir alguns termos, bem como, elaborar mais detalhadamente a descrição das necessidades.

Com o documento gerado, procurou-se identificar as atividades desempenhadas pelo setor como o intuito de estruturar os objetivos, conforme o modelo proposto. Observou-se que, embora as entrevistas foram conduzidas livremente, a narrativa apresentou uma certa organização natural por atividades. Essa estruturação espontânea pelos próprios entrevistados (gerência operacional) colaborou com o processo de entendimento das necessidades. E contribuiu favoravelmente à argumentação da abordagem proposta.

Este caso, ainda que com um escopo de necessidades diminuto, serviu como base de estudo para a elaboração e avaliação do modelo proposto. A partir das atividades identificadas, estruturou-se os objetivos e derivou-se os requisitos. A gerência operacional participou, também, da validação dos requisitos, na qual foram utilizados cenários que representavam as interações entre os agentes e o software. Do modelo de requisitos gerou-se os modelos de análise do Método Fusion, sem ainda ter-se a definição do processo de sistematização que orientasse tal atividade. Este caso constituiu, portanto, a base de estudo para definição do modelo proposto.

### 7.2.2 Caso: Sistema de Protocolo

O segundo caso, *Sistema de Protocolo*, foi desenvolvido em uma empresa municipal de saneamento. O protocolo é um serviço de atendimento à população, ou aos próprios servidores municipais, para solicitações de serviços ou informações. Exemplos desses serviços são ligação de água e manutenção de cadastro de usuários, projetos de redes de abastecimento, licitações e cadastramento de fornecedores, além de processo com assuntos administrativos. O protocolo realiza o registro das solicitações, em forma de processos administrativos que tramitam pelos

setores da empresa, onde cada setor é responsável por ações pertinentes, até a execução completa do serviço ou o indeferimento da solicitação.

Para a atividade de elicitação das informações foram utilizadas as técnicas de análise de documentos e do software existente, observações, entrevistas estruturadas e reuniões também estruturadas. A abordagem utilizada para a fase de aquisição do conhecimento foi orientada às atividades realizadas no contexto do domínio do problema.

Primeiramente, a análise do software existente e de documentos foram realizadas com o objetivo de definir-se as atividades principais do sistema. Tendo essas atividades definidas, pode-se estruturar as entrevistas e reuniões por objetivos, com o intuito de discutir e esclarecer o conhecimento relacionado à atividade específica. Nas entrevistas e reuniões, as informações foram organizadas conforme a estrutura definida por atividades e ações. Também os diferentes pontos de vista foram dirimidos nas reuniões e elicitadas as informações esclarecidas e validadas. Observações, simples e com interrupções, também foram aplicadas com o propósito de elucidar as ações realizadas pelos usuários. Essa técnica foi utilizada no acompanhamento das atividades que envolvem os processos administrativos, inclusive nas atividades informatizadas do processo, nas quais é utilizado o software existente.

Da análise preliminar do contexto da aplicação, pode-se perceber duas atividades principais que conduzem a dois objetivos: o "cadastramento dos processos administrativos" e o "controle do andamento dos processos administrativos". Esses dois objetivos modelam o topo da estrutura de requisitos, segundo o modelo proposto. Representam o mais alto grau de abstração das necessidades, no qual tem-se retratados requisitos do negócio. Cada objetivo é representado por um objeto, conforme o modelo de requisitos.

A abordagem utilizada conduziu os envolvidos a determinar as ações necessárias para o atendimento aos objetivos. Esses objetivos orientaram o processo de especificação dos requisitos, estruturando as entrevistas e reuniões. Para cada objetivo tratado, buscou-se elicitar e estruturar os requisitos que correspondiam às ações necessárias para o atendimento ao objetivo em questão, e retratavam os requisitos de usuário e funcionais.

Do objetivo de "cadastramento dos processos administrativos", obtiveram-se os requisitos: "cadastrar assunto", "cadastrar requerente" e "abrir processo administrativo". Do objetivo de "controle do andamento dos processos administrativos", obtiveram-se os requisitos: "cadastrar setor" e "cadastrar autorizado", "tramitar processo", "finalizar processo", "acompanhar processo". O requisito "tramitar processo" é detalhado em: "receber processo" e "encaminhar processo". O requisito "finalizar processo" é detalhado em: "encerrar processo" e "arquivar processo", ou "reabrir processo". O requisito "acompanhar processo" é detalhado em: "consultar processo" ou "quantificar processos".

A estrutura dos objetivos e requisitos especificada para o caso do *Sistema de Protocolo* é a apresentada abaixo (figura 7.2). Uma breve descrição do contexto da aplicação, a definição dos objetivos, requisitos e cenários e o glossário de termos são apresentados em anexo.

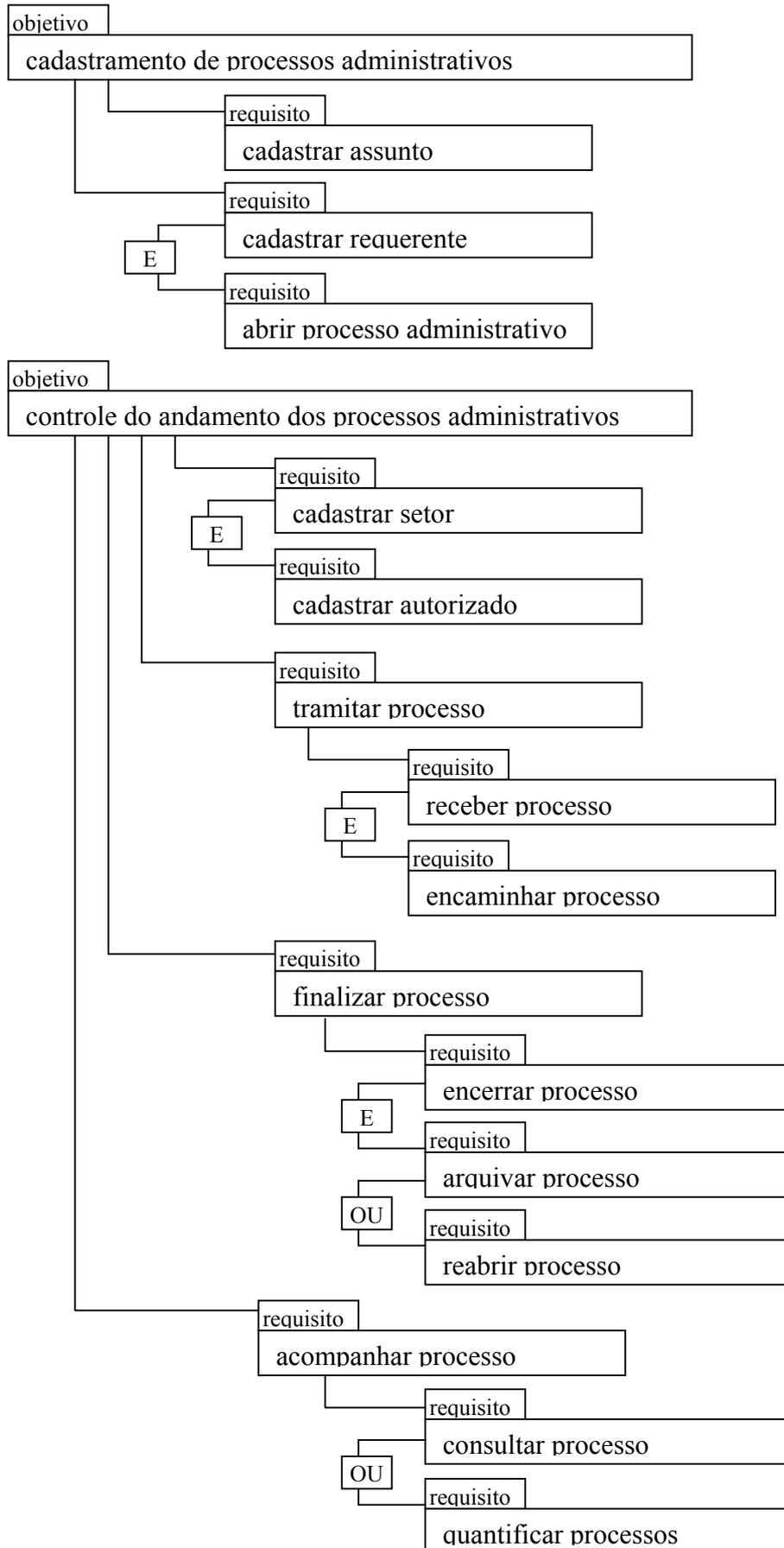


FIGURA 7.2 - Estrutura de objetivos e requisitos do sistema de protocolo

### 7.2.3 Caso: Controle de Imóveis

O terceiro caso consiste em uma aplicação de *Controle de Imóveis*. Um grupo de empresas que atuam no ramo de loteamentos é proprietário de um conjunto de imóveis localizados, principalmente, nos municípios de Caxias do Sul e Farroupilha. Os imóveis compreende lotes, distribuídos em alguns loteamentos, e edificações, como casas, apartamentos, salas e vagas em estacionamentos. O propósito do grupo empresarial é a gerência dos imóveis para comercialização (venda). Dentre esses imóveis, alguns estão sob judicío, outros, locados ou disponíveis para venda. Os imóveis sob judicío são objetos de processos judiciais, cujo andamento deve ser acompanhado. Em razão da dificuldade de manter o controle sobre a situação dos imóveis, surgiu a necessidade de organizar uma administração centralizada. O objetivo principal é congregiar as informações de todos os imóveis do grupo empresarial.

Neste caso foram utilizadas técnicas de entrevistas informais e entrevistas estruturadas para a realização da elicitação e modelagem dos requisitos, além da análise de documentos. A atividade de especificação dos requisitos partiu de entrevistas informais buscando situar o domínio do problema. Destas entrevistas estabeleceu-se os objetivos do sistema, para, então, estruturar as entrevistas seguintes.

Os objetivos, demarcados pelas atividades realizadas no sistema real, serviram de diretrizes para a estruturação das entrevistas. Em cada entrevista, a atividade delimitou o escopo da abordagem, com o propósito de tratar os requisitos relativos ao atendimento àquele objetivo. Os documentos analisados também foram organizados, elicitando-se suas informações em conformidade à atividade na qual eram utilizados.

O diferencial da abordagem do processo de elicitação neste caso do controle de imóveis para o caso do sistema de protocolo reside na modelagem dos objetivos e requisitos. Enquanto no sistema de protocolo os requisitos foram modelados após as entrevistas estruturadas por atividades, no caso da administração de imóveis a abordagem e estruturação das entrevistas foi conduzida à representação direta dos requisitos conforme o modelo. Em outras palavras, os questionamentos levantadas durante as entrevistas tiveram o propósito de especificar diretamente os objetos que compunham os objetivos e requisitos.

Para cada objetivo, foram definidos o resultado, objetos, sujeitos e ferramentas pelos questionamentos sugeridos (Qual o resultado esperado no atendimento ao objetivo? Por que há essa necessidade? Para o que esse objetivo faz-se necessário ao sistema? Quem atua em prol do atendimento a esse objetivo? Quais são os recursos necessários para os sujeitos interagirem com os objetos?). Na seqüência da definição de cada objetivo, foram definidos os requisitos derivados, também seguindo a mesma abordagem. Para cada ação elicitada, procurou-se estruturá-la segundo o modelo proposto e representá-la conforme seus objetos agente, produto, recurso e anotação. Complementando a elicitação e modelagem, cenários foram desenhados para a validação das operações especificadas, em um esforço conjunto entre engenheiro de requisitos e clientes e usuários.

Foram definidos três objetivos para o software de controle de imóveis: "cadastramento dos imóveis" (figura 7.3), "manutenção da situação dos imóveis" (figura 7.4) e "controle da situação dos imóveis" (figura 7.5).

<b>Objetivo</b>	<i>Uma necessidade essencial é manter todas as informações sobre os imóveis das empresas em um cadastros centralizado, com o objetivo de facilitar a administração do patrimônio.</i>
<b>Resultado</b>	<i>Qual o resultado esperado? Por que há a necessidade?</i> cadastramento dos imóveis
<b>Objeto</b>	<i>Para o que esse objetivo faz-se necessário ao sistema?</i> cadastro de imóveis
<b>Sujeito</b>	<i>Quem atua em prol desse objetivo?</i> secretária
<b>Ferramenta</b>	<i>Quais são os recursos necessários para os sujeitos interagirem com o objeto?</i> mapas dos loteamentos escrituras dos imóveis contratos de locação de imóveis carnês de IPTU (imposto predial e territorial urbano)

FIGURA 7.3 - Definição do objetivo de cadastramento dos imóveis

<b>Objetivo</b>	<i>Há a necessidade de manter atualizada a situação dos imóveis, considerando três situações: livre para comercialização ou usufruto, sob judicie ou vendido. É necessário, também, manter atualizado o andamento do processo judicial dos imóveis sob judicie, a fim de que possa-se ter um acompanhamento apurado das suas situações legais.</i>
<b>Resultado</b>	<i>Qual o resultado esperado? Por que há a necessidade?</i> manutenção da situação dos imóveis
<b>Objeto</b>	<i>Para o que esse objetivo faz-se necessário ao sistema?</i> situação dos imóveis
<b>Sujeito</b>	<i>Quem atua em prol desse objetivo?</i> secretária
<b>Ferramenta</b>	<i>Quais são os recursos necessários para os sujeitos interagirem com o objeto?</i> contratos de vendas de imóveis extratos de andamento de processos judiciais (obtidos pela internet ou diretamente nos Fórum de Justiça)

FIGURA 7.4 - Definição do objetivo de manutenção da situação dos imóveis

<b>Objetivo</b>	<i>O propósito em ter-se o cadastro de imóveis centralizado com as suas situações atualizadas é poder consultar prontamente todas as informações dos imóveis, com o objetivo de facilitar a administração do patrimônio das empresas e tomar decisões mais apuradas.</i>
<b>Resultado</b>	<i>Qual o resultado esperado? Por que há a necessidade?</i> controle da situação dos imóveis
<b>Objeto</b>	<i>Para o que esse objetivo faz-se necessário ao sistema?</i> administração dos imóveis
<b>Sujeito</b>	<i>Quem atua em prol desse objetivo?</i> administrador
<b>Ferramenta</b>	<i>Quais são os recursos necessários para os sujeitos interagirem com o objeto?</i> cadastro de imóveis situação dos imóveis

FIGURA 7.5 - Definição do objetivo de controle da situação dos imóveis

Do objetivo de "cadastramento dos imóveis", obtiveram-se os requisitos: "cadastrar proprietário" (figura 7.7), "cadastrar loteamento" (figura 7.8), "registrar imóvel" (figura 7.9) e "cadastrar lote" (figura 7.10) ou "cadastrar edificação" (figura 7.11), ou "informar contrato de locação" (figura 7.12), que compreende "cadastrar imobiliária" (figura 7.13) e "cadastrar inquilino" (figura 7.14), conforme estrutura representada pelo esquema a seguir (figura 7.6).

