



Universidade Metodista de Piracicaba
Faculdade de Ciências Exatas e da Natureza
Mestrado em Ciência da Computação

**APLICAÇÃO DO MODELO I* EM UM PROCESSO DE ANÁLISE DE
REQUISITOS ORIENTADO A METAS**

CRIVANIL OLIVEIRA DOS SANTOS

ORIENTADOR: PROF. DR. LUIZ EDUARDO GALVÃO MARTINS

PIRACICABA, SP
2006



Universidade Metodista de Piracicaba
Faculdade de Ciências Exatas e da Natureza
Mestrado em Ciência da Computação

**APLICAÇÃO DO MODELO I* EM UM PROCESSO DE ANÁLISE DE
REQUISITOS ORIENTADO A METAS**

CRIVANIL OLIVEIRA DOS SANTOS

ORIENTADOR: PROF. DR. LUIZ EDUARDO GALVÃO MARTINS

Dissertação apresentada ao Mestrado em Ciência da Computação, da Faculdade de Ciências Exatas e da Natureza, da Universidade Metodista de Piracicaba – UNIMEP, como requisito para obtenção do Título de Mestre em Ciência da Computação.

PIRACICABA, SP
2006

Santos, Crivanil Oliveira dos
Aplicação do Modelo i^* em um Processo de Análise de Requisitos Orientado a Metas. Piracicaba, 2006.

Orientador: Prof. Dr. Luiz Eduardo Galvão Martins.

Dissertação (Mestrado) - Programa de Pós-Graduação em Ciência da Computação – Faculdade de Ciências Exatas e da Natureza - Universidade Metodista de Piracicaba.

1- Análise de Requisitos. 2- Elicitação de Requisitos. 3- Engenharia de Requisitos.

Ao

*Meu esposo Marcus Aurélio pelo seu amor,
pelo apoio que tem sido meu melhor incentivo, por tornar possível a minha
dedicação ao trabalho e pela paciência, capaz de compreender as muitas
horas de ausência em família.*

*Às minhas filhas Liv e Eloah, razão de meu esforço para evoluir como pessoa e
profissional.*

*À minha mãe Maria do Carmo e ao meu pai de
coração Antonio que possibilitaram a minha trajetória, cuidando das minhas
filhas.*

AGRADECIMENTOS

A Deus, sua onipotência e onipresença, e por permitir o aprendizado, a minha evolução acadêmica e por ter incluído em minha vida pessoas que me apoiaram e me incentivaram.

Ao meu orientador professor Luiz Eduardo Galvão Martins, que desde o primeiro contato teve a capacidade de entender o que eu poderia fazer, os incentivos constantes, a confiança e o respeito com minhas idéias, dispensado no desenvolvimento deste trabalho.

Aos Docentes do Mestrado em Ciência da Computação da FACEN/UNIMEP, os ensinamentos.

Aos familiares e amigos, o apoio e o incentivo, que, direta ou indiretamente a mim dispensaram durante o decorrer do presente trabalho.

“O ÚNICO HOMEM QUE NUNCA COMETE ERROS É AQUELE QUE NUNCA FAZ COISA
ALGUMA. NÃO TENHA MEDO DE ERRAR, POIS VOCÊ APRENDERÁ A NÃO COMETER DUAS
VEZES O MESMO ERRO”

ROOSEVELT

RESUMO

Este trabalho apresenta um guia de orientação para aplicação da técnica *i** durante a análise de requisitos, de forma a identificar atores e atividades, estabelecendo a rede de ligações nos processos entre estes e seus relacionamentos, considerando desdobramentos e refinamentos do processo de elicitación de requisitos. O referencial teórico deste trabalho está centrado na Engenharia de Requisitos e na abordagem análise de requisitos orientada a metas. Destaca-se a abordagem orientada a metas, amplamente utilizada em diferentes partes da Engenharia de Requisitos, que se constitui numa técnica eficaz para a definição dos objetivos de um sistema e para a identificação dos requisitos. Nesta abordagem elegeu-se a técnica de modelagem *i** (*i-star*) que auxilia no entendimento do ambiente organizacional e suas metas, pois está focada no relacionamento estratégico entre os atores envolvidos. Como exemplo de aplicabilidade da metodologia proposta desenvolveu-se um estudo de caso, a partir de um sistema para gerenciamento e organização de simpósios. No final do trabalho, com os resultados obtidos no estudo de caso, identificou-se que o processo de Engenharia de Requisitos envolve desenvolvimento, descobrimento e o embasamento do documento de requisitos, enquanto fatores fundamentais para uma proposta eficiente de construção de sistemas e que são vinculados ao entendimento dos requisitos. Identificar usuários, suas necessidades e como supri-las são atributos a serem desenvolvidos pelo engenheiro de requisitos - um mediador entre usuário e suas necessidades.

PALAVRAS-CHAVE: Técnica *i**, Requisitos, Elicitación de Requisitos, Engenharia de Requisitos, Análise de Requisitos Orientada a Metas.

ABSTRACT

This dissertation presents an orientation guide for application of the i* technique during the requirements analysis, in order to identify actors and activities, establishing the net of linking in the processes, among them and their relationships, considering unfolding and refinements of the process of requirements elicitation. The theoretical referential of this dissertation is centered in the Requirements Engineering and in the approach of requirements analysis guided to goals. We detach the approach guided to goals, widely used in different parts of Requirements Engineering that constitutes an efficient technique for the definition of the objectives of a system and for the identification of the requirements. In this approach we elected the modeling technique i* (i-star) that supports in the comprehension of the organizational environment and its goals, therefore is focused in the strategical relationship among the involved actors. As example of applicability of the methodology proposal we develop a case study starting from a system for management and organization of symposiums. At the end of the work, with the results reached in the case study, we identify that the ER process involves development, discovery and the basement of the document of requirements, while basic factors for an efficient construction of systems proposal and, that they are tied with the comprehension of the requirements. Identify users, its necessities and how to supply them are attributes to be developed by the requirements engineer - a mediator between users and their its needs.

KEYWORDS: Technique i*, Requirements, Requirements Elicitation, Requirements Engineering, Analysis of Requirements Guided to the Goals.

SUMÁRIO

1. INTRODUÇÃO.....	01
2. ENGENHARIA DE REQUISITOS	05
2.1. ELICITAÇÃO DE REQUISITOS	10
2.1.1. TÉCNICAS DE ELICITAÇÃO DE REQUISITOS	12
2.1.1.1. OBSERVAÇÃO.....	12
2.1.1.2. QUESTIONÁRIO.....	13
2.1.1.3. ENTREVISTA.....	14
2.1.1.4. <i>BRAINSTORMING</i>	16
2.1.1.5. JAD	16
2.1.1.6. CENÁRIOS.....	17
2.1.1.7. CASOS DE USO	20
2.1.1.8. TEORIA DA ATIVIDADE	21
2.2. ANÁLISE DE REQUISITOS	25
2.3. MODELAGEM OU ESPECIFICAÇÃO DE REQUISITOS	28
2.4. VALIDAÇÃO DE REQUISITOS	30
2.5. GERENCIAMENTO DE REQUISITOS	32
3. Análise de Requisitos Orientada a Metas	36
3.1. CONCEITOS SOBRE METAS	36
3.2. TÉCNICAS E MÉTODOS DE SUPORTE À ORIENTAÇÃO A METAS	41
3.2.1. ORIENTAÇÃO A METAS UTILIZANDO KAOS	41
3.2.2. ABORDAGEM <i>CREWS-L'ECRITOIRE</i>	43
3.2.3. MÉTODO DE ANÁLISE DE REQUISITOS BASEADO EM METAS - GBRAM	45
3.2.4. CASOS DE USO DIRIGIDO A METAS – MÉTODO GDUC	47
4. Técnica de modelagem i*	51
4.1. MODELO DE DEPENDÊNCIA ESTRATÉGICA – SD	52
4.2. RELACIONAMENTOS COMPLEXOS ENTRE ATORES	55
4.3. MODELO ESTRATÉGICO DETALHADO – SR	55

5. contextualização do estudo de caso.....	58
5.1. MODELO DE DEPENDÊNCIA ESTRATÉGICA – SD	59
5.2. MODELO ESTRATÉGICO DETALHADO – SR.....	61
6. contribuições para utilização da técnica i*	70
6.1. FLUXOGRAMA DE APLICAÇÃO DA TÉCNICA I*	70
6.1.1. IDENTIFICAR ATORES	72
6.1.2. IDENTIFICAR OS RELACIONAMENTOS DE DEPENDÊNCIA	73
6.1.3. CRIAR MATRIZ DE RELACIONAMENTO ENTRE OS ATORES, ALOCANDO AS DEPENDÊNCIAS IDENTIFICADAS.....	73
6.1.4. CRIAR UM MODELO DE DEPENDÊNCIA ESTRATÉGICA (SD) PRELIMINAR	75
6.1.5. CRIAR MODELOS DETALHADOS (SR) POR ATOR (DECOMPOSIÇÃO DE TAREFAS).....	75
6.1.6. JUNTAR OS MODELOS SR ISOLADOS NUM MODELO ESTRATÉGICO DETALHADO.....	76
7. Conclusão.....	78
Referências Bibliográficas	81