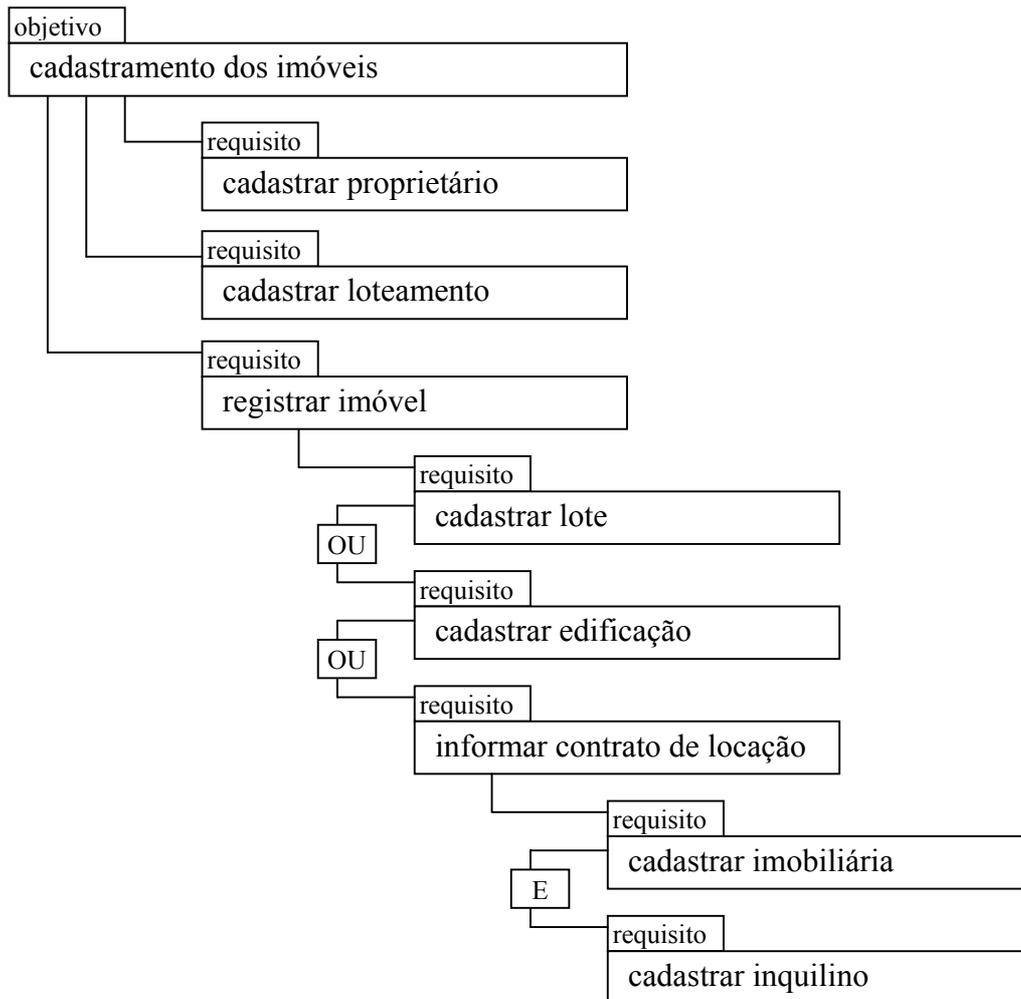


FIGURA 7.6 - Estrutura dos requisitos do objetivo de cadastramento de imóveis

<b>Requisito</b>	<i>Nome: cadastrar proprietário</i>
	<i>Descrição: cadastrar a empresa proprietária de imóveis</i>
<b>Agente</b>	secretária
<b>Produto</b>	cadastro de proprietários
<b>Recurso</b>	CNPJ e razão social (das escrituras de imóveis)
<b>Anotação</b>	

FIGURA 7.7 - Requisito “cadastrar proprietário”

<b>Requisito</b>	<i>Nome: cadastrar loteamento</i>
	<i>Descrição: cadastrar o loteamento onde localizam-se os imóveis</i>
<b>Agente</b>	secretária
<b>Produto</b>	cadastro de loteamentos
<b>Recurso</b>	nome e localização (dos mapas dos loteamentos)
<b>Anotação</b>	

FIGURA 7.8 - Requisito “cadastrar loteamento”

<b>Requisito</b>	<i>Nome: registrar imóvel</i>
	<i>Descrição: cadastrar o imóvel, com os dados do registro de imóveis</i>
<b>Agente</b>	secretária
<b>Produto</b>	cadastro de imóveis
<b>Recurso</b>	inscrição na prefeitura (dos carnês de IPTU) matrícula, zona, livro e folha (das escrituras de imóveis) medidas e área total do imóvel (das escrituras de imóveis)
<b>Anotação</b>	

FIGURA 7.9 - Requisito “registrar imóvel”

<b>Requisito</b>	<i>Nome: cadastrar lote</i>
	<i>Descrição: se imóvel for lote, cadastrar dados do lote</i>
<b>Agente</b>	secretária
<b>Produto</b>	cadastro de imóveis
<b>Recurso</b>	cadastro de imóveis cadastro de loteamentos quadra e número do lote (dos mapas de loteamentos e escrituras de imóveis)
<b>Anotação</b>	

FIGURA 7.10 - Requisito “cadastrar lote”

<b>Requisito</b>	<i>Nome: cadastrar edificação</i>
	<i>Descrição: se imóvel possuir edificação, cadastrar dados da edificação</i>
<b>Agente</b>	secretária
<b>Produto</b>	cadastro de imóveis
<b>Recurso</b>	cadastro de imóveis endereço e tipo de edificação (das escrituras de imóveis e carnês de IPTU)
<b>Anotação</b>	domínio dos tipos de edificação: casa, apartamento, sala ou box

FIGURA 7.11 - Requisito “cadastrar edificação”

<b>Requisito</b>	<i>Nome: informar contrato de locação</i>
	<i>Descrição: se imóvel locado, informar dados do contrato de locação</i>
<b>Agente</b>	secretária
<b>Produto</b>	cadastro de imóveis
<b>Recurso</b>	cadastro de imóveis cadastro de imobiliárias cadastro de inquilinos data do contrato, valor e vencimento (dos contratos de locação)
<b>Anotação</b>	

FIGURA 7.12 - Requisito “informar contrato de locação”

<b>Requisito</b>	<i>Nome: cadastrar imobiliária</i>
	<i>Descrição: cadastrar a imobiliária responsável por locações de imóveis</i>
<b>Agente</b>	secretária
<b>Produto</b>	cadastro de imobiliárias
<b>Recurso</b>	nome, contato e telefone da imobiliária
<b>Anotação</b>	

FIGURA 7.13 - Requisito “cadastrar imobiliária”

<b>Requisito</b>	<i>Nome: cadastrar inquilino</i>
	<i>Descrição: cadastrar o inquilino, locador de imóveis</i>
<b>Agente</b>	secretária
<b>Produto</b>	cadastro de inquilinos
<b>Recurso</b>	nome, identidade e CPF do inquilino (dos contratos de locação)
<b>Anotação</b>	

FIGURA 7.14 - Requisito “cadastrar inquilino”

Foram desenhados alguns cenários com o propósito conversacional entre os envolvidos de melhorar o entendimento sobre os requisitos associados ao objetivo de "cadastramento dos imóveis": cadastrar lote livre (figura 7.15), cadastrar edificação livre (figura 7.16), cadastrar lote com edificação (figura 7.17) e cadastrar edificação locada (figura 7.18).

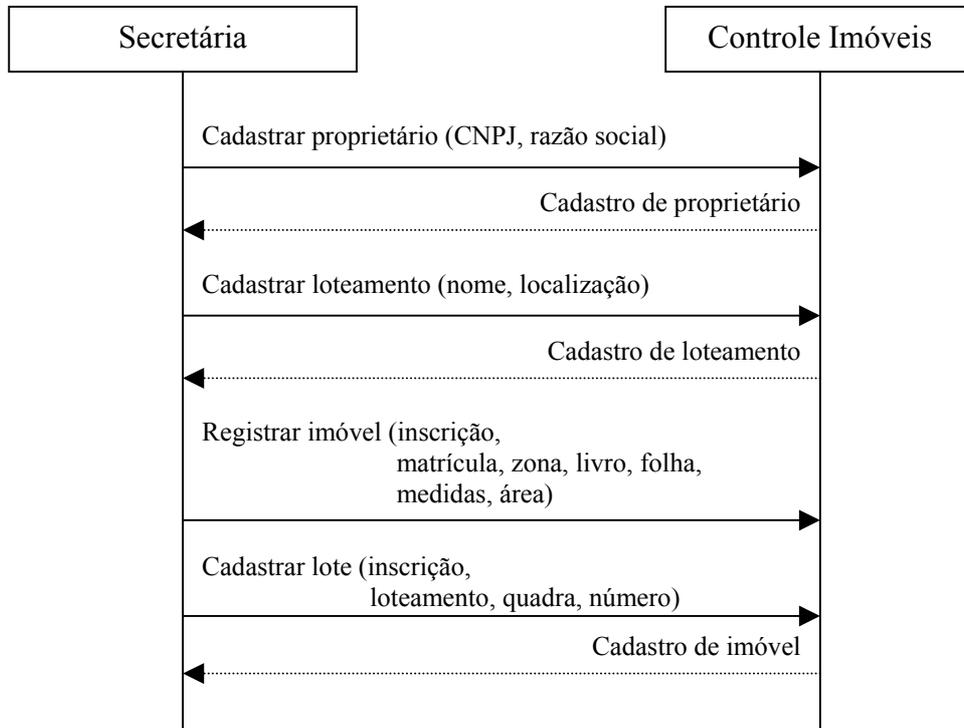


FIGURA 7.15 - Cenário de cadastrar lote livre

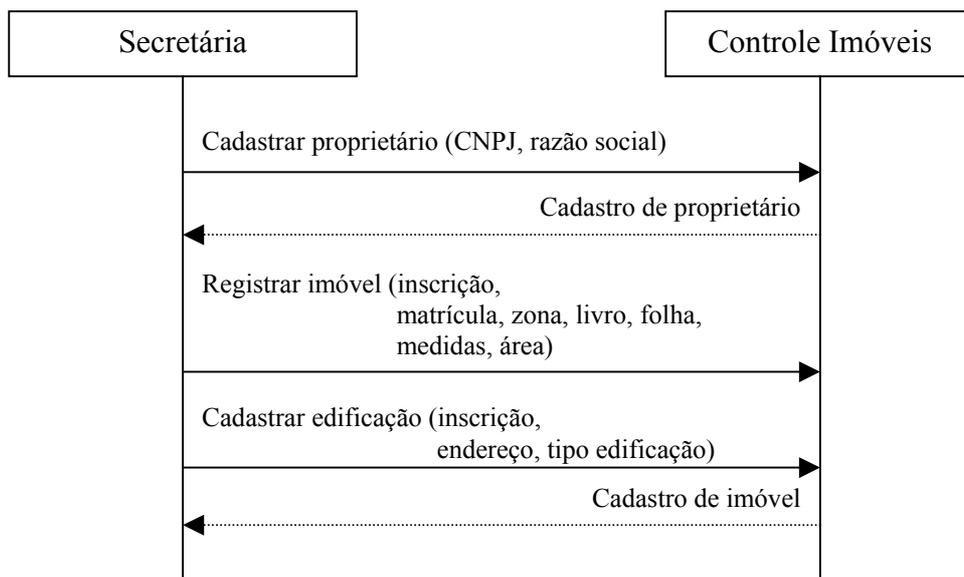


FIGURA 7.16 - Cenário de cadastrar edificação livre

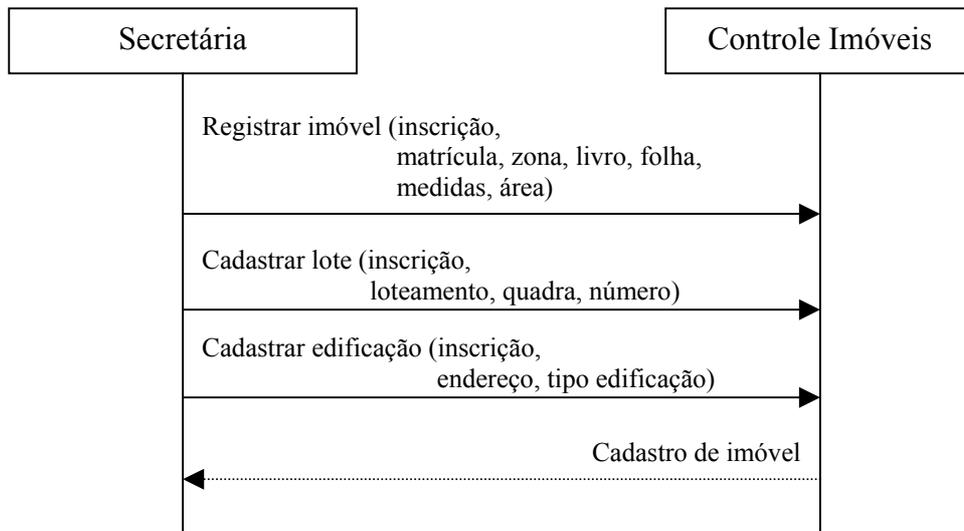


FIGURA 7.17 - Cenário de cadastrar lote com edificação

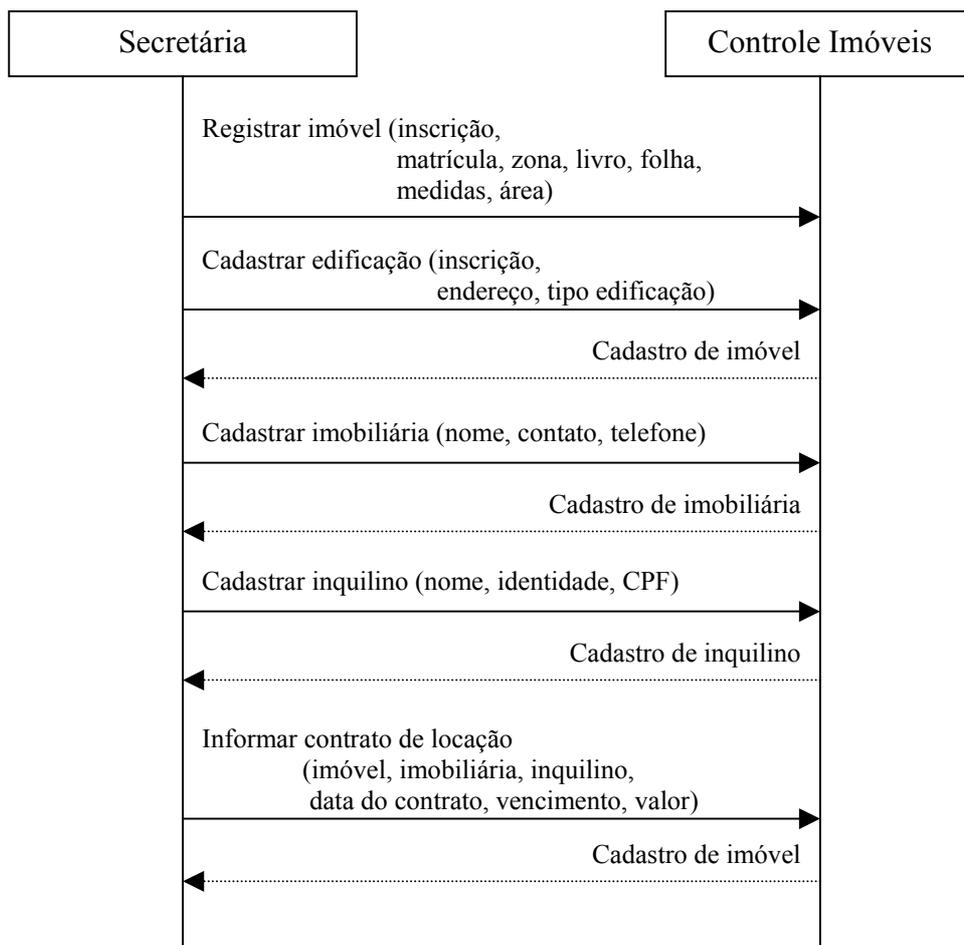


FIGURA 7.18 - Cenário de cadastrar edificação locada

Do objetivo de "manutenção da situação dos imóveis", obtiveram-se os requisitos: "atualizar situação judicial do imóvel" (figura 7.20), que compreende "informar dados do processo judicial" (figura 7.21) ou "informar andamento do processo" (figura 7.22), "vender imóvel" (figura 7.23), conforme estrutura representada pelo esquema a seguir (figura 7.19).