LISTA DE FIGURAS

FIGURA 1 – COMPONENTES DA ELICITAÇÃO DE REQUISITOS	08
FIGURA 2 - EXEMPLO DE CASO DE USO.....	21
FIGURA 3 – ETAPAS DA METODOLOGIA DE ELICITAÇÃO DE REQUISITOS	23
FIGURA 4 - MODELO ESPIRAL DE ELICITAÇÃO, ANÁLISE E NEGOCIAÇÃO	27
FIGURA 5 - MODELO ESPIRAL DO PROCESSO DE ENGENHARIA DE REQUISITOS	29
FIGURA 6 – VALIDAÇÃO DE REQUISITOS.....	31
FIGURA 7 – DECOMPOSIÇÃO DE CADA META FUNCIONAL EM UMA HIERARQUIA DE RELAÇÕES E (AND) / OU (OR)	42
FIGURA 8 – MODELO DE PROCESSO DA ABORDAGEM <i>CREWS-L'ECRITOIRE</i>	45
FIGURA 9 – ATIVIDADES DO MÉTODO GBRAM.....	46
FIGURA 10 – NOTAÇÃO GRÁFICA DO RELACIONAMENTO DE DEPENDÊNCIA	53
FIGURA 11 – SIMBOLOGIA PARA O MODELO DE DEPENDÊNCIA	54
FIGURA 12 – MEDIADORES DAS RELAÇÕES ENTRE ATORES.....	54
FIGURA 13 – MODELO DE DEPENDÊNCIA ESTRATÉGICA PARA SISTEMA DE SIMPÓSIO	61
FIGURA 14 – MODELO ESTRATÉGICO DETALHADO PARA O ATOR “AVALIADOR”	62
FIGURA 15 – MODELO ESTRATÉGICO DETALHADO PARA O ATOR “PARTICIPANTE AUTOR”.....	63
FIGURA 16 – MODELO ESTRATÉGICO DETALHADO PARA O ATOR “PARTICIPANTE OUVINTE”.....	64
FIGURA 17 – MODELO ESTRATÉGICO DETALHADO DO ATOR “ <i>SOFTWARE</i> DO SIMPÓSIO”	65
FIGURA 18 – MODELO ESTRATÉGICO DETALHADO DO ATOR “COORDENADOR”	66
FIGURA 19 – MODELO ESTRATÉGICO DETALHADO DO ATOR “BANCO”	66
FIGURA 20 – MODELO ESTRATÉGICO DETALHADO DO SISTEMA DE SIMPÓSIO.....	69
FIGURA 21 – FLUXOGRAMA DAS ETAPAS PARA APLICAÇÃO DA TÉCNICA I*	71
FIGURA 22 – MEDIADORES NAS RELAÇÕES ENTRE ATORES	73

LISTA DE QUADROS

QUADRO 1 – PALAVRAS CHAVE PARA EXPRESSAR <i>SOFTGOAL</i>	56
QUADRO 2 – SIMBOLOGIA DAS CONTRIBUIÇÕES POSITIVAS E NEGATIVAS PARA <i>SOFTGOAL</i>	56
QUADRO 3 – MATRIZ DAS RELAÇÕES DE DEPENDÊNCIA PARA O SISTEMA DE SIMPÓSIO	60
QUADRO 4 – EXEMPLO DE MATRIZ PARA RELAÇÕES DE DEPENDÊNCIA.....	75

1. INTRODUÇÃO

A Engenharia de Requisitos (ER) é estabelecida enquanto um processo; por isso denota movimentação de idéias e de elaborações com o fim de desenvolvimento e evolução de um sistema. Em tal complexidade, a ER se beneficia da diversificação de técnicas, métodos e recursos, especialmente aplicados à descoberta e compreensão de requisitos que definem um sistema a ser construído.

O processo de ER envolve inúmeras repetições de atividades que, conseqüentemente, resultam em versões sucessivas de documentos para a relação e descrição dos requisitos que, ainda, se diferenciam pelo nível distinto de detalhamento das informações elaboradas.

Uma relação final de requisitos é um documento decorrente do processo continuado de descobrimentos e (re) elaborações de requisitos, reunindo necessidades e propósitos demandados pelos usuários, que especifica o que é requerido a um sistema fazer.

Dentre as técnicas aplicáveis à ER para a definição dos objetivos de um sistema e para a identificação dos requisitos associados, a abordagem orientada a metas é considerada muito eficaz (ÁNTON, 1996). A abordagem orientada a metas é amplamente utilizada em diferentes partes da ER e se justifica pelas razões de um sistema determinar suas diretrizes de desenvolvimento, para a organização na qual é requerido.

Reconhecendo a importância das metas no processo de ER, diversos pesquisadores passaram aos estudos sobre a modelagem de metas, especificação de metas e raciocínio baseado em metas para múltiplos propósitos, dentre eles a elaboração de requisitos e a verificação ou o gerenciamento de conflitos (LAMSWEEERDE, 2001).

Diante da complexidade e importância do tema ER, neste trabalho houve o esforço para facilitar seu entendimento e sua aplicação

prática. O objetivo primordial, e básico, do presente trabalho é apresentar um guia de orientação para a aplicação da técnica de modelagem i^* (*i-star*).

A técnica de modelagem i^* subsidia a orientação a metas durante a análise de requisitos, para garantir clareza na atuação e identificação de atores, estabelecendo a rede de ligações, processos e relacionamentos, além de considerar, complementarmente, os desdobramentos e refinamentos do processo. Essa técnica torna-se recurso fundamental para o entendimento do ambiente organizacional e de suas metas, tendo como foco o relacionamento estratégico entre todos atores envolvidos.

Para que fosse possível atingir o objetivo proposto adotou-se, enquanto metodologia, a implementação de um estudo de caso. Nesse estudo, a ER mediada através da técnica i^* para a elicitación de requisitos, tem a intenção de explicitar pontos críticos e advertências, dificuldades e facilidades encontradas durante esse exercício, aos engenheiros de requisitos. Especialmente, a intenção do presente trabalho é destacar situações referenciadas no âmbito teórico, vivenciada no estudo de caso implementado, que se destacam a partir da discussão teórico-metodológica de um sistema para a organização de um simpósio, abordando suas atividades e integrações no contexto da ER.

O desenvolvimento do presente trabalho dá-se a partir de uma revisão teórica, desenvolvida no capítulo 2, onde se destaca a conceituação do processo de definição dos requisitos compreendendo também os seus subprocessos aplicáveis à ER. Estabelecendo um *link*, apresentamos as algumas técnicas empregadas na elicitación de requisitos. Com tal elaboração, objetiva-se desenvolver alternativas metodológicas para os engenheiros de requisitos ultrapassarem os limites impostos ao desenvolvimento de sistemas. As técnicas de elicitación trazem, adicionalmente, recomendações para a comunicação entre as pessoas de modo a obter informações mais precisas e pertinentes ao desenvolvimento de sistemas.

O capítulo 3 traz um panorama sobre o desenvolvimento da análise de requisitos orientada a metas. O desenvolvimento teórico desse

capítulo, em especial, é pertinente à discussão deste trabalho à medida que se optou por desenvolver um estudo de caso que se apoiou na análise de requisitos orientada a metas. Nesse contexto, pode-se dizer que as metas têm sido consideradas componentes essenciais que estão envolvidas no processo de ER e são conduzidas para dar apoio à elaboração dos requisitos, sendo que a especificação é completa, se todas as metas relatadas são obtidas pela especificação.

Aborda-se no capítulo 4 a Técnica de Modelagem *i**. Essa técnica é um recurso fundamental para o entendimento do ambiente organizacional e suas metas e tem como foco o relacionamento estratégico entre os atores envolvidos no sistema. A técnica oferece dois modelos o Modelo de Dependência Estratégico (SD) e o Modelo Estratégico Detalhado (SR) (YU, 2000).

O estudo de caso é apresentado no capítulo 5, onde se delineou a situação e, a partir da mesma, desenvolveu-se os procedimentos da ER, aplicando a modelagem *i** com a identificação de atores, requisitos e mediadores.

O capítulo 6 é, de fato, o cerne da proposta apresentada, onde se apresenta a contribuição desta dissertação, enquanto um guia para a aplicação da técnica de modelagem, na qual apoiou-se na ferramenta de implementação *i** para construir passos orientadores aos principiantes na técnica. Também, a partir dessa elaboração, foi possível destacar e reforçar pontos já elencados na explanação teórica que, por meio desse guia são explicitados por simples dedução.