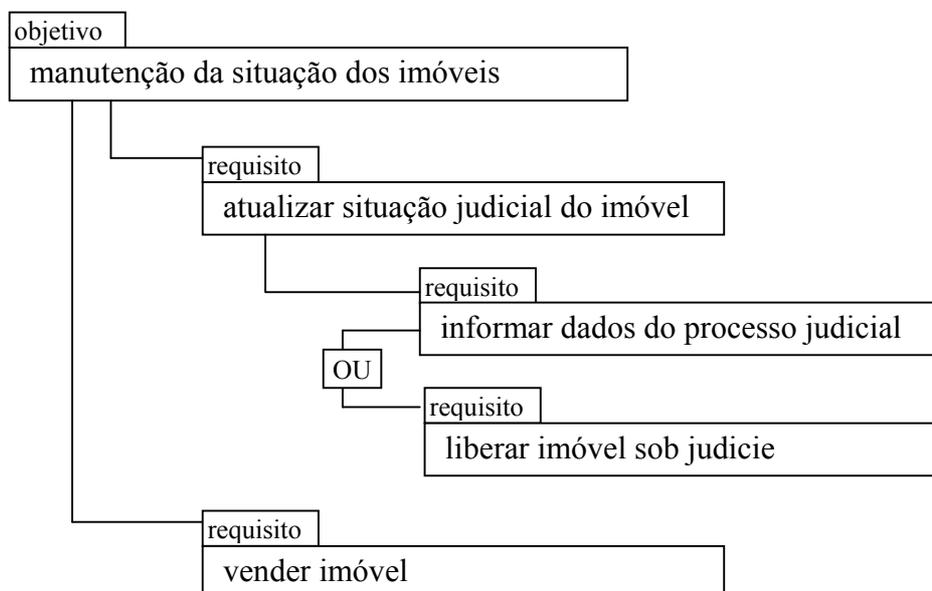


FIGURA 7.19 - Estrutura dos requisitos do objetivo de manutenção da situação dos imóveis

<b>Requisito</b>	<b>Nome: atualizar situação judicial do imóvel</b>
	<i>Descrição:</i> atualizar a situação judicial do imóvel, se livre ou sob judicie, informando o processo judicial e seu andamento
<b>Agente</b>	secretária
<b>Produto</b>	andamento dos processos judiciais dos imóveis
<b>Recurso</b>	cadastro de imóveis informações do processo judicial (número, vara e comarca) informações sobre o andamento do processo (data e extrato do andamento)
<b>Anotação</b>	

FIGURA 7.20 - Requisito "atualizar situação judicial do imóvel"

<b>Requisito</b>	<b>Nome: informar dados do processo judicial</b>
	<i>Descrição:</i> cadastrar os dados do processo judicial associado ao imóvel, no caso de imóvel sob judicío
<b>Agente</b>	secretária
<b>Produto</b>	andamento dos processos judiciais dos imóveis
<b>Recurso</b>	cadastro de imóveis informações do processo judicial (número, vara e comarca)
<b>Anotação</b>	

FIGURA 7.21 - Requisito “informar dados do processo judicial”

<b>Requisito</b>	<b>Nome: informar andamento do processo</b>
	<i>Descrição:</i> informa o andamento do processo judicial sobre o imóvel, através de extratos judiciais de andamento; a liberação do imóvel sob judicío coloca-o disponível para a venda ou usufruto
<b>Agente</b>	secretária
<b>Produto</b>	andamento dos processos judiciais dos imóveis
<b>Recurso</b>	cadastro de imóveis andamento dos processos judiciais dos imóveis informações sobre o andamento do processo (data e extrato do andamento)
<b>Anotação</b>	

FIGURA 7.22 - Requisito “informar andamento do processo”

<b>Requisito</b>	<b>Nome: vender imóvel</b>
	<i>Descrição:</i> atualizar a situação do imóvel por venda; o imóvel deixa de ser patrimônio das empresas mas seu cadastramento é mantido
<b>Agente</b>	secretária
<b>Produto</b>	cadastro de imóveis
<b>Recurso</b>	cadastro de imóveis informações de venda (data, comprador, valor)
<b>Anotação</b>	

FIGURA 7.23 - Requisito “vender imóvel”

Os cenários associados aos requisitos do objetivo de "manutenção da situação dos imóveis" foram: informar processo judicial do imóvel (figura 7.24), atualizar situação judicial do imóvel (figura 7.25) e vender imóvel (figura 7.26).

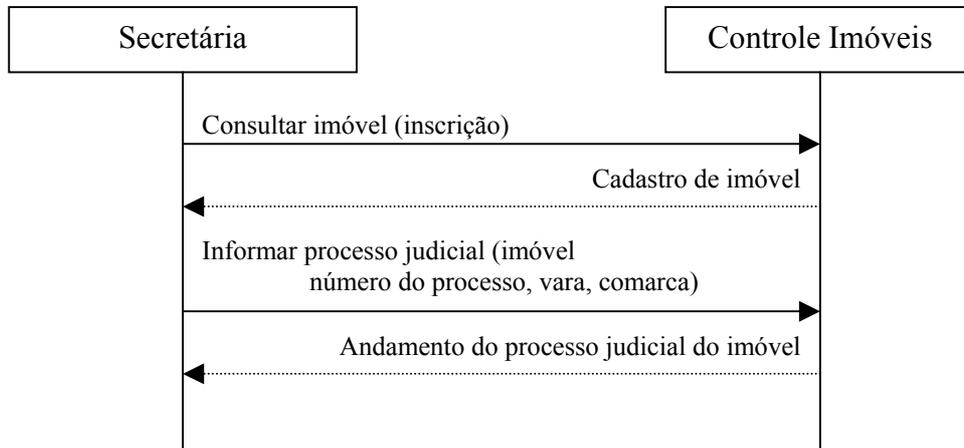


FIGURA 7.24 Cenário de informar processo judicial do imóvel

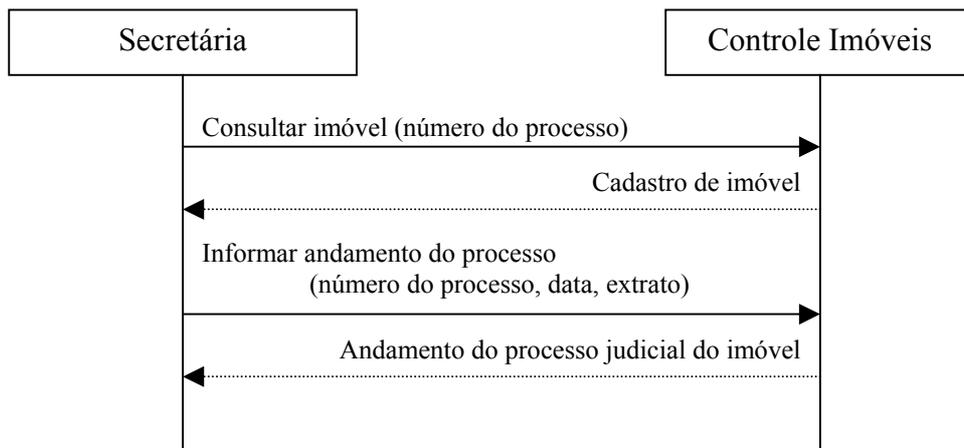


FIGURA 7.25 - Cenário de atualizar situação judicial do imóvel

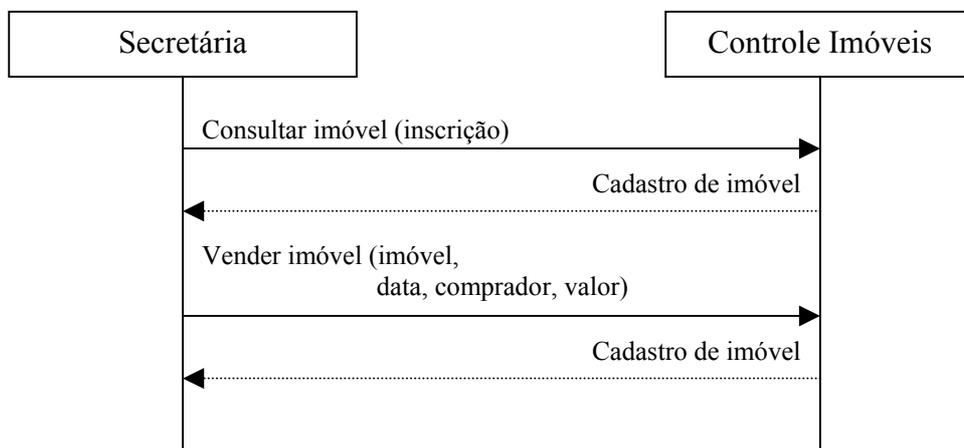


FIGURA 7.26 - Cenário de vender imóvel

Do objetivo de "controle da situação dos imóveis", obtiveram-se os requisitos: "consultar imóvel" (figura 7.28), que compreende "consultar imóvel por inscrição" (figura 7.29) ou "consultar imóvel por processo" (figura 7.30), conforme estrutura representada pelo esquema a seguir (figura 7.27).

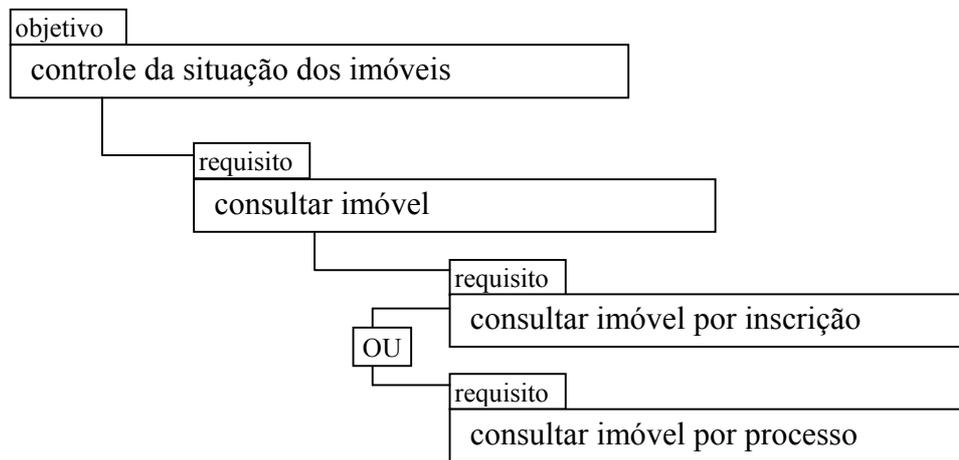


FIGURA 7.27 - Estrutura dos requisitos do objetivo de controle da situação dos imóveis

<b>Requisito</b>	<b>Nome: consultar imóvel</b>
	<i>Descrição:</i> consulta completa do cadastro do imóvel e do andamento dos processos judiciais associados ao imóvel
<b>Agente</b>	administrador
<b>Produto</b>	controle da situação dos imóveis
<b>Recurso</b>	cadastro de imóveis cadastro de loteamentos cadastro de imobiliárias cadastro de inquilinos andamento dos processos judiciais dos imóveis
<b>Anotação</b>	

FIGURA 7.28 - Requisito “consultar imóvel”

<b>Requisito</b>	<b>Nome: consultar imóvel por inscrição</b>
	<i>Descrição:</i> consulta completa do cadastro do imóvel, informado a sua inscrição na prefeitura (constante no carnê do IPTU) ou informação de endereço ou quadra e número do lote
<b>Agente</b>	administrador
<b>Produto</b>	controle da situação dos imóveis
<b>Recurso</b>	cadastro de imóveis cadastro de loteamentos cadastro de imobiliárias cadastro de inquilinos andamento dos processos judiciais dos imóveis
<b>Anotação</b>	

FIGURA 7.29 - Requisito “consultar imóvel por inscrição”

<b>Requisito</b>	<b>Nome: consultar imóvel por processo</b>
	<i>Descrição:</i> consulta completa do cadastro do imóvel e andamento do processo judicial associado, informado o número do processo
<b>Agente</b>	administrador
<b>Produto</b>	controle da situação dos imóveis
<b>Recurso</b>	cadastro de imóveis cadastro de loteamentos cadastro de imobiliárias cadastro de inquilinos andamento dos processos judiciais dos imóveis
<b>Anotação</b>	

FIGURA 7.30 - Requisito “consultar imóvel por processo”

Os cenários associados aos requisitos do objetivo de "controle da situação dos imóveis" foram: consultar imóvel por inscrição (figura 7.31) e consultar imóvel por processo (figura 7.32) .