Finalmente, apresenta-se o capítulo 7, onde é tratada a dimensão do trabalho de pesquisa relatado, as contribuições à guisa de conclusões. A discussão que se estabelece encerra o presente trabalho com as contribuições e abstrações que se deram com o desenrolar dos meandros acadêmicos que, felizmente, encontram-se em plena produção e, nesse sentido, as conclusões apresentadas tornam-se integrantes de um corpo de

solidificação teórico-metodológico, que pode gerar uma ligação do trabalho atual com o desenvolvimento de trabalhos futuros.

2. ENGENHARIA DE REQUISITOS

A necessidade de uma nova área de pesquisa no âmbito das organizações, aplicável à problemática de como definir a melhor forma de administrar a informação, e a influência das diferentes pessoas nesse contexto para o desenvolvimento de *software*, levaram ao surgimento da Engenharia de Requisitos (ER).

O processo de ER implica que técnicas sistemáticas sejam utilizadas para a descoberta dos requisitos do *software*, que retrate as necessidades dos usuários e a posterior implementação desses requisitos, sendo elemento fundamental para o sucesso na construção do *software* (NUSEIBEH, 2000). Assim, pode-se dizer que a ER é uma abordagem sistemática de obtenção de conhecimento, envolvendo as atividades de descobrimento, documentação e manutenção de um conjunto de requisitos, com vistas a minimizar os problemas diante da complexidade do *software* (INSFRÁN, 2002).

O termo em inglês é *Requirements Engineering*, traduzido para o português como Engenharia de Requisitos¹, e estuda o processo de definição dos requisitos aos quais o *software* deverá atender. A seguir são apresentadas algumas definições para ER. O processo de ER, para Kotonya (1998), é um conjunto estruturado de atividades para extrair requisitos, validá-los e mantê-los.

Segundo Zave (1995), a ER é a área da engenharia de *software*, preocupada com os objetivos do mundo real, funções e condições sobre o *software*. Também diz respeito à influência desses fatores para especificações precisas de comportamento de *software*, para sua evolução com o tempo e cruzamento de compartilhamento de dados e funções.

Para Macaulay (1996), é um processo sistemático de desenvolvimento de requisitos, por meio de um processo iterativo e cooperativo

¹ A área surgiu em 1993 com a realização do *I International Symposium on Requirements Engineering*.

de análise do problema de documentação das observações resultantes, de diversos formatos de representação e da certificação do entendimento obtido.

Ainda para Kotonya (1998), é um termo relativamente novo que foi criado para cobrir todas as atividades envolvidas em descobrimento, documentação e manutenção de um conjunto de requisitos, para um *software* baseado em computador. A utilização do termo engenharia implica que técnicas sistemáticas e passíveis de repetição podem ser usadas para assegurar que os requisitos do *software* sejam completos, consistentes, relevantes, etc.

As atividades da ER que estabelecem o processo de definição de requisitos, no qual o que deve ser feito é elicitado, analisado e modelado, resultam em diferentes pontos de vista e utiliza uma combinação de métodos, ferramentas e pessoal (LEITE, 2000). O resultado é um modelo, onde o documento de requisitos é produzido a partir do Universo de Informação (Udel).

Udel é o conjunto de todas as informações e todas as pessoas envolvidas com o *software* a ser desenvolvido, estabelecido por um grupo de objetivos definidos por quem solicitou o *software*. Nesse sentido, ao longo do processo de desenvolvimento, os engenheiros de *software* deverão entender o ambiente no qual o *software* irá funcionar e, dentre outros aspectos, estabelecer uma linha objetiva diante das necessidades do usuário.

Consolidando as idéias, a ER auxilia o papel de prover o engenheiro de *software*, de métodos, técnicas e ferramentas que auxiliem o processo de entendimento e documentação dos requisitos. Essas informações, não estando documentadas claramente, dificultam a realização das atividades, gerando uma inadequada definição dos requisitos (PRESSMAN, 2001).

As atividades de formulação dos requisitos são estabelecidas de forma iterativa. Começa com a atividade de elicitação dos requisitos a partir do Udel e, em seguida, com as informações elicidadas, o modelo é construído e as informações serão validadas junto aos usuários. Com o modelo validado e as informações sistematizadas, o engenheiro de *software* pode definir os

requisitos para a construção do *software*. Assim, finaliza-se o processo de ER, com a construção do documento de requisitos. Mesmo após a validação pode ser necessário retornar ao Udel, ou a outras atividades para elicitare informações novas, esclarecer e identificar outras informações, refazer o modelo ou até mesmo realizar a reengenharia dos processos, até que seja possível validar o modelo.

Poucas organizações têm um processo de ER padronizado e definido explicitamente (KOTONYA, 1998). O funcionamento do processo ocorre em forma de espiral, pois é iterativo e envolve repetição das atividades na geração de versões do documento de requisitos. Dependendo da organização, o documento de requisitos pode ter diferentes denominações e ser de vários tipos ou níveis de detalhamento. Pode ter ainda vários papéis, várias formas e conteúdo. É uma declaração formal de requisitos de clientes, usuários e desenvolvedores de *software* (KOTONYA, 1998) (BERRY, 1998).

Qualquer *software* que envolva intervenção humana possui características de ser volátil, não previsível e complexo. Dessa forma, uma das grandes dificuldades iniciais é estabelecer os requisitos. O documento de requisitos é o produto final do processo de descobrimento de requisitos e reúne necessidades e propósitos demandados pelos usuários. É uma especificação do que é requerido a um *software* fazer (MACAULAY, 1996).

O ciclo da informação identifica as necessidades e viabilidade para a construção do *software*, que permitirá que as etapas da ER especifiquem os requisitos através do documento de requisitos. Começa com a definição das necessidades do *software*, tendo como produto um relatório de necessidades. Passa pelo estudo de viabilidade a partir das informações, gerando também um relatório de viabilidade. Assim a modelagem do *software* é iniciada com vistas ao documento de requisitos. O processo de determinar requisitos é uma tarefa difícil, mas de extrema importância, pois estes formam a base para o planejamento, o acompanhamento do desenvolvimento e a aceitação dos resultados do *software*. Segundo Brooks (1987), “o processo de definir requisitos é inerentemente incompleto, tendo em vista a grande

complexidade do mundo real. É óbvio, no entanto, que sempre estaremos procurando ter requisitos os mais completos possíveis”.

Para Macaulay (1996), os requisitos também podem ser definidos como *"algo que necessita ser projetado"*. Nessa linha, pode-se dizer que um requisito é uma declaração descritiva de exigências, escrita do ponto de vista dos envolvidos, aos quais estão disponíveis a tecnologia e o compartilhamento dos recursos.

Para o engenheiro de requisitos, devem estar bem claros dois aspectos de sua área de atuação: definir requisitos e sob quais condições o fazer. Os requisitos de *software* são a declaração completa do que o *software* irá fazer sem, porém, referir-se à maneira como irá fazê-lo. Para guiar essa etapa, várias técnicas são utilizadas pelos engenheiros de *software*: entrevistas (estruturadas, informais, cenários, etc.), observações, leitura de documentos, questionários, reuniões (GOGUEN, 1993). Assim, a sistematização e a utilização de um conjunto de técnicas, combinadas com o entendimento junto aos usuários, aumenta a confiabilidade e a estabilidade do *software*.



FIGURA 1 – COMPONENTES DA ELICITAÇÃO DE REQUISITOS (KOTONYA, 1998)

A Figura 1 propõe quatro dimensões para ER: **Aplicação do domínio**: é o conhecimento geral da área onde o *software* está sendo construído. É onde ocorrem os fenômenos que caracterizam os problemas

referentes aos requisitos (JACKSON, 1995). **Problema a ser solucionado:** nesse contexto são identificados os problemas específicos do usuário, que devem ser minuciosamente entendidos, aumentando assim o conhecimento do domínio da aplicação. **Contexto do negócio:** os *softwares* são geralmente construídos para contribuir de algum modo para o desenvolvimento do negócio ou da organização. Contribuem para atingir plenamente os objetivos da organização e aumentar os ganhos na área do negócio, interagindo nas diferentes áreas da organização. **Necessidades e restrições dos envolvidos:** os *stakeholders* são as pessoas que, direta ou indiretamente, são afetadas pela solução do *software* a ser construído, e para quem o resultado do processo de desenvolvimento interessa.

Algumas conseqüências do desenvolvimento inadequado da atividade de definição de requisitos podem ameaçar o projeto do *software*, tais como: a produção de *software* que não atende às necessidades dos usuários; o aumento de custos e de cronograma; a realização de atividades desnecessárias ou até mesmo duplicadas; usuários insatisfeitos; aumento da atividade de manutenção e, conseqüentemente, desentendimento entre os engenheiros de *software*. Também, há possibilidades de ocorrerem problemas com a especificação, projeto técnico, implementação, interfaces e manutenção (ambigüidades na especificação, incertezas tecnológicas, tecnologia de ponta). São os outros riscos da construção do *software*.

O objetivo da atividade de especificação de requisitos é documentar os resultados da análise de requisitos. Dessa forma, os requisitos acordados durante a análise de requisitos deverão ser especificados de forma clara e precisa, em um conjunto de documentos, que recebe o nome de especificação de requisitos.

Normalmente são expressos em linguagem natural, diagrama informal ou usando alguma notação que é apropriada para o entendimento do problema. Kotonya e Sommerville (1998) afirmam que a linguagem natural é a única notação que é entendida por todos os envolvidos com o processo de requisitos. Reconhecem que, se necessário, pode-se utilizar diagramas,

tabelas e gráficos, incluídos para melhorar o entendimento do documento de requisitos. Nesse contexto, os requisitos podem ser ainda classificados como estáveis, ou seja, mudam mais lentamente, ou voláteis, que se alteram mais rapidamente. Alguns pesquisadores classificam os requisitos em funcionais e não funcionais (KOTONYA e SOMMERVILLE, 1998).

Sendo o produto do processo de ER o documento de requisitos, é de suma importância agregar qualificação à fonte de informação e quantificação no processo de extração dos requisitos. Essa preocupação é fundamental a fim de permitir a validação dos requisitos, que é o enfoque principal.

Discute-se, a seguir, o processo de definição de requisitos, compreendendo basicamente os subprocessos: elicitação, análise, modelagem ou especificação, validação e gerenciamento dos requisitos.

2.1. ELICITAÇÃO DE REQUISITOS

A elicitação de requisitos é a primeira tarefa do processo de construção de um *software*. Procura encontrar, descobrir os fatos e as situações que compõem o *software*, considerando as necessidades do usuário (KOTONYA, 1998). Neste sentido, usuários e desenvolvedores fazem um trabalho em conjunto para definir o que o *software* deve oferecer, enfocando o problema a ser solucionado.

Nessa etapa, o desafio maior na construção do *software* é a comunicação, ou seja, saber transpor e entender as necessidades do usuário, considerando que muitas alterações podem ser necessárias, decorrentes desse repasse de informação, pois o usuário muitas vezes não sabe o que quer, ou, mudanças no processo de trabalho do usuário, afetam e exigem a modificação do *software*. Para compreender as reais necessidades do usuário, é necessário entender as atividades que realiza no contexto de seu trabalho. A partir desse ponto, entender as atividades realizadas pelo usuário, por meio de seus principais aspectos, é o fator central durante a elicitação de requisitos.

A atividade de elicitação de requisitos não se resume apenas a questionar, perguntar ao usuário o que deseja, mas também é um processo de análise criteriosa da sistematização do *software*.

Sendo a elicitação de requisitos um processo que está em constante evolução, toda vez que eventuais alterações ou inovações dos requisitos se modificam, o modelo resultante do processo de elicitação deve ser alterado também. Esse movimento de atualizar os requisitos, por um lado pode gerar custos altos no desenvolvimento e, por outro lado, pode aumentar consideravelmente o custo do produto final, em decorrência de alterações que serão demandadas para a aceitação do *software* (THOMAS, 2005). Sommerville e Sawyer (1997) sugerem um conjunto de diretrizes detalhadas para a elicitação de requisitos:

- Avaliar a viabilidade técnica e de negócio do *software* proposto.
- Identificar as pessoas que irão auxiliar a especificar os requisitos e entender seu viés organizacional.
- Definir o ambiente técnico, por exemplo: arquitetura computacional, sistema operacional, onde o *software* será desenvolvido.
- Identificar “restrições de domínio”, ou seja, características do ambiente de negócio, específico do ambiente de aplicação, que limitam a funcionalidade ou o desempenho do *software* a ser construído.
- Definir um ou mais métodos de elicitação de requisitos, por exemplo: entrevistas, reunião de equipes.

Sommerville (2003) ainda sugere diretrizes, ou seja:

- Encontrar as pessoas certas para trabalhar no *software*;
- Definir diversos métodos para desenvolver a elicitação de requisitos, como entrevistas, discussão em grupo, reuniões em grupo.
- Identificar requisitos ambíguos como amostras para a prototipação.

- O produto final dessa fase deve ser revisado por todas as pessoas envolvidas. Quanto mais pessoas participarem, maior será o auxílio, já que os requisitos serão mais bem definidos a partir de diferentes pontos de vista.
- Certificar-se de documentar as razões de cada requisito.
- Criar cenários para ajudar os envolvidos a melhor localizar os requisitos-chaves.

2.1.1. TÉCNICAS DE ELICITAÇÃO DE REQUISITOS

As técnicas de elicitação de requisitos são os recursos utilizáveis como suporte aos processos de ER e consistem em procedimentos sistemáticos que auxiliam o engenheiro de requisitos na tarefa de elicitação. As técnicas variam de acordo com o tipo de modelagem utilizado. Funcionam como canais de comunicação entre as fontes de informação, especialistas do domínio e engenheiros de requisitos (NUSEIBEH, 2000). A principal vantagem consiste em respeitar a perspectiva do usuário, fazendo com que a coleta de requisitos atenda aos seus desejos e tornando essencial sua participação no processo (BATISTA, 2003).

O conceito de técnica pode ser definido como o conjunto de processos de uma arte. Estabelecem um conjunto de métodos e de ferramentas aplicáveis às atividades dos processos de elicitação, análise, validação, documentação e manutenção de requisitos.

As várias técnicas constituem uma coleção de alternativas para a escolha da técnica específica ou de um conjunto de técnicas adaptáveis ao processo de ER, à comunicação entre os envolvidos; ao desenvolvimento do conhecimento do problema, à documentação e ao gerenciamento dos requisitos. A seguir apresentam-se algumas técnicas de elicitação.

2.1.1.1. OBSERVAÇÃO

Na observação os desenvolvedores devem estar inseridos na rotina de trabalho do usuário, tentando entender e descrever as principais

atividades que realiza, sem ocorrer interferências. Uma técnica para observação amplamente utilizada nas áreas da sociologia, antropologia e psicologia, e agora utilizada para o processo de elicitação de requisitos, é a etnografia.

A etnografia é uma técnica de elicitação de requisitos utilizada para efetuar a aquisição do conhecimento, de forma a ter um conhecimento o mais amplo possível do *software*. Esta integração da observação com a etnografia deve-se dar de forma a quem irá obter as informações passe a ser visto como um usuário, ou seja, podendo ter as mesmas perspectivas que teria um usuário. Sua principal vantagem é a possibilidade de uma visão de dentro para fora mais completa e perfeitamente ajustada ao contexto, evitando o problema do conhecimento tácito, pois as pessoas não precisam descrever o que fazem e sim demonstrar o que e como fazem. Uma das desvantagens apontadas refere-se ao tempo gasto e também à pouca sistematização do seu processo. A técnica de etnografia insere o observador dentro do ambiente de trabalho, a fim de observar e anotar detalhadamente as tarefas do usuário (KOTONYA, 1998).

Estudos etnográficos mostram que a descrição do trabalho é mais rica e complexa do que o sugerido pelos modelos de *softwares* usuais. Os envolvidos não têm que explicar o seu trabalho. Nesse contexto as interações implícitas são reveladas, assim como fatores sociais e organizacionais importantes podem ser observados.

Sommerville (2003) diz que a etnografia pode ser combinada com outras técnicas de elicitação de requisitos, tal como a prototipação. A etnografia não é, portanto, uma abordagem completa para a obtenção de requisitos, devendo ser completada com outras abordagens, tal como entrevistas específicas com os futuros usuários.

2.1.1.2. QUESTIONÁRIO

O questionário é um instrumento que permite estabelecer a elicitação de requisitos, a partir do conteúdo e estrutura de informações

contidas de forma descrita, por um conjunto de eventos e ações, para obter informações de um grande número de pessoas. Nem todas as informações integrantes do contexto do desenvolvimento do *software* são passadas. Muitas dessas informações são do conhecimento cultural de cada indivíduo. Entretanto, precisam ser obtidas diretamente na fonte.

Assim a elaboração do questionário é desenvolvida para que se entenda melhor o processo, e se adquira mais eficazmente o conhecimento sobre o domínio das informações, além do que foi comunicado. Para a utilização de questionários, Fournier (1994) sugere:

- Elaboração do questionário: estabelece o tipo de informação que se deseja coletar, escolha do layout do questionário (questões de múltipla escolha, descritivas ou dissertativas); define a amostra, ou seja, a identificação dos respondentes no universo.
- Identificação dos respondentes: estabelece a opção de identificação do respondente ou não, do questionário.
- Distribuição do questionário: distribui os questionários contendo as instruções para o seu preenchimento, assim como estabelece os prazos de devolução;
- Análise das respostas: estabelece a classificação das questões para facilitar a tabulação, por grupos e assuntos. Ao tabular as repostas a partir dos questionários, e chegar aos resultados obtidos, enviar cópia aos respondentes.