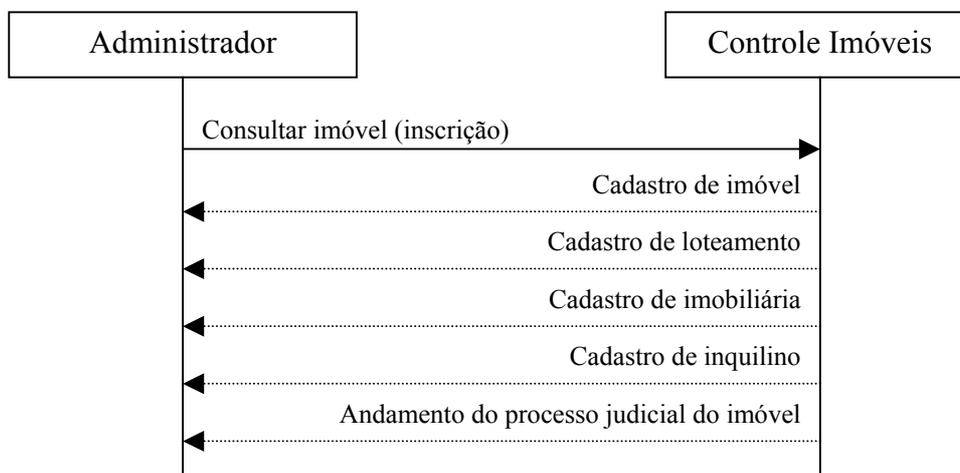


FIGURA 7.31 - Cenário de consultar imóvel por inscrição

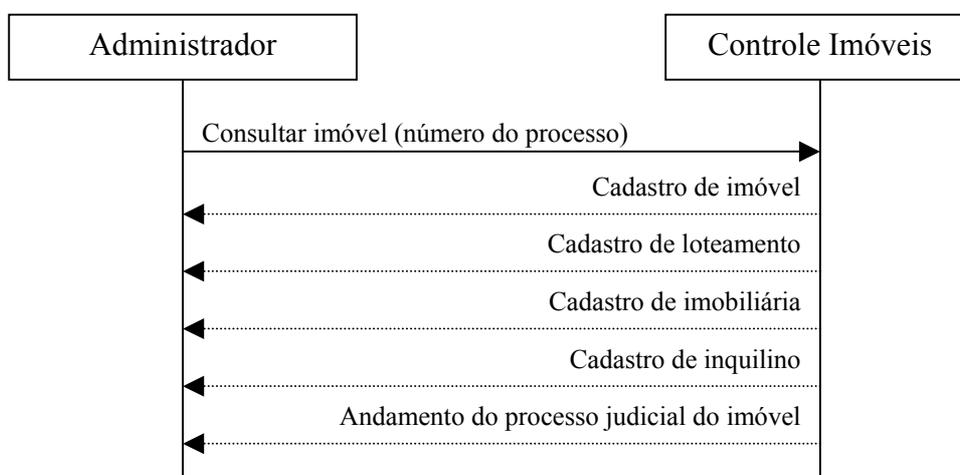


FIGURA 7.32 - Cenário de consultar imóvel por processo

O glossário de termos que definem os requisitos do caso do Controle de Imóveis é apresentado abaixo (figura 7.33).

<b>Termo</b>	<b>Definição</b>
secretária	á a pessoa responsável pelo cadastramento e manutenção das informações relacionadas aos imóveis e processos judiciais
administrador	é a pessoa responsável pela gestão das informações, quem as utiliza para a administração dos imóveis e tomada de decisões acerca deles
loteamento	áreas onde situam-se conjuntos de lotes de propriedade das empresas, podendo também haver edificações sobre os mesmos
imóvel	bem patrimonial das empresas; pode ser um lote que é uma porção de área localizada em algum loteamento e/ou edificação que consiste em uma construção em um lote ou outro endereço conhecido (casa, apartamento, sala ou box)
proprietário do imóvel	empresa do grupo que detém a propriedade do imóvel
registro de imóvel	dados do registro do imóvel em um cartório de imóveis
contrato de locação	documento que registra acordo de aluguel de algum imóvel (edificação) entre a administradora do imóvel e o inquilino
imobiliária	administradora das locações de imóveis
inquilino	indivíduo locador de um imóvel (casa, apartamento, sala ou box)
processo judicial	pleito judicial a respeito de um imóvel
extrato do processo	síntese de um trâmite de um processo judicial; o extrato do processo reflete a situação do andamento do processo jurídico

FIGURA 7.33 - Glossário de termos dos requisitos do controle de imóveis

Para o caso do Controle de Imóveis, os requisitos foram modelados em três diagramas de casos de uso (*use case*): cadastramento de imóveis (figura 7.34), manutenção da situação dos imóveis (figura 7.35) e controle da situação dos imóveis (figura 7.36). Esses caso de uso representam as três principais atividades a serem realizadas com a utilização do software.

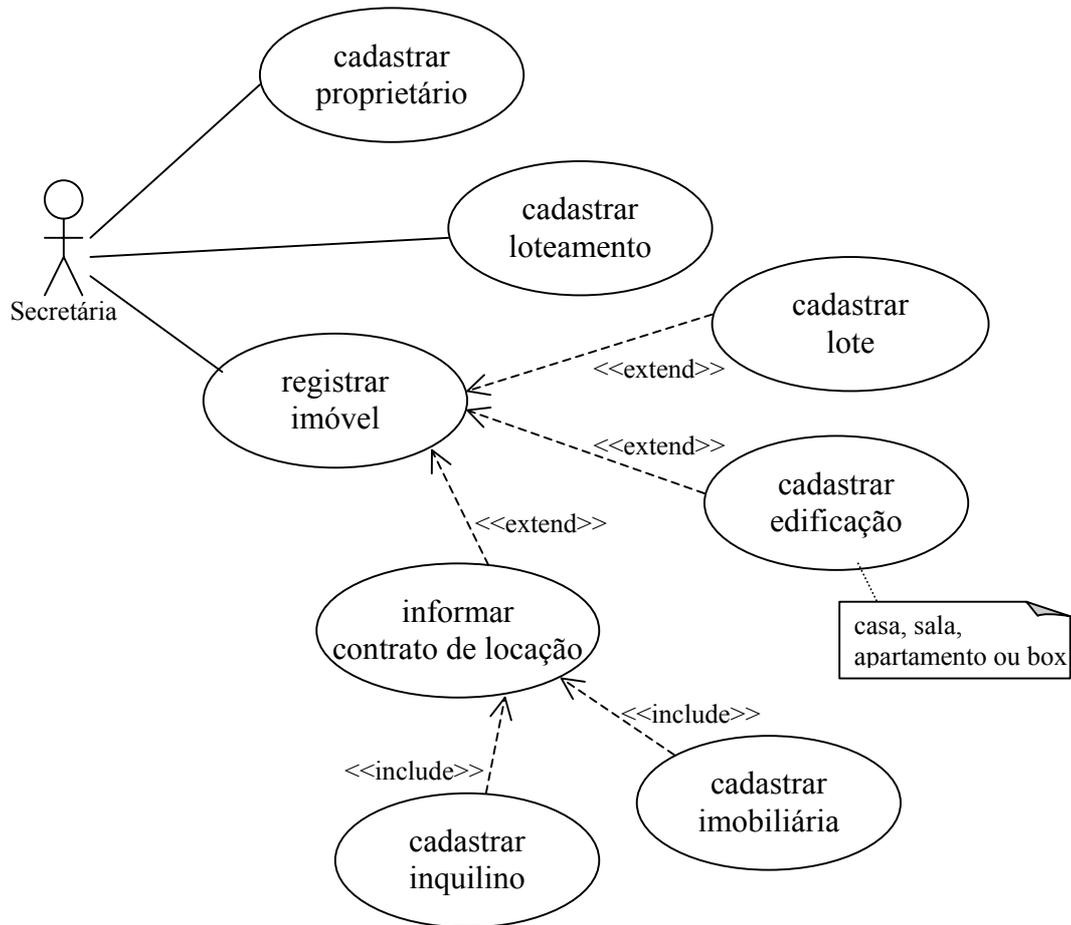
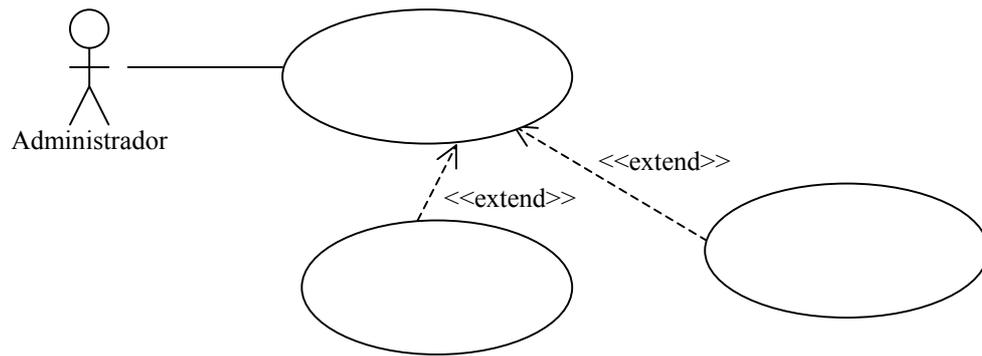
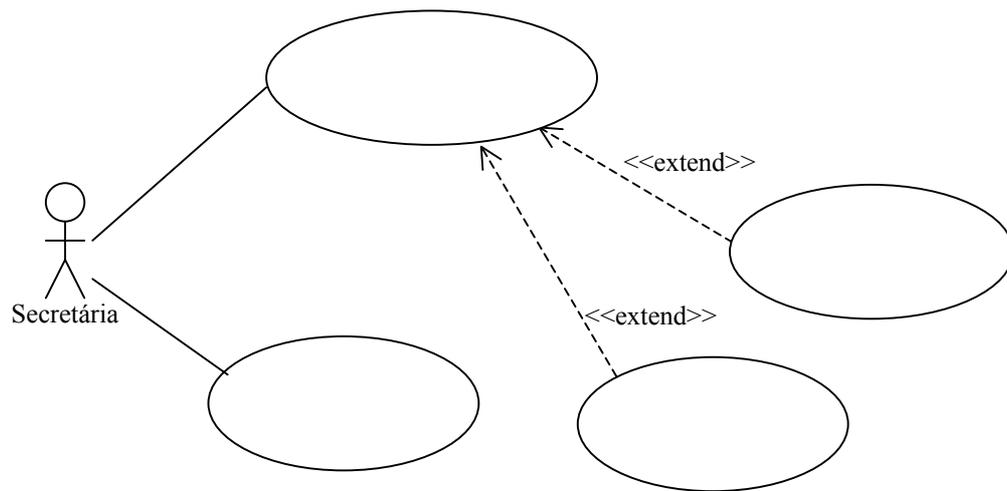


FIGURA 7.34 - Casos de uso de cadastramento de imóveis



A partir do modelo de requisitos são gerados os modelos de análise, segundo o Método Fusion. O modelo de requisitos participa do processo Fusion na etapa de documentação de requisitos (figura 6.4), para qual o método não prevê modelo de representação. São gerados, então, o modelo de objetos (figura 7.37) e os modelo de interface, ciclo-de-vida (figura 7.38) e de operações: *cadastrarProprietário* (figura 7.39), *cadastrarLoteamento* (figura 7.40), *registrarImóvel* (figura 7.41), *cadastrarLote* (figura 7.42), *cadastrarEdificação* (figura 7.43), *cadastrarImobiliária* (figura 7.44), *cadastrarInquilino* (figura 7.45), *informarContratoLocação* (figura 7.46), *informarProcesso* (figura 7.47), *informarAndamento* (figura 7.48), *venderImóvel* (figura 7.49), *consultarImóvelInscrição* (figura 7.50) e *consultarImóvelProcesso* (figura 7.51).

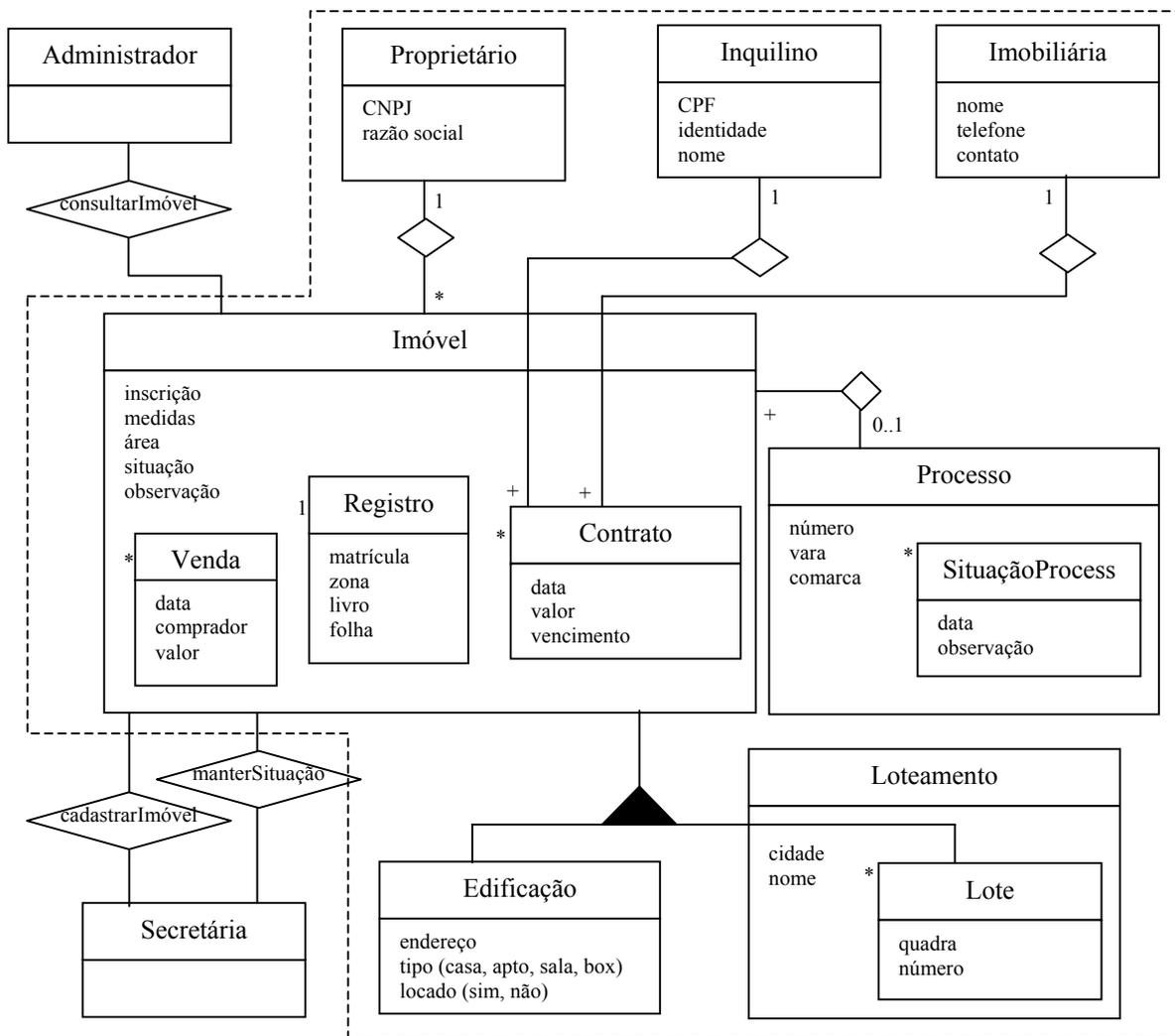


FIGURA 7.37 - Modelo de objetos do controle de imóveis

<i>LifeCycle</i>	= ( cadastrarImóvel . manterSituaçãoImóvel*   consultarImóvel )*
cadastrarImóvel	= ( [cadastrarProprietário . #CadasatroProprietário] . [cadastrarLoteamento . #CadastroLoteamento] . registrarImóvel . (cadastrarLote   cadastrarEdificação)* . [ [cadastrarImobiliária . #CadastroImobiliária] . [cadastrarInquilino . #CadastroInquilino] . informarContratoLocação ] ) . #ConfirmaçãoImóvel
manterSituaçãoImóvel	= consultarImóvel . ( (informarProcesso   informarAndamento) . #AndamentoProcesso )   ( venderImóvel . #CadastroImóvel )
consultarImóvel	= ( consultarImóvelInscrição   consultarImóvelProcesso ) . #CadastroImóvel . [#CadastroLoteamento] . [#CadastroImobiliária . #CadastroInquilino] . [#AndamentoProcesso]