Para as questões subjetivas, que são também importantes nesse contexto, a colaboração de pessoal mais capacitado sobre tal assunto é importante para a melhor interpretação do contexto.

2.1.1.3. ENTREVISTA

A entrevista é uma das técnicas mais utilizadas pelos desenvolvedores. Acontece por meio de diversos encontros com os usuários, onde relatam os seus trabalhos e as suas necessidades, a partir de questões que respondem ao entrevistador. Geralmente as perguntas são definidas

previamente para auxiliar os encontros e não deixar que outros assuntos sejam tratados. Táticas para a entrevista:

- Identificar os candidatos à entrevista: são os envolvidos no processo (financiadores do projeto, usuários). A partir de cada entrevista podem ser identificados novos entrevistados.
- Preparação para uma entrevista: é preciso esclarecer os objetivos da entrevista, assim como, agendar os encontros para que os entrevistados possam se preparar e contribuir com maior eficácia aos questionamentos abordados. Também deve ser preparada uma relação de questões aos entrevistados, visando sempre extrair informações detalhadas sobre os requisitos do *software*.
- Condução da entrevista: os envolvidos são apresentados, e, a partir daí, estabelecem-se os objetivos da entrevista, o tempo, os assuntos que serão abordados. Nesse momento é possível julgar se o entrevistado tem potencial e preparo para ser um dos atores do processo. Existem habilidades e estratégias para a comunicação oral, podendo ser utilizadas para elevar a qualidade da informação requerida. Deve-se estabelecer diferentes tópicos com questões que abordem as várias direções e níveis de abstração do entrevistado, pois este pode não ter uma compreensão clara das suas necessidades.
- Finalização da entrevista: a entrevista pode ser encerrada quando o entrevistador estiver com as respostas de todos os questionamentos, ou quando este perceber que existe uma exaustão por parte do entrevistado em relação aos questionamentos. As ações que se seguem a essa entrevista devem ser explicadas, inclusive destacando ao entrevistado que uma sumarização da entrevista, um resumo escrito, lhe será entregue para correções e revisão.

Segundo Kotonya e Sommerville (1998) existem dois tipos de entrevista: a **entrevista fechada** é estabelecida pelo entrevistador a partir de um conjunto de questões previamente definidas, focando apenas os assuntos

abordados por esse conjunto de questões junto ao entrevistado, não deixando margem para outras discussões. A **entrevista aberta** é abordada com mais flexibilidade, onde o entrevistador relaciona os principais assuntos a serem discutidos na entrevista. Nessa modalidade o entrevistador pode levantar temas correlatos ao assunto, para levantar discussões e aprofundar o conhecimento, de forma a obter maior compreensão sobre o *software*.

Esse método tem suas desvantagens, pois há dificuldades por parte dos usuários em expressar suas necessidades, omitindo informações, por achar que são óbvias, e que deveriam ser do conhecimento do engenheiro de requisito. Por outro lado, tem a vantagem de ser um canal mais flexível para a obtenção das informações mais subjetivas sobre o *software*, que contribuem também para o aprofundamento do conhecimento.

2.1.1.4. BRAINSTORMING

Essa técnica acontece a partir de várias reuniões, onde as pessoas envolvidas sugerem e abordam idéias sem que sejam estabelecidos critérios ou condução rigorosa das diferentes variações de idéias. Essa técnica pode ser útil, pois trata de diversas visões do problema, auxiliando na interação social.

Nessa fase, esse elenco de idéias é discutido, revisado e organizado. Por ser um processo aberto, pode, dependendo do encaminhamento e do líder do grupo, não produzir uma qualidade significativa, em comparação a outros tipos de processos.

2.1.1.5. JAD

Joint Application Design (JAD), é uma técnica que estabelece a cooperação, compreensão e trabalho em grupo entre usuários e desenvolvedores. Cria uma visão compartilhada sobre o que deve ser o produto do *software*, assim como os desenvolvedores auxiliam os usuários a identificarem problemas, explorando soluções, contribuindo com o maior

envolvimento e comprometimento com o sucesso final do *software* (CARVALHO, 2001). Essa técnica estabelece quatro procedimentos:

- Dinâmica de grupo: o envolvimento do grupo facilita o aumento da capacidade dos indivíduos.
- Uso de técnicas visuais: para criar maior entendimento e comunicação.
- Manutenção do processo, sistematização e racionalização.
- Utilização de documentação-padrão: que todos os participantes preenchem e assinam em cada sessão.

2.1.1.6. CENÁRIOS

Muitos usuários preferem descrever seus processos de trabalho utilizando os cenários. É considerada uma técnica de fácil assimilação, pois utiliza a representação gráfica, proporcionando a reorganização de componentes e de tarefas do domínio, a avaliação; as críticas e as sugestões dos clientes e usuários envolvidos.

Cenários permitem uma compreensão dos problemas atuais pelos engenheiros de requisitos e auxiliam na antecipação da situação futura. O objetivo dessa técnica é focar as atividades que as pessoas realizam nas organizações, auxiliando de forma mais ampla com relação aos problemas atuais onde o *software* será inserido, explicando sua necessidade e unificando critérios (RIDAO, 2000).

A partir dessa técnica, é possível levantar diversas narrativas de situações no domínio, que favorecem a obtenção das informações, como a identificação de problemas e a antecipação das soluções. É uma das formas de representar aos clientes e usuários os problemas atuais e as possibilidades que podem surgir. Segundo Zorman (1995), cenários narram as situações do mundo real, que envolvem agentes interagindo dentro de um determinado contexto.

Não têm como objetivo oferecer uma descrição precisa, mas sim, provocar discussão e estimular novos questionamentos. Essa técnica permite também documentar o levantamento de informações a respeito dos problemas atuais, possíveis eventos, oportunidades de ações e riscos.

Segundo Jacobson (1992), que propõe a introdução dos casos de uso, cenários são situações do macro *software* e a relação com o *software* a ser construído. É um meio de representação de fácil compreensão, proporcionando um desenvolvimento orientado a tarefas, possibilitando maior usabilidade do *software*. A seguir apresentam-se alguns métodos existentes para a utilização de cenários.

a) Método para Análise de Requisitos Baseados em Cenários (SUTCLIFFE, 1998):

- Utilizam-se as técnicas de entrevista, prototipagem; cenários e *rationale*.
- Para o descobrimento dos fatos, a entrevista é utilizada para identificar dados suficientes para a construção de um protótipo.
- Os protótipos são construídos a partir da utilização de ferramentas, tais como *Macromedia Director* ou *VB (Visual Basic)*.
- A partir dos protótipos construídos, a validação é realizada junto aos clientes, sendo este momento gravado em vídeo para análises futuras.
- A análise é feita pela utilização da reunião JAD, onde os cenários são descobertos a partir da descrição das tarefas dos usuários.

b) Ciclos de Questionamento (*Inquiry Cycle*) (POTTS, 1994):

- Segundo Potts (1994), é um modelo para a descrição e suporte de discussões sobre os requisitos do *software*, baseado em consecutivas etapas de questionamento e refinamento posterior desses requisitos.
- Os cenários são derivados pela identificação e decomposição dos objetivos.

- Com o resultado da documentação dos cenários e de uma lista de requisitos, é estabelecido um processo de questionamento onde respostas e justificativas (*rationale*) são registradas, terminando o processo somente quando uma decisão é tomada. Sendo esta negativa, a documentação é modificada e pode resultar em um novo processo.

c) Cenários como apoio à visualização de requisitos (ZORMAN, 1995):

- Aborda a dificuldade de comunicação entre usuários e desenvolvedores, apontando ocorrências de mistura de terminologia e erros de interpretação.
- Os envolvidos, usuários especializados e desenvolvedores utilizam uma representação e vocabulário comum, com vistas à elaboração de cenários.
- Deve utilizar linguagens formais, por pessoas da mesma área.
- É sugerido pela autora utilizar uma ferramenta gráfica para a captura e representação dos cenários, assim cada cenário é identificado pelo nome, categoria e descrição.

Em resumo, os cenários permitem a compreensão dos problemas atuais pelos analistas, e antecipação da situação futura pelos usuários e desenvolvedores. Como toda atividade de análise e especificação de requisitos, a descrição do domínio através de cenários requer técnicas de comunicação e modelagem (BREITMAN, 1998). Assim, a descrição de cenários deve ser apoiada pelas três técnicas de comunicação:

- Entrevistar usuários para entender os problemas e requisitos iniciais. A entrevista com usuários permite que cada um descreva as suas tarefas e os problemas associados.
- A observação direta *in loco* é fundamental para que os analistas possam descrever a situação de uso, como realmente vem ocorrendo na prática.
- Após a elaboração dos cenários, clientes, usuários e desenvolvedores podem participar de encontros para que possam discutir cada um destes cenários. Estes podem ser afixados em quadros na parede, onde os participantes

possam analisá-los e fazer comentários, possivelmente afixando pequenos pedaços de papel a cada uma das cenas.

2.1.1.7. CASOS DE USO

Após saber quais as tarefas associadas a cada papel de usuário, é hora de elaborar os casos de uso (*use cases*), que permitem definir as funções de aplicação que o *software* deverá oferecer ao usuário. Os casos de uso podem ser utilizados durante a análise e especificação dos requisitos para descrever a funcionalidade do *software*. Pode-se dizer que casos de uso são técnicas baseadas em cenários para obtenção de requisitos.

Para Rumbaugh (1994), casos de uso são as possíveis seqüências de interações entre o *software* e um ou mais atores. Descreve que caso de uso não pode ser considerado simplesmente como cenário, mas um conjunto de descrições de potenciais cenários.

O caso de uso descreve um conjunto de seqüências de ações que o *software* desempenha para produzir um resultado esperado pelo usuário. Cada seqüência representa a interação de entidades externas com o *software*. Estas entidades são chamadas de atores, e podem ser usuários ou outros *softwares*. No caso de usuários, um ator representa, na verdade, uma função de usuários. Assim utilizam-se agentes ou atores, representados por bonecos que interagem, e cada classe de interação é representada por uma elipse com um nome (SOMMERVILLE, 2003).

Um caso de uso especifica o comportamento do *software* a ser desenvolvido sem, no entanto, especificar como esse comportamento será implementado. Representa o que o *software* faz e não como o *software* faz, proporcionando uma visão externa e não interna do *software*. Cada caso de uso define um requisito funcional do *software*. Diagramas de seqüência podem ser usados para adicionar detalhes ao caso de uso, ao mostrar a seqüência de eventos no *software*. Por exemplo, a Figura 2 descreve o caso de uso de compras de produtos e distribuição de alimentos.



FIGURA 2 – EXEMPLO DE CASO DE USO (COMPRAR PRODUTOS E DISTRIBUIR ALIMENTOS)

2.1.1.8. TEORIA DA ATIVIDADE

Imbuídos da preocupação do envolvimento no contexto social do processo de ER, pesquisadores estudam formas de resolver os problemas durante a elicitaco de requisitos (MARTINS, 2001) (MYLOPOULOS, 1999) (NUSEIBEH, 2000).

Nessa linha, Martins (2001) prope uma nova tcnica de elicitaco baseada na Teoria da Atividade (TA), acreditando que problemas de comunicao entre desenvolvedores e usurios, mudanas dos requisitos, dificuldade dos usurios em saberem exatamente o que querem, no so apenas problemas oriundos dos fatores tecnolgicos.

A TA  formada por um conjunto de princpios e conceitos que buscam compreender e explicar porque e como as atividades humanas so desenvolvidas. Esses princpios so apresentados a seguir:

1) Princpio da unidade entre conscincia e atividade:  o princpio fundamental da TA, onde conscincia e atividade so concebidas de forma integrada. A conscincia pode ser compreendida a partir de vrios aspectos psicolgicos, utilizados no âmbito racional, e a atividade como a interao humana com sua realidade objetiva.

2) Princpio da orientao a objetos: esse princpio identifica a TA onde os seres humanos interagem com o ambiente. Esse ambiente  formado por entidades (objetos) combinados por todos os tipos de caractersticas objetivas.

3) Princípio da estrutura hierárquica da atividade: a TA estabelece diferenças para os procedimentos humanos em três níveis - atividade: orientada a motivos; ação: orientada a metas e a operação: orientada a condições de realização.

4) Princípio da internalização-externalização: descreve os mecanismos básicos da origem dos processos mentais. Relata que os processos mentais são derivados das ações externas através do curso da internalização. A internalização relata a absorção das informações pela mente humana, a partir do contato com o ambiente onde está inserida. A externalização é o processo contrário. É manifestada a partir dos atos, de forma que possa ser verificados e corrigidos, caso necessário.

5) Princípio da mediação: a atividade humana é medida por ferramentas externas e internas, que são “veículos” da experiência social e do conhecimento cultural.

6) Princípio do desenvolvimento: segundo a TA, entender um fenômeno significa conhecer como este se desenvolveu até sua forma atual, pois ao longo do tempo sofre alterações.

De acordo com a TA, uma atividade retrata uma forma de agir, de um sujeito, direcionada a um objeto. Em nível individual, uma atividade possui três elementos: sujeito, objeto e ferramenta de mediação. Sendo o sujeito, o agente que atua sobre o objeto da atividade; o objeto (pode ser material ou menos tangível) é o elemento para o qual as ações da atividade estão direcionadas. O relacionamento recíproco entre o sujeito e o objeto da atividade é sempre mediado por uma ou mais ferramentas (também chamadas de artefatos de mediação) (MARTINS, 1999) (KUUTI, 1996).

Martins (2001) propõe, a partir da TA, uma metodologia que utiliza os conceitos da atividade a fim de auxiliar a organização do processo de elicitação de requisitos. Essa metodologia está dividida em três etapas, a saber:

1) Divisão do problema em atividades (unidades de elicitação de requisitos).

- 2) Delineamento do contexto das atividades (para cada atividade).
- 3) Descrição da estrutura hierárquica das atividades (para cada atividade).

Na Figura 3 destacam-se essas atividades. À medida que se identificam e descrevem os requisitos, estes vão se tornando mais claros e a sua eliciação vai se dando de forma gradual.

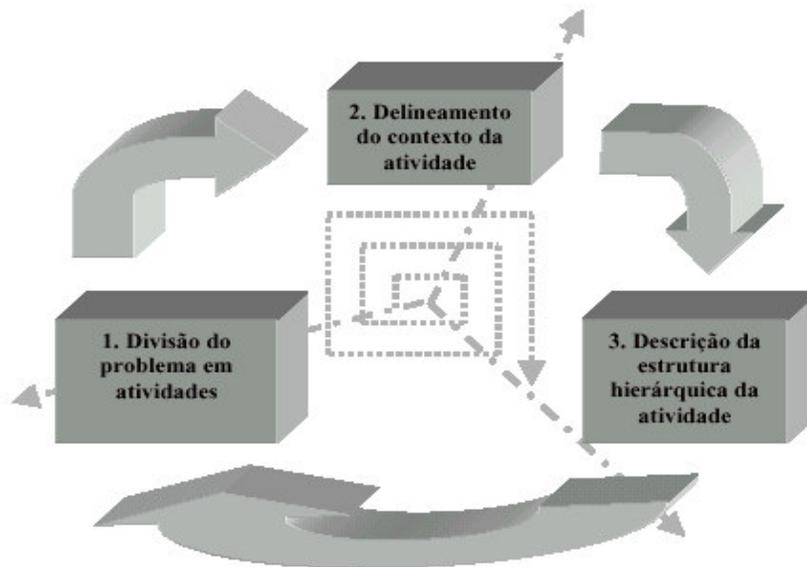


FIGURA 3 – ETAPAS DA METODOLOGIA DE ELICITAÇÃO DE REQUISITOS (MARTINS, 2001)

De forma resumida, a TA é dada quando a ação humana é analisada dentro de uma atividade. Assim, a atividade passa a ser vista como a unidade básica de análise de situações, colaborando para a obtenção mais adequada dos requisitos do *software*.

A metodologia de eliciação de requisitos baseada na TA tem como foco o conceito de atividade. Dessa forma, a atividade deve ser considerada como uma unidade de eliciação de requisitos. Assim, a ação humana é analisada dentro do contexto de uma atividade, onde a atividade passa a ser vista como a unidade básica para a análise de situações, colaborando para a obtenção dos requisitos do *software*. Baseando-se nos preceitos da TA, Martins (2001) identifica os seguintes procedimentos:

- Levantar atividades candidatas: qualquer tarefa que os envolvidos no *software* realizam, pode ser vista como uma atividade. Nesse momento não é feito um “filtro” para estabelecer atividades.
- Selecionar atividades: realizado o levantamento inicial das atividades, deve-se ter como meta uma classificação mais precisa das mesmas (tarefas).
- Descrever Histórico das atividades selecionadas: nesse momento busca-se definir a trajetória da atividade realizada pelo sujeito (usuário), a partir de um resgate histórico da sua origem e desenvolvimento. Essa descrição deve destacar o formato inicial da atividade, a fim de contextualizar as atividades diárias de seus atores.
- Delinear o Contexto da Atividade: definidas as atividades, a etapa seguinte é a contextualização de cada uma das atividades.
- Identificar os Motivos e Resultados da Atividade: a identificação dos elementos (motivo – desejos e necessidades humanas), (resultado das atividades – processo final da transformação embutido na atividade), resulta na dimensão da atividade, ou seja, a sua delimitação.
- Identificar os Elementos em Nível Individual: três elementos formam a atividade em nível individual que são, sujeito, ferramenta de mediação (técnica ou psicológica) e objeto de transformação da atividade (pode haver uma variedade de cada elemento de uma mesma atividade). O sujeito é o agente principal, atuando diretamente sobre o objeto de transformação da atividade. Na seqüência, parte-se para a identificação do objeto, onde ocorre a compreensão sobre a atividade, pois o envolvimento dos participantes colabora para a identificação dos demais elementos da atividade.
- Identificar os Elementos em Nível Social: o elemento principal é a comunidade em que o sujeito da atividade está inserido. Dessa forma, a comunidade pode influenciar, de alguma maneira, a realização da atividade.