FIGURA 7.38 - Modelo ciclo-de-vida do controle de imóveis

<i>Operation</i>	cadastrarProprietário
<i>Description</i>	cadastrar a empresa proprietária de imóveis
<i>Reads</i>	<i>supplied</i> CNPJ, razãosocial
<i>Changes</i>	Proprietário
<i>Sends</i>	Secretária: {CadastroPropreitário}
<i>Assumes</i>	
<i>Result</i>	

FIGURA 7.39 - Operação cadastrarProprietário

<i>Operation</i>	cadastrarLoteamento
<i>Description</i>	cadastrar o loteamento onde localizam-se os imóveis
<i>Reads</i>	<i>supplied</i> nome, localização
<i>Changes</i>	Loteamento
<i>Sends</i>	Secretária: {CadastroLoteamento}
<i>Assumes</i>	
<i>Result</i>	

FIGURA 7.40 - Operação cadastrarLoteamento

<i>Operation</i>	registrarImóvel
<i>Description</i>	cadastrar o imóvel, com os dados do registro de imóveis
<i>Reads</i>	<i>supplied</i> inscrição, matrícula, zona, livro, folha, medidas, área
<i>Changes</i>	Imóvel, Registro
<i>Sends</i>	
<i>Assumes</i>	
<i>Result</i>	É criada um objeto Imóvel, com o Registro, em situação de “livre”. Esta operação é executada com cadastrarLote e/ou cadastrarEdificação.

FIGURA 7.41 - Operação registrarImóvel

<i>Operation</i>	cadastrarLote
<i>Description</i>	cadastrar dados do lote
<i>Reads</i>	Imóvel, Loteamento <i>supplied</i> inscrição, quadra, número
<i>Changes</i>	Lote
<i>Sends</i>	Secretária: {CadastroImóvel}
<i>Assumes</i>	Existe objeto Imóvel, identificado por inscrição.
<i>Result</i>	

FIGURA 7.42 - Operação cadastrarLote

<i>Operation</i>	cadastrarEdificação
<i>Description</i>	cadastrar dados da edificação
<i>Reads</i>	Imóvel <i>supplied</i> inscrição, endereço, tipoedificação
<i>Changes</i>	Lote
<i>Sends</i>	Secretária: {CadastroImóvel}
<i>Assumes</i>	Existe objeto Imóvel, identificado por inscrição. Valor de tipoedificação pertence a (casa, sala, apartamento, box).
<i>Result</i>	

FIGURA 7.43 - Operação cadastrarEdificação

<i>Operation</i>	cadastrarImobiliária
<i>Description</i>	cadastrar a imobiliária responsável por locações de imóveis
<i>Reads</i>	<i>supplied</i> nome, contato, telefone
<i>Changes</i>	Imobiliária
<i>Sends</i>	Secretária: {CadastroImobiliária}
<i>Assumes</i>	
<i>Result</i>	

FIGURA 7.44 - Operação cadastrarImobiliária

<i>Operation</i>	cadastrarInquilino
<i>Description</i>	cadastrar o inquilino, locador de imóveis
<i>Reads</i>	<i>supplied</i> nome, identidade, CPF
<i>Changes</i>	Inquilino
<i>Sends</i>	Secretária: {CadastroInquilino}
<i>Assumes</i>	
<i>Result</i>	

FIGURA 7.45 - Operação cadastrarInquilino

<i>Operation</i>	informarContratoLocação
<i>Description</i>	informar dados do contrato de locação
<i>Reads</i>	Imóvel, Imobiliária, Inquilino <i>supplied</i> datacontrato, vencimento, valor
<i>Changes</i>	Contrato
<i>Sends</i>	Secretária: {CadastroImóvel}
<i>Assumes</i>	Existe objeto Imóvel, identificado por inscrição. Existem objetos Imobiliária e Inquilino, que são associados a Contrato.
<i>Result</i>	

FIGURA 7.46 - Operação informarContratoLocação

<i>Operation</i>	informarProcesso
<i>Description</i>	cadastrar os dados do processo judicial associado ao imóvel sob júdice
<i>Reads</i>	<i>supplied</i> númeroprocesso, vara, comarca
<i>Changes</i>	Imóvel, Processo
<i>Sends</i>	Secretária: {AndamentoProcesso}
<i>Assumes</i>	Existe objeto Imóvel, identificado por inscrição.
<i>Result</i>	Altera situação do Imóvel para “sob júdice”.

FIGURA 7.47 - Operação informarProcesso

<i>Operation</i>	informarAndamento
<i>Description</i>	informa o andamento do processo judicial sobre o imóvel, através de extratos judiciais de andamento; a liberação do imóvel sob júdice coloca-o disponível para a venda ou usufruto
<i>Reads</i>	Processo <i>supplied</i> númeroprocesso, data, extrato
<i>Changes</i>	Imóvel, SituaçãoProcesso
<i>Sends</i>	Secretária: {AndamentoProcesso}
<i>Assumes</i>	Existe objeto Processo, identificado por númeroprocesso.
<i>Result</i>	Altera situação do Imóvel para “livre”, conforme andamento do processo.

FIGURA 7.48 - Operação informarAndamento

<i>Operation</i>	venderImóvel
<i>Description</i>	atualizar a situação do imóvel por venda; o imóvel deixa de ser patrimônio das empresas mas seu cadastramento é mantido
<i>Reads</i>	<i>supplied</i> inscrição, data, comprador, valor
<i>Changes</i>	Imóvel, Venda
<i>Sends</i>	Secretária: {CadastroImóvel}
<i>Assumes</i>	Existe objeto Imóvel, identificado por inscrição.
<i>Result</i>	Altera situação do Imóvel para “vendido”.

FIGURA 7.49 - Operação venderImóvel

<i>Operation</i>	consultarImóvelInscrição
<i>Description</i>	consulta completa do cadastro do imóvel, informado a sua inscrição na prefeitura (constante no carnê do IPTU) ou informação de endereço ou quadra e número do lote
<i>Reads</i>	<i>supplied</i> inscrição Imóvel, Loteamento, Imobiliária, Inquilino, Processo
<i>Changes</i>	
<i>Sends</i>	Administrador: {CadastroImóvel, CadastroLoteamento, CadastroImobiliária, CadastroInquilino, AndamentoProcesso}
<i>Assumes</i>	Existe objeto Imóvel, identificado por inscrição.
<i>Result</i>	

FIGURA 7.50 - Operação consultarImóvelInscrição

<i>Operation</i>	consultarImóvelProcesso
<i>Description</i>	consulta completa do cadastro do imóvel e andamento do processo judicial associado, informado o número do processo
<i>Reads</i>	<i>supplied</i> numeroprocesso Imóvel, Loteamento, Imobiliária, Inquilino, Processo
<i>Changes</i>	
<i>Sends</i>	Administrador: {CadastroImóvel, CadastroLoteamento, CadastroImobiliária, CadastroInquilino, AndamentoProcesso}
<i>Assumes</i>	Existe objeto Processo, identificado por numeroprocesso.
<i>Result</i>	

FIGURA 7.51 - Operação consultarImóvelProcesso

#### *7.2.4 Avaliação da Aplicação do Modelo de Estruturação de Requisitos*

A realização dos três estudos de caso teve o propósito de avaliar a aplicação de diferentes técnicas de elicitação com abordagens diferentes na estruturação dos requisitos segundo o modelo proposto, visando validar o processo de modelagem.

No primeiro caso, o controle de serviços operacionais, foram aplicadas técnicas sem estruturação. Neste caso pode-se observar que, naturalmente, os envolvidos trataram o domínio do problema segundo as atividades realizadas e os objetivos almejados. Essa organização espontânea apontou favoravelmente a aplicação de uma abordagem voltada às atividades, segundo preceitos da Teoria da Atividade.

No segundo caso, o sistema de protocolo, a aplicação das técnicas de elicitação foi orientada a definir as atividades do sistema e tratar cada atividade modularmente, definido um escopo para o tratamento do domínio do problema. Essa abordagem permitiu uma melhor organização dos requisitos e, conseqüentemente, uma maneira mais harmoniosa para a estruturação segundo o modelo proposto.

No terceiro caso, o controle de imóveis, a abordagem dada a elicitação e modelagem dos requisitos foi perfeitamente afinada ao modelo proposto. Inicialmente, definiu-se as atividades, que estabeleceram as diretrizes para o processo de elicitação, com a aplicação de técnicas estruturadas por objetivos. A elicitação das informações com os envolvidos foi diretamente direcionado a estruturação dos requisitos e definição dos objetos que os compõem, segundo o modelo proposto. Basicamente, trabalhou-se diretamente sobre o modelo.

Neste último caso o processo foi desenvolvido até a definição dos modelos de análise, segundo o Método Fusion. Pode-se, então, ter uma avaliação sobre os aspectos de rastreabilidade e a usabilidade do modelo na validação dos requisitos especificados. Também, avaliou-se o grau de contribuição que o modelo proposto agrega a atividade de análise no ciclo de desenvolvimento do software.

O modelo propõe uma estruturação de requisitos, com uma abordagem direcionada às atividades e objetivos. Em uma conclusão preliminar, essa abordagem mostrou-se natural a definição das necessidades entre os envolvidos, o que facilitou a aquisição do conhecimento do sistema. A aplicação de uma técnica de elicitação estruturada, conforme estrutura proposta no modelo, colabora muito fortemente para a modelagem dos requisitos. Os estudos de caso demonstraram que a estruturação dos requisitos é um fator que qualifica o processo de desenvolvimento do software, e a sintonia entre as atividades de elicitação e modelagem contribuem decisivamente para o sucesso do software, considerando a qualidade de precisão e a validação da definição das necessidades.

## 8 Conclusão

O estudo realizado neste trabalho teve como diretriz a perspectiva de propor um modelo adequado à estruturação de requisitos segundo uma abordagem voltada aos clientes e usuários do pretense software. Inserida no contexto do Projeto FILM, cujo objetivo é expandir o Método Fusion agregando uma etapa de modelagem de requisitos, a proposta estabeleceu a qualidade de usabilidade como motivadora da definição de um modelo de estruturação de requisitos.

Deve-se entender o termo usabilidade, neste trabalho, como a facilidade com que o modelo é utilizado para representar os requisitos de forma inteligível para os clientes e usuários e para os engenheiros de software, atuando tanto como uma estrutura para a especificação e acompanhamento do desenvolvimento dos requisitos, quanto como uma ferramenta conversacional para a validação dos mesmos entre os envolvidos.

Buscando-se fundamentação nas abordagens orientadas a objetivos e nos princípios da Teoria da Atividade, o modelo foi concebido de forma a prover uma construção gradual e incremental do entendimento compartilhado sobre os requisitos do sistema. A abordagem por objetivos e atividades permite uma segmentação eficiente para o trato do domínio do problema. A organização hierárquica em atividades, ações e operações contribui positivamente para a especificação incremental das necessidades de desenvolvimento, desde os objetivos até os requisitos funcionais do software.

A circunscrição de escopos do problema, delimitados pelas atividades, pôde ser avaliada como harmônica ao processo de aquisição e tratamento do conhecimento, e positiva para o desenvolvimento de uma solução modularizada. Contribui diretamente para facilitar a modelagem do software e estabelecer um vínculo entre os objetivos e os componentes de software, provendo a rastreabilidade do desenvolvimento dos requisitos.

A rastreabilidade permite um acompanhamento do desenvolvimento dos requisitos, colaborando com a tarefa de validação do software. Essa característica é promovida pela estrutura hierárquica dos objetivos e requisitos, e a amarração desses últimos aos modelos da fase de análise. A sistematização do Método Fusion coopera diretamente ao provimento da rastreabilidade, dadas as ligações entre os seus modelos da fase de análise e projeto. Assim, pode-se relacionar os requisitos aos componentes de software implementados, facilitando a avaliação se tais componentes atendem aos requisitos especificados.

O modelo é apropriado à representação de requisitos funcionais, permitindo a identificação de objetos que definem esse requisitos e compõem os modelos de projeto do software. Os requisitos não funcionais são tratados como anotações aos requisitos funcionais. A proposta não trabalha tais requisitos, apenas oferece uma maneira de, tendo-os descritos, associá-los aos requisitos funcionais e, conseqüentemente, rastrear seu desenvolvimento. Entretanto, a avaliação, no que tange a atendimento a requisitos não funcionais, depende diretamente da habilidade dos envolvidos em perceber se o que está descrito está sendo atendido.

O estabelecimento de um guia básico para a geração de diagramas de casos de uso a partir do modelo proposto, apresentado no trabalho, amplia a aplicação do modelo a outras metodologias de desenvolvimento que utilizam tais diagramas no processo de modelagem do software. Os casos de uso são aplicados para capturar o comportamento pretendido do software, fornecem uma maneira para os desenvolvedores chegarem a uma compreensão comum com os usuários finais e

especialistas do domínio do sistema e ajudam a validar a arquitetura e verificar a evolução do software durante o desenvolvimento [BOO2000].