- Modelar a Atividade através do Diagrama de Engeström: com todos os elementos da atividade identificados, é utilizado o diagrama de Engeström para modelagem (ENGESTRÖM,1987). Existe um relacionamento intrínseco entre sujeito, objeto e comunidade.
- Descrever a Estrutura Hierárquica da Atividade: nesta etapa a atividade deve ser estruturada em suas ações e operações. Com vistas a elicitação dos requisitos essa etapa é fundamental, pois aponta as tarefas executadas inicialmente pelas pessoas e, neste momento, podem ser mapeadas como requisitos funcionais do *software*.
- Identificar as Ações e Operações da Atividade: esta tarefa identifica ações realizadas pelos sujeitos durante a execução das atividades, considerando regras de conduta na comunidade em que estão inseridas.
- Descrever as Metas das Ações: a descrição das metas das ações colabora com a compreensão sobre as ações que o sujeito realiza, conectando-as aos resultados esperados da atividade. Isso acontece na medida em que as metas vão sendo alcançadas pela execução das ações.
- Descrever as Condições de Realização das Operações: esta etapa completa a captura de todos os elementos relativos à atividade, sendo que o sujeito deve reconhecer a operação antes realizada.

2.2. ANÁLISE DE REQUISITOS

A análise de requisitos captura as necessidades básicas funcionais e não-funcionais do *software* que deve ser desenvolvido, modela o problema principal (classes, objetos) e cria um modelo ideal, sem levar em conta requisitos técnicos do *software*. Assim, os envolvidos negociam para decidir sobre quais requisitos serão aceitos, de forma que resulte num “acordo” comum, cujo resultado são os requisitos acordados.

A finalidade da análise de requisitos é estabelecer um modelo de requisitos. Deve considerar a comunicabilidade, apresentando

representações que auxiliem a compreensão sobre os objetivos do *software*, pois o forte envolvimento e a participação dos usuários, desde o princípio do processo de desenvolvimento, garantem o sucesso do *software*. Como se trabalha sob um universo volátil, a análise de requisitos é necessária porque existem invariavelmente conflitos entre os requisitos a partir de diferentes fontes, informações incompletas ou os requisitos descritos podem estar incompatíveis com as restrições ambientais.

O fator humano no desenvolvimento de um *software* é algo existente no processo de ER. Sendo assim, possui características de volatilidade não previsíveis e complexos. Estabelecer um rigor matemático ou hierarquia para reduzir essa complexidade pode acarretar problemas de distorção no entendimento.

Descobrir os requisitos é uma tarefa difícil, pois se trata da exploração do que perguntar. Nesse sentido, pode-se dizer que a análise de requisitos é estabelecida sob três aspectos fundamentais: do que se trata (fatos e fenômenos), a quem diz respeito (fontes de informação) e a definição de uma forma de comunicação entre os envolvidos (linguagem para o descobrimento e documentação dos requisitos).

Identificar os fatos ou fenômenos depende do conhecimento do ambiente ou domínio da aplicação pelo engenheiro de requisitos. Enquanto os requisitos estão sendo identificados, concomitantemente, alguma análise acaba acontecendo. Nesse momento é comum encontrar problemas, conflitos e ambigüidades, que necessariamente devem ser tratados junto aos desenvolvedores e usuários, para obter um acordo em relação aos requisitos.

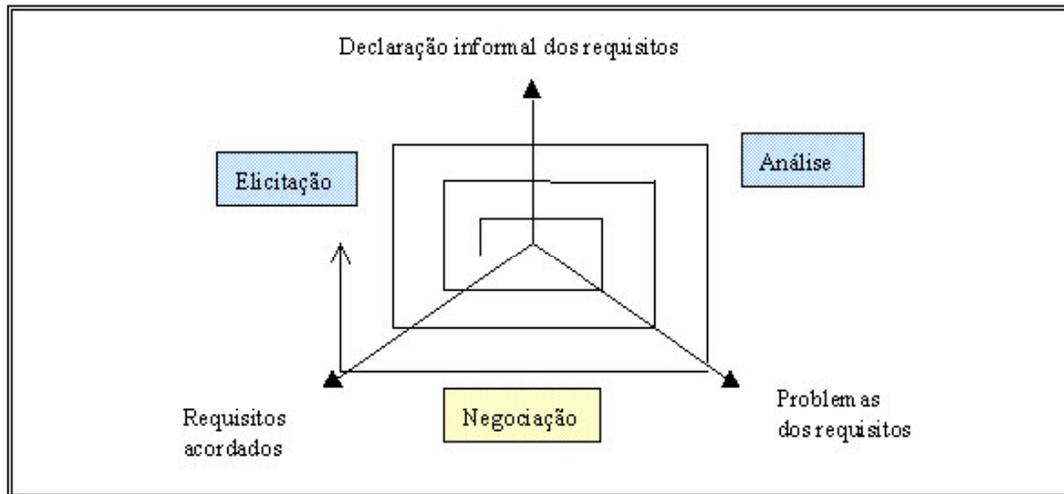


FIGURA 4 – MODELO ESPIRAL DE ELICITAÇÃO, ANÁLISE E NEGOCIAÇÃO (KOTONYA, 1998)

A Figura 4 apresenta a relação entre as atividades de elicitação e a análise de requisitos, atividades estas, que são ligadas e interagem com a atividade de negociação. Essas atividades são discutidas e acordadas a partir dos problemas encontrados nos requisitos.

A definição precisa de requisitos do *software* depende diretamente do grau de comunicação entre os envolvidos, no domínio da aplicação. Dessa forma a atividade de análise de requisitos está inteiramente ligada à elicitação de requisitos.

Segundo Kotonya (1998), as atividades a serem desenvolvidas são as seguintes:

- Extração de requisitos: os requisitos são descobertos por meio de consultas às pessoas envolvidas, dos documentos, do domínio do conhecimento e de estudos de mercado.
- Análise e negociação: os requisitos são analisados detalhadamente, descartados ou aceitos.
- Documentação: os requisitos aceitos são devidamente documentados, utilizando-se para tanto uma forma apropriada de descrição e detalhamento.
- Validação: os requisitos são checados rigorosamente em busca de consistência e certificação, estabelecendo com clareza o que se quer atingir.

- Gerenciamento: onde os requisitos são controlados em função da dinâmica das mudanças no ambiente do *software*, ou do usuário, em relação às reais demandas.

2.3. MODELAGEM OU ESPECIFICAÇÃO DE REQUISITOS

A modelagem ou especificação de requisitos é atividade fundamental para fazer do desenvolvimento do *software* uma atividade de engenharia. É onde os resultados da elicitação e da análise de requisitos são documentados, organizando-se assim os requisitos do *software*.

Mylopoulos (1992) descreve que a especificação é uma atividade que declara formalmente, de forma precisa, aspectos do mundo físico e social, com o objetivo de melhorar o entendimento e a comunicação. Para Kotonya (1998), a especificação é uma forma de comunicação sistemática entre analistas e projetistas do *software*.

O uso de modelos nas diversas áreas da engenharia sempre foi fundamental para o desenvolvimento de bons *softwares*. Os modelos permitem uma visão antecipada do *software* a ser desenvolvido e permitem análise e avaliação prévia, antes mesmo dos produtos serem construídos.

Assim, essa atividade é resultante da análise dos requisitos, que deve descrever de maneira sistemática e precisa quais propriedades funcionais são necessárias para resolver o problema do domínio, utilizando-se um conjunto de documentos.