O modelo proposto foi aplicado a três casos, escolhidos pela viabilidade de estudo em função do tempo de desenvolvimento do trabalho. Embora os casos não apresentaram situações de maior complexidade em termos de engenharia de requisitos, pode-se avaliar que o modelo dispõe de qualidades que permitem a organização dos requisitos independentemente do porte do sistema.

A aplicação do modelo de forma modular, por objetivos, e incremental, no sentido de tratar diversos níveis de abstração e refinamento dos requisitos, agrega ao processo de modelagem facilidade para o seu gerenciamento. Identificados os objetivos, e as atividades, é possível estabelecer diretrizes para a abordagem dos requisitos através de escopos do domínio do problema. Essas diretrizes, aliadas à estruturação dos requisitos em níveis de abstração, permitem definir e estruturar técnicas de elicitação e modelagem adequadas a um processo gradual de desenvolvimento dos requisitos. Esse processo gradual facilita intensamente a compreensão e o tratamento do conhecimento do sistema e as necessidades dos clientes e usuários.

Em cada caso estudado, a metodologia de aplicação de técnicas de elicitação e modelagem foi progressiva no sentido de afinar-se ao modelo proposto. Desde o primeiro caso, no qual foram utilizadas técnicas de elicitação mais livres, até o terceiro caso, no qual as técnicas empregadas forma estruturadas conforme a abordagem proposta, pode-se perceber uma gradativa facilidade em trabalhar os requisitos.

A abordagem por atividades, segundo os preceitos da Teoria da Atividade, foi reconhecidamente aplicável e válida aos estudos de casos. A segmentação do domínio do problema por atividades, no qual são definidos os objetivos almejados e seus requisitos derivados, apresenta-se como uma maneira natural ao entendimento e tratamento do sistema, e positiva para o desenvolvimento da solução, já que colabora diretamente na modelagem do software delineando as necessidades.

Os cenários também constituem uma ferramenta muito prática para a análise e validação dos requisitos. Demonstrou ser um modelo inteligível para análise do comportamento do sistema, considerando a sua grafia visual, natural para a compreensão dos eventos que ocorrem na interação entre agentes externos e o software. Embora nos casos estudados foram poucos os cenários desenhados, dado o prévio conhecimento sobre a lógica básica dos negócios por parte do engenheiro de software, o modelo é flexível quanto à construção de cenários, haja visto a multiplicidade de associação deles com os requisitos do modelo. Pode-se, inclusive, especificar cenários positivos, que confirmam os requisitos, e cenários negativos, que representam situações de exceção ao atendimento aos requisitos. A representação de cenários positivos e negativos colaboram efetivamente na elaboração de casos de teste para a validação dos requisitos do software.

Dentre os casos desenvolvidos, no Controle de Imóveis, por ter sido estruturado mais afinadamente à abordagem proposta e terem sido desenvolvidos alguns protótipos, ficou mais evidente a facilidade de avaliação do software provida pelo modelo. A estruturação dos requisitos permitiu um vínculo dos componentes do software aos requisitos especificados no modelo, possibilitando a associação entre as atividades e ações do sistema e os protótipos de interface construídos com base nos requisitos e cenários já discutidos entre os envolvidos. Os clientes e usuários puderam, assim, validar mais facilmente o software com base nessa associação entre o objetivo definido e as interfaces que implementam as operações providas pelo software para o seu atendimento.

A atividade de validação dos requisitos proposta não é formal, depende da percepção e avaliação dos clientes e usuários. Entretanto, a maneira com que os requisitos são estruturados e organizados permite uma boa rastreabilidade e acompanhamento do seu desenvolvimento. Pode-se, ainda que empiricamente, avaliar os modelos de software e seus conseqüentes componentes executáveis e comparar com os requisitos aos quais estão ligados. Observa-se que, como a estrutura dos requisitos parte dos objetivos, pode-se validar se os mesmos são atendidos pelo software em execução.

O modelo está bem estruturado, fundamentado nos princípios da Teoria da Atividade e na especificação de objetos que descrevem o comportamento requerido para o software. Os objetivos e requisitos possuem uma estrutura estática com uma semântica bem definida. O processo de elicitação e modelagem requisitos apresentado é afinado à estrutura do modelo proposto. A rastreabilidade prevista estabelece um vínculo entre o modelo de requisitos e os modelos do Método Fusion ou os diagramas de casos de uso, permitindo a amarração entre a fase de elaboração (definição dos requisitos) e a fase de construção (projeto do software). Esses fatores colaboram para a especificação e desenvolvimento de uma ferramenta apropriada para o processo de modelagem de requisitos, segundo a proposta apresentada. O desenvolvimento desta ferramenta de Engenharia de Software é objetivo do Projeto FILM.

O modelo proposto pôde ser avaliado como uma ferramenta para a modelagem de requisitos no Método Fusion, conforme objetivo do trabalho. Demonstrou-se adequado a organização dos requisitos e afinado à sistematização do método, integrando-se apropriadamente ao processo e ajustando-se aos modelos da fase de análise. O modelo proposto mantém a orientação e gerenciamento, peculiar ao Fusion, para a fase de especificação de requisitos e a conseqüente integração com as fases subseqüentes.

No contexto do Projeto FILM, a contribuição do modelo reside na apropriação ao Método Fusion, com uma estruturação adequada a modelagem comportamental através de diagramas de casos de uso, providos pela UML. A modelagem por objetivos, segundo princípios da Teoria da Atividade, aliada a hierarquia do modelo, permitem, também, um tratamento cíclico ao processo de desenvolvimento do software. Pode-se, com o modelo, estabelecer ciclos de desenvolvimento, priorizando objetivos, o que vem a colaborar diretamente com as diretrizes do projeto FILM no que tange a adaptação do Método Fusion à UML.

O modelo, em suma, atendeu aos objetivos a que se propunha: estruturar os requisitos funcionais em uma abordagem voltada ao usuário, provendo a qualidade de usabilidade. A estruturação proposta permite uma organização dos requisitos, facilitando a continuidade do processo de desenvolvimento e permitindo a manutenção de vínculos que relacionam as necessidades dos cliente e usuários aos modelos de definição do software e, conseqüentemente, aos seus produtos executáveis.

O modelo proposto foi aplicado a casos tradicionais de desenvolvimento de sistemas de informação e pôde ser avaliado como uma ferramenta aplicável para o propósito da atividade de modelagem de requisitos. A fundamentação conceitual do modelo e a usabilidade provida proporcionou o estabelecimento de um processo de especificação de requisitos gerenciável segundo uma abordagem voltada às necessidades dos usuários. A conseqüência é uma precisão maior na definição dos requisitos e uma facilidade na validação do software, objetivando a satisfação das necessidades dos clientes e usuários.

## **Anexo 1 Estudo de Caso: Sistema de Protocolo**

### **Breve descrição do contexto da aplicação:**

*O SAMAE (Serviço Autônomo Municipal de Água e Esgoto) é uma autarquia responsável pelo saneamento da cidade de Caxias do Sul. Compreende saneamento os serviços de captação, tratamento e distribuição de água, coleta e tratamento do esgoto sanitário e gestão dos recursos hídricos da cidade. Associados a esse fluxo básico, têm-se serviços relacionados a essas atividades que são solicitados pela população, ou pelos próprios servidores. Exemplos desses serviços são ligação de água e manutenção de cadastro de usuários, projetos de redes de abastecimento, licitações e cadastramento de fornecedores, além de processo com assuntos administrativos.*

*A solicitação desses serviços deve ser realizada através de um protocolo. Segundo a estrutura organizacional do SAMAE, a seção de atendimento, sob a gerência de atendimento ao público, da divisão comercial, é o setor responsável pelo protocolo das solicitações. É o início do atendimento da solicitação, onde é aberto um processo administrativo que tramita pela autarquia, pelos setores responsáveis por cada ação, até a execução completa do serviço ou indeferimento da solicitação.*

*Atualmente já existe um sistema informatizado atendendo à aplicação, porém, o SAMAE passou por uma evolução tecnológica da infra-estrutura de informática. O objetivo é o desenvolvimento de um novo software adequado às novas necessidades e atualizado tecnologicamente, harmonizado à nova realidade de informática presente no SAMAE.*

*O propósito da aplicação é manter informações sobre os processos administrativos protocolados e suportar as atividades que envolve o andamento e a situação dos processos. A utilização do aplicativo deve ser difundida entre todos os setores da autarquia.*

**Objetivos:**

<b>Objetivo</b>	<i>Um objetivo é manter um cadastro de processos administrativos abertos, com todas as informações necessárias para que a autarquia possa analisar e atender, ou indeferir, as solicitações.</i>
<b>Resultado</b>	<i>Qual o resultado esperado? Por que há a necessidade?</i> cadastro de processo administrativo
<b>Objeto</b>	<i>Para o que esse objetivo faz-se necessário ao sistema?</i> cadastro de processos administrativos
<b>Sujeito</b>	<i>Quem atua em prol desse objetivo?</i> setores da autarquia
<b>Ferramenta</b>	<i>Quais são os recursos necessários para os sujeitos interagirem com o objeto?</i> conhecimento sobre os assuntos conhecimento dos usuários documentação do processo administrativo

<b>Objetivo</b>	<i>Outro objetivo é ter-se o controle sobre o andamento dos processos administrativos, com informações sobre suas situações e também informações gerenciais para avaliação do atendimento às solicitações.</i>
<b>Resultado</b>	<i>Qual o resultado esperado? Por que há a necessidade?</i> acompanhamento do processo administrativo
<b>Objeto</b>	<i>Para o que esse objetivo faz-se necessário ao sistema?</i> controle do andamento dos processos administrativos
<b>Sujeito</b>	<i>Quem atua em prol desse objetivo?</i> setores da autarquia
<b>Ferramenta</b>	<i>Quais são os recursos necessários para os sujeitos interagirem com o objeto?</i> organograma da autarquia cadastro de funcionários da autarquia processo administrativo (documento em papel)

**Glossário de Termos:**

<b>Termo</b>	<b>Definição</b>
processo administrativo	documento de uso interno da autarquia para o atendimento de solicitações que requeiram o armazenamento de alguma documentação ou que existe alguma determinação legal para a manutenção do registro da solicitação
setor de protocolo	representa os servidores responsáveis pelo serviço de protocolo, incluindo as tarefas de prover informações para o funcionamento do sistema e gerenciamento do serviço
setor de atendimento	representa os servidores responsáveis pelo atendimento a população, incluindo a abertura de processos administrativos e prestação de informação sobre o andamento dos mesmos
setores da autarquia	representa os servidores autorizados a realizar as tarefas relacionadas ao trâmite de processos administrativos nos setores que compõem o organograma da autarquia, incluindo os servidores dos setores de protocolo e de atendimento
número do processo	identificação unívoca do processo administrativo; os processos são numerados seqüencialmente durante o ano corrente (iniciando pelo número um na virada do ano)
assunto do processo	assunto que trata a solicitação de serviço ou informação por processo administrativo; o assunto determina o encaminhamento do processo pelos setores da autarquia, indicando o trâmite inicial
requerente	usuário que protocola alguma solicitação de serviço ou informação da autarquia, através da abertura de um processo administrativo
requerimento do processo	documento de abertura de processo administrativo que informa o número do processo, o assunto tratado, o requerente que solicitou e a solicitação protocolada
documentação do processo	documentos juntados ao processo na abertura ou nos trâmites; depende do assunto e do parecer informado nos encaminhamentos
parecer do processo	pronunciamento escrito de um trâmite do processo administrativo sobre a ação executada pelo setor; todo o trâmite possui um parecer que pode ser acompanhado de documentação
guia de encaminhamento do processo administrativo	documento que acompanha o processo nas tramitações, contendo as informações do processo, setor de origem e setor de destino, com a informação dos responsáveis pelo encaminhamento e recebimento do processo administrativo nos setores

**Requisitos:**

<b>Requisito</b>	<i>Nome:</i> <b>cadastrar assunto</b> <i>Descrição:</i> o assunto é necessários para qualificar o processo administrativo; determina as informações necessárias e a documentação a ser juntada, além de definir o encaminhamento inicial (em qual setor inicia a análise da solicitação); quem informa a necessidade de criação de assunto são os setores
<b>Agente</b>	setor de protocolo
<b>Produto</b>	cadastro de assuntos
<b>Recurso</b>	informações sobre os assuntos (software existente)
<b>Anotação</b>	

<b>Requisito</b>	<i>Nome:</i> <b>cadastrar requerente</b> <i>Descrição:</i> o requerente é um usuário que requer algum serviço ou informação, através de um processo administrativo; o requerente é identificado no momento do atendimento, e é associado ao processo administrativo o qual está protocolando; também são requerentes os funcionários da autarquia que podem solicitar alguma informação ou serviço através de processo administrativo
<b>Agente</b>	setor de atendimento
<b>Produto</b>	cadastro de requerentes
<b>Recurso</b>	informações sobre os requerentes (software existente)
<b>Anotação</b>	

<b>Requisito</b>	<i>Nome:</i> <b>abrir processo administrativo</b>
	<i>Descrição:</i> a abertura de um processo administrativo ocorre por uma solicitação de serviço ou informação; a abertura de um processo administrativo, em geral, ocorre no setor de atendimento ao público; entretanto alguns processos (por exemplo, licitações) podem ser abertos em outros setores da autarquia; na abertura identifica-se o requerente e o assunto do processo, junta-se a documentação necessária e emite-se o requerimento e a etiqueta de identificação do processo administrativo; o processo aberto é encaminhado ao setor responsável por tratar do assunto solicitado; a abertura do processo administrativo é realizada em atendimento a um usuário dos serviços da autarquia, portanto deve prover uma interface adequada e um desempenho aceitável para a prestação de um atendimento de qualidade
<b>Agente</b>	setor de atendimento setores da autarquia
<b>Produto</b>	cadastro de processos administrativos
<b>Recurso</b>	cadastro de assuntos cadastro de setores cadastro de requerentes requerimento, etiqueta e documentos para o processo administrativo
<b>Anotação</b>	avaliar qualidades de funcionalidade, usabilidade e eficiência

<b>Requisito</b>	<i>Nome:</i> <b>cadastrar setor</b>
	<i>Descrição:</i> os setores são responsáveis por executar alguma tarefa específica, visando o atendimento a solicitação; são unidades funcionais por onde os processos administrativos passam e ocorre alguma ação em prol da solicitação
<b>Agente</b>	setor de protocolo
<b>Produto</b>	cadastro de setores
<b>Recurso</b>	organograma da autarquia
<b>Anotação</b>	

<b>Requisito</b>	<i>Nome:</i> <b>cadastrar autorizado</b> <i>Descrição:</i> os setores determinam quem são os funcionários autorizados a tramitar os processos administrativos pertinentes ao setor; os funcionários do setor de atendimento são autorizados a tramitar qualquer processo administrativo, independentemente de qual setor o processo se encontre
<b>Agente</b>	setores da autarquia
<b>Produto</b>	cadastro de autorizados
<b>Recurso</b>	cadastro de funcionários da autarquia
<b>Anotação</b>	

<b>Requisito</b>	<i>Nome:</i> <b>tramitar processo</b> <i>Descrição:</i> o processo tramita pelos setores conforme a ação a ser executada; é recebido pelos setor que realiza alguma tarefa, emite um parecer e encaminha-o, segundo o trâmite da solicitação; o recebimento e encaminhamento somente pode ser efetuado pelos autorizados do setor o qual encontra-se o processo, a exceção do setor de atendimento
<b>Agente</b>	setores da autarquia
<b>Produto</b>	controle do andamento dos processos administrativos
<b>Recurso</b>	cadastro de setores cadastro de autorizados cadastro de processos administrativos
<b>Anotação</b>	avaliar qualidade de confiabilidade (controle de permissões para execução de operação sobre informações)

<b>Requisito</b>	<i>Nome:</i> <b>receber processo</b> <i>Descrição:</i> o recebimento é a ação de registrar a entrada do processo no setor; deve ser comprovado para que o setor encaminhador tenha a certificação de que ele foi recebido pelo destino
<b>Agente</b>	setores da autarquia
<b>Produto</b>	controle do andamento dos processos administrativos
<b>Recurso</b>	cadastro de processos administrativos cadastro de setores e cadastro de autorizados controle do andamento dos processos administrativos guia de encaminhamento de processo administrativo
<b>Anotação</b>	avaliar qualidade de confiabilidade (mecanismo de garantia de encaminhamento e recebimento do processo)

<b>Requisito</b>	<i>Nome:</i> <b>encaminhar processo</b> <i>Descrição:</i> o encaminhamento é a ação de registrar a saída do processo do setor, e ocorre após o setor ter efetuado a ação solicitada e emitido um parecer; o setor encaminhador define qual o setor que deverá tomar a próxima ação, à exceção do encaminhamento inicial do processo, que é dado conforme o assunto; também ocorrem tramitações externas à autarquia, quando, então, o processo fica localizado no setor que o encaminhou, porém com o registro do órgão e responsável externo; para ter-se um conhecimento mais preciso dos trâmites é importante que o setor possa registrar um resumo do parecer; deve ter-se a certificação que o processo foi recebido pelo setor destino
<b>Agente</b>	setores da autarquia
<b>Produto</b>	controle do andamento dos processos administrativos
<b>Recurso</b>	cadastro de processos administrativos cadastro de setores e cadastro de autorizados controle do andamento dos processos administrativos guia de encaminhamento de processo administrativo parecer e documentação do trâmite do processo administrativo
<b>Anotação</b>	avaliar qualidade de confiabilidade (mecanismo de garantia de encaminhamento e recebimento do processo)

<b>Requisito</b>	<i>Nome:</i> <b>finalizar processo</b> <i>Descrição:</i> finalizado o trâmite do processo administrativo, com o atendimento, ou indeferimento, da solicitação, é necessário encerrá-lo e arquivá-lo; opcionalmente, pode-se reabrir o processo administrativo encerrado ou arquivado, caso em que é retomado o atendimento à solicitação, voltando o processo a tramitar normalmente; assim como na tramitação, somente os autorizados podem fazer o encerramento, arquivamento e reabertura de um processo administrativo
<b>Agente</b>	setores da autarquia
<b>Produto</b>	controle do andamento dos processos administrativos
<b>Recurso</b>	cadastro de setores e cadastro de autorizados cadastro de processos administrativos controle do andamento dos processos administrativos
<b>Anotação</b>	avaliar qualidade de confiabilidade (controle de permissões para execução de operação sobre informações)

<b>Requisito</b>	<i>Nome:</i> <b>encerrar processo</b>
	<i>Descrição:</i> o encerramento do processo consiste na informação do parecer final; é executado no setor que conclui o atendimento à solicitação
<b>Agente</b>	setores da autarquia
<b>Produto</b>	controle do andamento dos processos administrativos
<b>Recurso</b>	cadastro de setores e cadastro de autorizados cadastro de processos administrativos controle do andamento dos processos administrativos parecer final do processo administrativo
<b>Anotação</b>	

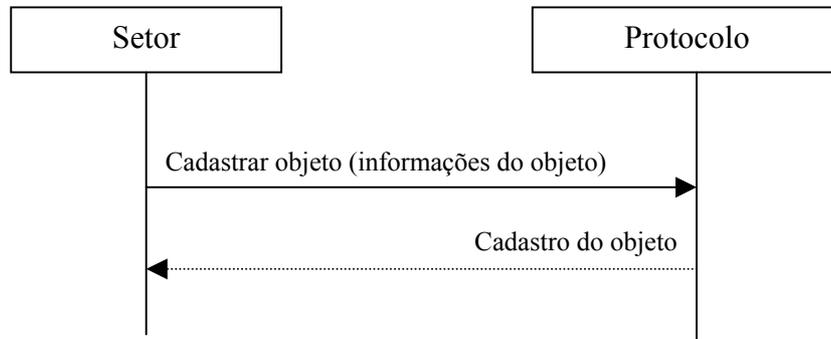
<b>Requisito</b>	<i>Nome:</i> <b>arquivar processo</b>
	<i>Descrição:</i> o arquivamento é o armazenamento do processo administrativo em algum local (setor); os processo podem ser arquivados, temporariamente, nos setores para depois serem encaminhados ao arquivo geral e, posteriormente, ao arquivo morto; o arquivamento em setores facilita o acesso a processo com assuntos em vigor (como contratos em andamento), entretanto, somente alguns setores podem manter processos arquivados
<b>Agente</b>	setores da autarquia
<b>Produto</b>	controle do andamento dos processos administrativos
<b>Recurso</b>	cadastro de setores e cadastro de autorizados cadastro de processos administrativos controle do andamento dos processos administrativos
<b>Anotação</b>	

<b>Requisito</b>	<i>Nome:</i> <b>reabrir processo</b>
	<i>Descrição:</i> um processo arquivado ou encerrado pode ser reaberto e voltar a tramitar normalmente pelos setores; isso se deve à necessidade de reavaliação do parecer final ou dar seqüência ao atendimento de uma solicitação
<b>Agente</b>	setores da autarquia
<b>Produto</b>	controle do andamento dos processos administrativos
<b>Recurso</b>	cadastro de setores e cadastro de autorizados cadastro de processos administrativos controle do andamento dos processos administrativos
<b>Anotação</b>	

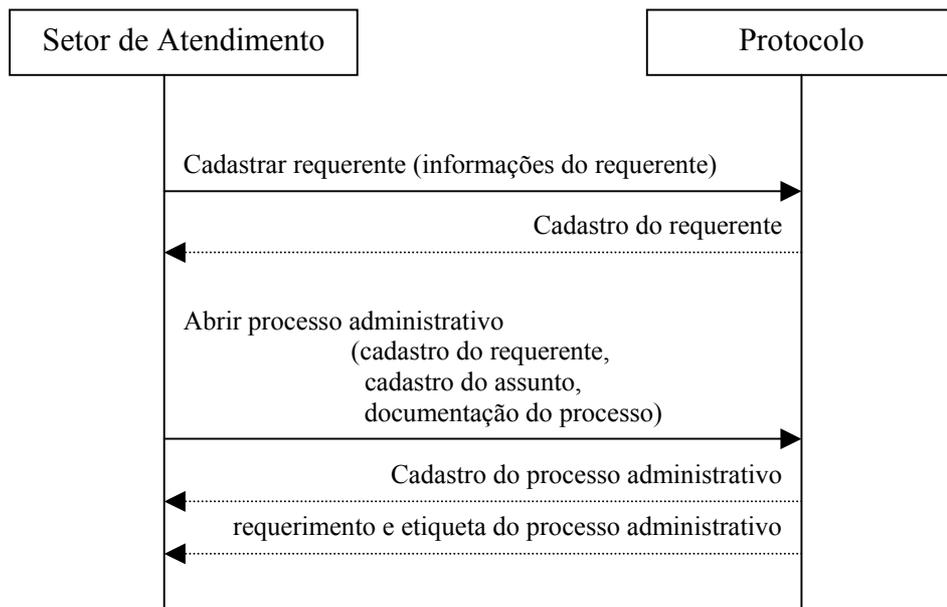
<b>Requisito</b>	<i>Nome:</i> <b>acompanhar processo</b>
	<i>Descrição:</i> uma necessidade essencial é o acompanhamento do andamento dos processos administrativos; por solicitação do requerente deve ser possível consultar as tramitações e a situação atual do processo; também, deve ser possível obter informações gerenciais sobre o andamento dos processos e, conseqüentemente, avaliar o atendimento às solicitações
<b>Agente</b>	setor da autarquia
<b>Produto</b>	controle do andamento dos processos administrativos
<b>Recurso</b>	cadastro de processos administrativos controle do andamento dos processos administrativos cadastro de requerentes cadastro de setores
<b>Anotação</b>	

<b>Requisito</b>	<i>Nome:</i> <b>consultar processo</b>
	<i>Descrição:</i> a consulta de um processo deve apresentar todas as informações relativas ao processo, selecionado por critério como identificação do processo, assunto, requerente, setor
<b>Agente</b>	setores da autarquia
<b>Produto</b>	controle do andamento dos processos administrativos
<b>Recurso</b>	cadastro de processos administrativos controle do andamento dos processos administrativos cadastro de requerentes cadastro de setores
<b>Anotação</b>	prever a qualidade de usabilidade da interface para futura disponibilidade de consulta para os usuários

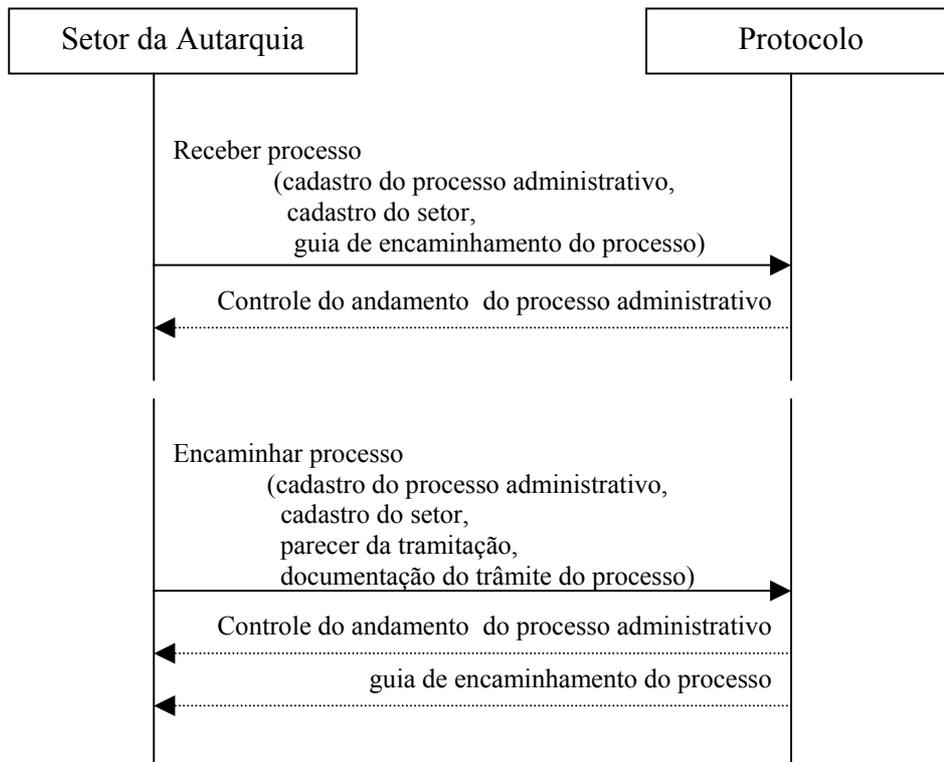
<b>Requisito</b>	<i>Nome:</i> <b>quantificar processos</b>
	<i>Descrição:</i> a quantificação dos processos administrativos deve possibilitar a extração de dados estatísticos para um acompanhamento gerencial das atividades do protocolo
<b>Agente</b>	setor de protocolo
<b>Produto</b>	controle do andamento dos processos administrativos
<b>Recurso</b>	cadastro de processos administrativos controle do andamento dos processos administrativos cadastro de requerentes cadastro de setores
<b>Anotação</b>	prever a qualidade de usabilidade da interface para consultas estatísticas através de gráficos

**Cenários:**

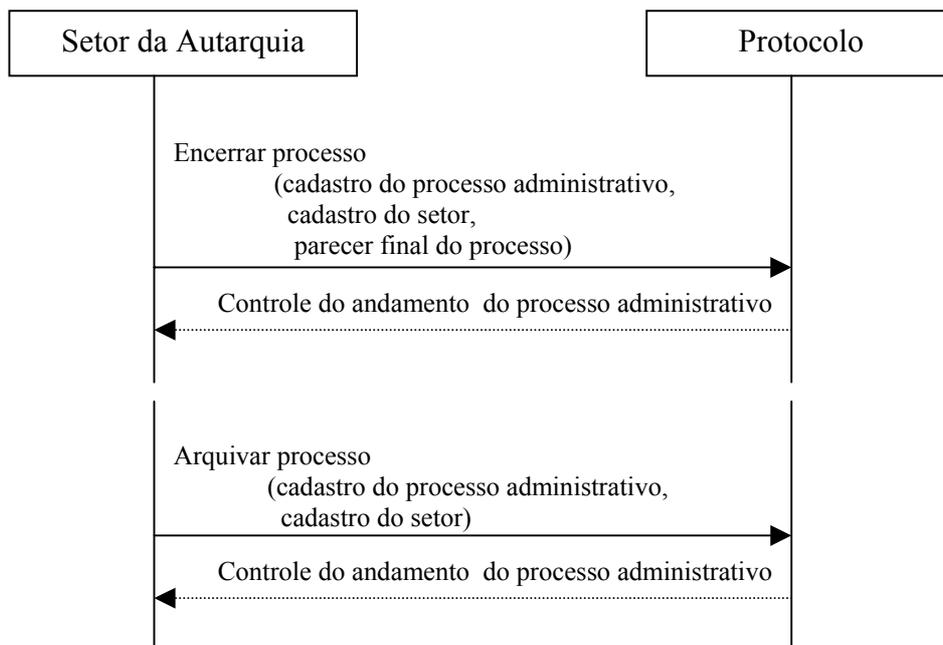
Cenário: operação genérica de cadastramento



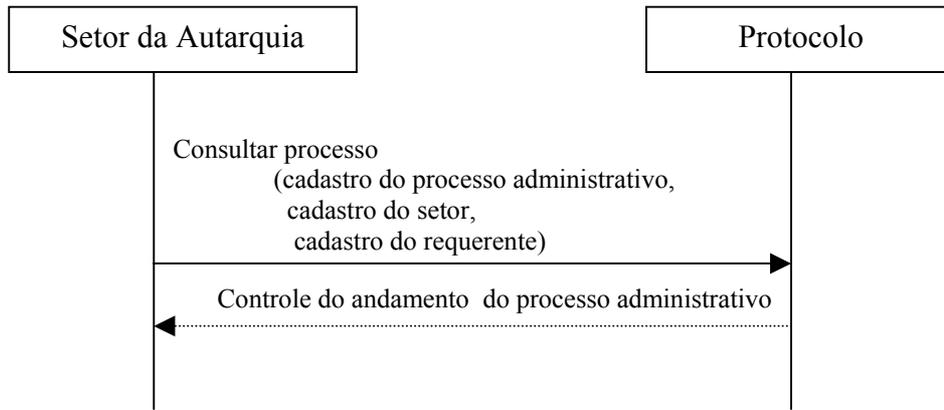
Cenário: abrir processo administrativo



Cenário: tramitar processo



Cenário: finalizar processo



Cenário: consultar processo

## Bibliografia

- [ALE 98] ALENCAR, F.M.R.; CASTRO, J.F.B. Formalização de Requisitos Organizacionais. In: WORKSHOP IBEROAMERICANO DE ENGENHARIA DE REQUISITOS E AMBIENTES DE SOFTWARE, 1., 1998, Torres. **Anais...** Porto Alegre: Instituto de Informática da UFRGS, 1998. v.1.
- [ALE 99] ALENCAR, F.M.R.; CASTRO, J.F.B. Integrating Early and Late-Phase Requirements: a Factory Case Study. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 13., 1999, Florianópolis. **Anais...** Florianópolis: UFSC, 1999.
- [ANT 96] ANTÓN, A.I. Goal-Based Requirements Analysis. In: INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING, 2., 1996, Colorado Springs, USA. **Proceedings...** [S.l.: s.n.], 1996.
- [BER 94] BERRY, D. The Requirement Iceberg and Various Icepicks Chipping at It. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 8., 1994, Curitiba. **Anais...** Curitiba: SBC, 1994.
- [BEJ 98] BERTOLIN J.C.G. Uma Análise de Técnicas da Psicologia para a Elicitação de Requisitos de Software. In: SEMANA ACADÊMICA PPGC/UFRGS, 3., 1998, Porto Alegre. **Anais...** Porto Alegre: PPGC da UFRGS, 1998.
- [BOE 81] BOEHM, B.W. **Software Engineering Economics**. Englewoods Cliffs, USA: Prentice Hall International, 1981.
- [BOO 94] BOOCH, G. **Object-Oriented Analysis and Design with Applications**. 2nd ed. Santa Clara, USA: Addison-Wesley, 1994.
- [BOO 2000] BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. Rio de Janeiro: Campus, 2000.
- [BRO 87] BROOKS, F.P. No Silver Bullet: Essence and Accidents of Software Engineering. **Computer**, New York, v.20, n.4, p.10-19, Apr.1987.
- [CAL 99] CALDAS JÚNIOR, J.; MASIERO, P.C. Obtenção de Modelo de Análise OO a partir de um Modelo de Requisitos Baseado em Cenários. In: JORNADAS IBEROAMERICANAS DE INGENIERIA DE REQUISITOS Y AMBIENTES DE SOFTWARE, 2., 1999, Alajuela, Costa Rica. **Memórias...** Cartago: CIT, 1999.
- [CAR 93] CARMEL, E.; WHITAKER, R.D.; GEORGE, F.J. Participatory Design and Joint Application Design: a Transatlantic Comparison. **Communications of the ACM**, New York, v.34, n.4, p.40-46, June 1993.
- [CAJ 98] CARROLL, J.M. et al. Requirements Development in Scenario-Based Design. **IEEE Transactions on Software Engineering**, New York, v.24, n.12, p.1156-1170, Dec.1998.
- [CAS 95] CASTRO J.F.B. Introdução a Engenharia de Requisitos. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA DO CONGRESSO DA

- SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 14., 1995, Canela. **Anais...** Porto Alegre: Instituto de Informática da UFRGS, 1995.
- [COL 96] COLEMAN, D. et al. **Desenvolvimento Orientado a Objetos: o Método Fusion**. Rio de Janeiro: Campus, 1996.
- [DEB 99] DE BORTOLI, L.A. **Um Método de Trabalho para Auxiliar a Definição de Requisitos**. 1999. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-Graduação em Computação, UFRGS.
- [FOW 2000] FOWLER, M.; SCOTT, K. **UML Distilled: a brief guide to the standard object modeling language**. 2nd ed. Massachusetts, USA: Addison-Wesley, 2000.
- [GAL 2000] GALIMBERTI, M.F.; BUSSMANN, J.E.C.; ROCCO, G.E. **Proposta de Adaptação do Modelos do Método Fusion à UML**. Caxias do Sul: Departamento de Informática, Universidade de Caxias do Sul, 2000. Documento de Projeto.
- [GIL 97] GILB, T. Towards the Engineering of Requirements. **Requirements Engineering Journal**, London, v.2, n.3, p.165-169, 1997.
- [GOG 94] GOGUEN, J. Requirements Engineering as the Reconciliation of Social and Technical Issues. In: JIROTKA, M.; GAGUEN, J.A. **Requirements Engineering: Social and Technical Issues**. San Diego, USA: Academic Press, 1994.
- [GON 98] GONZALES, M.; POHLMANN FILHO, O. Plano de Cenários como Expansão para a Metodologia OMT. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 20., 1998, Kyoto, Japan. **Proceedings...** Los Alamitos: IEEE Computer Society, 1998.
- [GUN 2000] GUNTER, C.A. et al. A Reference Model for Requirements and Specifications. **IEEE Software**, New York, v.18, n.3, p.37-43, May/June 2000.
- [HAU 98] HAUMER, P.; POHL, K.; WEIDENHAUPT, K. Requirements Elicitation and Validation with Real World Scenes. **IEEE Transactions on Software Engineering**, New York, v.24, n.12, p.1036-1054, Dec.1998.
- [HER 96] HERLEA, D.E. **Users' Involvement in the Requirements Engineering Process**. Calgary, Canada: Knowledge Science Institute of University of Calgary, 1996.
- [IEEE 97] IEEE. **IEEE Software Standards Collection**. Los Alamitos, 1997.
- [JAI 95] JACKSON, M. **Software Requirements and Specifications: a lexicon of practice, principles and prejudices**. Santa Clara, USA: Addison-Wesley, 1995.
- [JAC 92] JACOBSON, I. et al. **Object-Oriented Software Engineering: a Use Case driven approach**. Santa Clara, USA: Addison-Wesley, 1992.
- [JAC 99] JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. The Unified Process. **IEEE Software**, New York, v.18, n.3, p.96-102, May/June1999.

- [KAP 95] KAPTELININ, V.; KUUTTI, K.; BANNON, L. Activity Theory: Basic Concepts and Applications. In: EAST-WEST HUMAN COMPUTER INTERACTION CONFERENCE, 5., 1995, Moscow, Russia. **Proceedings...** Berlin: Springer-Verlag, 1995.
- [KAP 97] KAPTELININ, V.; NARDI, B.A. Activity Theory: Basic Concepts and Applications. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1997, Atlanta, USA. **Proceedings...** New York: ACM, 1997.
- [KEN 93] KENSING, F.; MADSEN, A.M. Participatory Design: Structure in the Toolbox. **Communications of the ACM**, New York, v.36, n.4, p.78-85, June 1993.
- [LAM 98] LAMSWEERDE, A.; WILLEMET, L. Inferring Declarative Requirements Specifications from Operational Scenarios. **IEEE Transactions on Software Engineering**, New York, v.24, n.12, p.1089-1114, Dec.1998.
- [LAM 98b] LAMSWEERDE, A.; DARIMONT, R.; LETIER E. Managing Conflicts in Goal-Driven Requirements Engineering. **IEEE Transactions on Software Engineering**, New York, v.24, n.11, p.908-926, Nov.1998.
- [LAP 88] LAPLATINE F. **Aprender Antropologia**. São Paulo: Brasiliense, 1988.
- [LEE 98] LEE, W.J.; CHA, S.D.; KWON, Y.R. Integration and Analysis of Use Cases Using Modular Petri Nets in Requirements Engineering. **IEEE Transactions on Software Engineering**, New York, v.24, n.12, p.1115-1130, Dec.1998.
- [LEE 99] LEE, J.; XUE, N.L. Analyzing User Requirements by Use Cases: a Goal-Driven Approach. **IEEE Software**, New York, v.18, n.4, p.92-101, July/Aug.1999.
- [LEF 2000] LEFFINGWELL, D.; WIDRIG, D. **Managing Software Requirements: a unified approach**. Massachusetts, USA: Addison-Wesley, 2000.
- [LEI 94] LEITE, J.C.S.P. **Engenharia de Requisitos**: notas de aula. Rio de Janeiro: PUC, 1994.
- [LOU 95] LOUCOPOULOS, P.; KARAKOSTAS, V. **System Requirements Engineering**. New York, USA: McGraw-Hill, 1995.
- [MAC 99] MACEDO, N.A.M.; LEITE, J.C.S.P. Integrando Requisitos Não Funcionais aos Requisitos Baseados em Ações Concretas. In: JORNADAS IBEROAMERICANAS DE INGENIERIA DE REQUISITOS Y AMBIENTES DE SOFTWARE, 2., 1999, Alajuela, Costa Rica. **Memórias...** Cartago: CIT, 1999.
- [MAR 99] MARTINS, L.E.G.; DALTRINI, B.M. Utilização dos Preceitos da Teoria da Atividade na Elicitação dos Requisitos de Software. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 13., 1999, Florianópolis. **Anais...** Florianópolis: UFSC, 1999.
- [MAT 98] MATSURA, F.; SHINGO, T.; NAKAKOJI, K. A Study on Communication During Requirement Elicitation Processes. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 20., 1998, Kyoto, Japan. **Proceedings...** Los Alamitos: IEEE Computer Society, 1998.

- [NAR 96] NARDI, B.A. **Context and Consciousness - Activity Theory and Human-Computer Interaction**. Massachusetts, USA: MIT Press, 1996.
- [NIS 97] NISSEN, H.W.; JARKE, M. Requirements Engineering Repositories: Formal Support for Informal Teamwork Methods. In: INTERNATIONAL WORKSHOP ON REQUIREMENTS TARGETING SOFTWARE AND SYSTEMS ENGINEERING, 1997, Bernried, Germany. **Requirements Targeting Software and System Engineering**: proceedings. Berlin: Springer-Verlag, 1998. (Lecture Notes in Computer Science, v.1526).
- [NUS 96] NUSEIBEH, B. Conflicting Requirements: When the Customer is not Always Right. **Requirements Engineering Journal**, London, v.1, n.1, p.70-71, 1996.
- [OLI 96] OLIVEIRA, K.M. **Aquisição de Conhecimento**: velha fórmula, nova aplicação. 1996. Exame de Qualificação (Doutorado em Ciência da Computação) - COOPE, Universidade Federal do Rio de Janeiro.
- [PIN 98] PINHEIRO, F.A.C. **Da Importância dos Modelos de Desenvolvimento de Software para a Engenharia de Requisitos**. Brasília: Departamento de Ciência da Computação, Universidade de Brasília, 1998. Relatório Técnico.
- [POT 94] POTTS, C.; TAKAHASHI, K.; ANTÓN, A.I. Inquiry-Based Requirements Analysis. **IEEE Software**, New York, v.11, n.2, p.21-40, Feb.1994.
- [PRE 95] PRESSMAN R.S. **Engenharia de Software**. São Paulo: Makron Books do Brasil, 1995.
- [ROL 98] ROLLAND, C.; SOUVEYET, C.; ACHOUR, C.B. Guiding Goal Modeling Using Scenarios. **IEEE Transactions on Software Engineering**, New York, v.24, n.12, p.1055-1071, Dec.1998.
- [ROS 99] ROSENBERG, D. **Use Case Driven Object Modeling with UML**: a practical approach. Massachusetts, USA: Addison-Wesley, 1999.
- [RUM 94] RUMBAUGH, J. et al. **Modelagem e Projetos Baseados em Objetos**. Rio de Janeiro: Campus, 1994.
- [SAW 99] SAWYER, P.; SOMMERVILLE, I.; VILLER, S. Capturing the Benefits of Requirements Engineering. **IEEE Software**, New York, v.18, n.2, p.78-85, Mar./Apr.1999.
- [SCH 98] SCHNEIDER, G.; WINTERS, J.P. **Applying Use Cases**: a practical guide. Massachusetts, USA: Addison-Wesley, 1998.
- [SIM 97] SIMONSEN, J.; KENSING, F. Using Ethnography in Contextual Design. **Communications of the ACM**, New York, v.40, n.7, p.82-88, July 1997.
- [SOM 93] SOMMERVILLE, I. et al. Ethnography into the Requirement Process. In: INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING, 1., 1993, San Diego, USA. **Proceedings...** San Diego: IEEE Computer Society, 1993.
- [SOM 97] SOMMERVILLE, I.; SAWER, P. **Requirements Engineering**: a good practice guide. Chichester, England: John Wiley & Sons, 1997.

- [SUT 98] SUTCLIFFE, A.G.; MAIDEN, N. The Domain Theory for Requirements Engineering. **IEEE Transactions on Software Engineering**, New York, v.24, n.3, p.174-196, Mar.1998.
- [SUT 98b] SUTCLIFFE, A.G. et al. Supporting Scenario-Based Requirements Engineering. **IEEE Transactions on Software Engineering**, New York, v.24, n.12, Dec.1998.
- [TAN 97] TANENBAUM A.S. **Redes de Computadores**. Rio de Janeiro: Campus, 1997.
- [WEI 98] WEIDENHAUPT, K. et al. Scenarios in System Development: Current Practice. **IEEE Software**, New York, v.17, n.2, p.34-45, Mar./Apr.1998.
- [WIE 96] WIERING, R.J. **Requirements Engineering: frameworks for understanding**. New York, USA: John Wiley & Sons, 1996.
- [WIE 99] WIEGERS, K.E. **Software Requirements**. Washington, USA: Microsoft Press, 1999.
- [YUE 98] YU, E.S.K. Why Goal-Oriented Requirements Engineering. In: INTERNATIONAL WORKSHOP OF REQUIREMENTS ENGINEERING: FOUNDATION FOR SOFTWARE QUALITY, 4., 1998, Pisa, Italy. **Proceedings...** Namur: Universitaires de Namur, 1998.