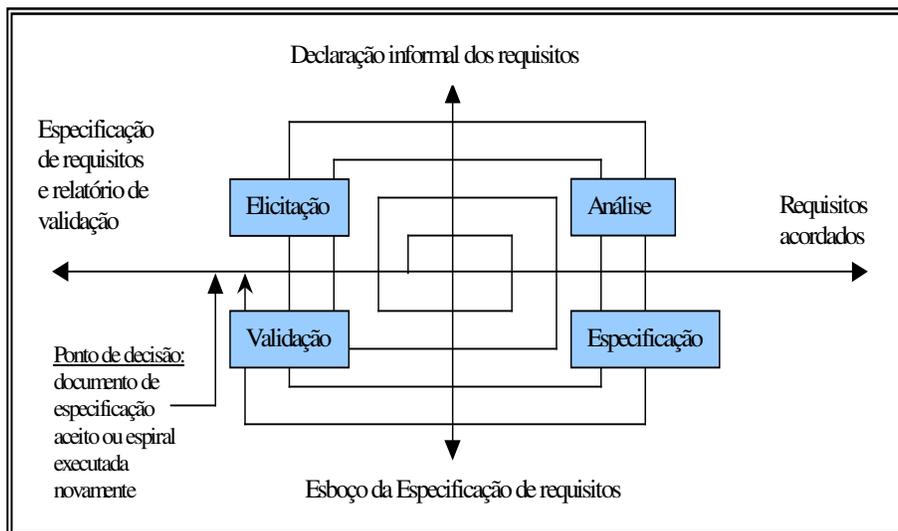


FIGURA 5 – MODELO ESPIRAL DO PROCESSO DE ENGENHARIA DE REQUISITOS (KOTONYA, 1998)

A Figura 5 apresenta um modelo espiral, onde as diferentes atividades da ER, elicitação, análise, especificação e validação são repetidas até que uma decisão seja tomada e o documento de requisitos seja aceito pelos envolvidos. Com a elicitação, obtêm-se os requisitos que oferecem subsídios para a análise dos requisitos. Esses requisitos acordados entre usuários e desenvolvedores, são especificados na etapa de especificação dos requisitos, gerando os esboços da documentação dos requisitos. Os esboços de especificação são validados gerando a especificação de requisitos final, juntamente com o relatório de validação.

Diversas técnicas e formalismos apareceram desde o surgimento da Engenharia de *Software* no final da década de 60, contribuindo para o avanço dessa área. No final dos anos 90, as melhores técnicas foram reunidas, tornando-se um padrão na indústria de *software* a Linguagem de Modelagem Unificada (UML - *Unified Modeling Language*), criada pela unificação das técnicas OMT (*Object Modeling Technique*) de Rumbaugh (1994), a OOSE (*Object-Oriented Software Engineering*) de Jacobson (1992) e a OOAD (*Object-Oriented Analysis and Design*) de Booch (1994), que são linguagens para especificação, visualização, construção e documentação de artefatos de *softwares* (BOOCH, 1999).

Uma vez que a UML foi desenvolvida para ser usada em diversos métodos de desenvolvimento, para utilizá-la com sucesso é

necessário adotar algum tipo de método de desenvolvimento, especialmente em *software* de grande porte onde a organização de tarefas é essencial.

A UML contém notações e regras que tornam possível expressar modelos orientados a objetos. Mas não prescreve, porém, como o trabalho tem que ser feito, ou seja, não possui um processo de como o trabalho tem que ser desenvolvido.

A utilização de um processo de desenvolvimento torna mais eficiente calcular o progresso do projeto, controlar e melhorar o trabalho, de forma que se tenha uma visão sobre "o que fazer", "como fazer", "quando fazer", e "por que deve ser feito". A modelagem de dados também pode ser realizada de duas maneiras diferentes: de-cima-para-baixo ("top-down") ou de-baixo-para-cima ("bottom-up"). No primeiro caso parte-se das classes genéricas para depois identificarem as especializações, enquanto que no segundo faz-se o caminho contrário.

2.4. VALIDAÇÃO DE REQUISITOS

A validação é a etapa final da ER. É onde os requisitos devem ser checados cuidadosamente, verificados para que se certifique que representam uma descrição aceitável de como o *software* deve ser implementado, ou seja, devem estar completos, expressando o que o usuário realmente quer. É neste momento que a checagem dos requisitos acordados e especificados representa a descrição estruturada do *software* a ser desenvolvido (KOTONYA, 1998).

O principal mecanismo de validação de requisitos é a revisão formal (exemplo: revisão técnica formal). O trabalho técnico precisa de revisões e pode ser realizado por auditorias, com base nas especificações de requisitos. Um pequeno *checklist* pode ser usado para que se possa avaliar se os requisitos estão claramente estabelecidos, se as fontes (por exemplo, pessoa, regulamento, documento) do requisito estão identificadas.

Evitar erros nas fases anteriores do desenvolvimento, que eventualmente são explicitados no ciclo final de vida do *software*, é uma das principais preocupações para os desenvolvedores, pois o custo para correções ao final do processo é bem maior, assim como o tempo de desenvolvimento.

Como o produto final da validação dos requisitos, os problemas são identificados. Algumas técnicas utilizadas tornam o relato desses problemas melhor identificados, pois os problemas encontrados sofrerão ações que serão executadas conforme os padrões organizacionais, tais como:

- Revisão de requisitos: análise manual sistemática dos requisitos.
- Prototipação: utilização de um modelo executável do *software*.
- Geração de casos de teste: desenvolvimento de testes para verificar os requisitos.
- Análise automatizada da consistência: verificação da consistência de uma descrição de requisitos estruturada.

Durante a validação alguns problemas de requisitos podem ser encontrados, tais como, requisitos ambíguos; falhas nos modelos de *software*; conflitos de requisitos e falta de conformidade com padrões de qualidade. Tais problemas devem ser tratados antes da aprovação do documento de requisitos. Alguns problemas podem ser tratados apenas corrigindo-se a documentação. Outros problemas terão de ser tratados a partir da revisão das atividades anteriores da ER, ou seja, estas devem ser realizadas novamente.

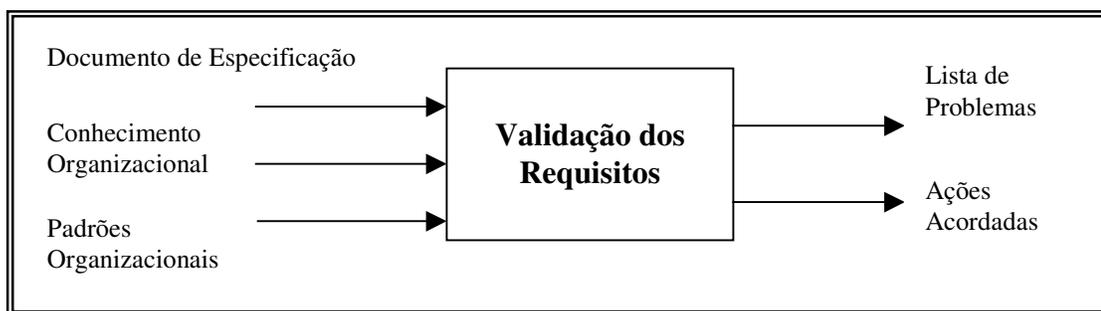


FIGURA 6 – VALIDAÇÃO DE REQUISITOS (KOTONYA, 1998)

A Figura 6 mostra as entradas e saídas do processo de validação de requisitos, sendo as entradas: documento de especificação (documentação formatada e sistematizada segundo os padrões organizacionais); conhecimento organizacional (envolvimento de pessoas ligadas à cultura e estrutura organizacional); padrões organizacionais (apresenta os requisitos acordados e negociados entre os envolvidos). As saídas: lista de problemas (apresenta os problemas encontrados, identificados por tipo de problema, ambigüidade, imperfeições, etc.) e ações acordadas (com os problemas relacionados na lista de problemas, as ações que serão efetuadas).

Assim, após esse processo, onde os requisitos são formalizados e corrigidos, a validação consiste em, juntamente com os usuários, checar cuidadosamente a consistência dos requisitos, de maneira a obter um documento de requisitos de *software* que satisfaça os clientes.

2.5. GERENCIAMENTO DOS REQUISITOS

O gerenciamento de requisitos envolve processos, procedimentos e padrões que são utilizados para gerenciar as alterações dos requisitos do *software*. Esse gerenciamento garante que sejam levantadas todas as informações, avaliação de custos e benefícios. Esse gerenciamento formal é essencial, pois garante que as alterações propostas aos requisitos tenham suporte com relação às metas gerais do *software*. Os objetivos principais do gerenciamento de requisitos, segundo Kotonya (1998), são:

- Gerenciar mudanças nos requisitos acordados.
- Gerenciar o relacionamento entre os requisitos.
- Gerenciar as dependências entre os documentos de requisitos e outros documentos elaborados durante o processo de ER.

Ainda segundo Kotonya (1988), o processo de gerenciamento de requisitos pode ocorrer da seguinte maneira:

- Detectado um problema com o requisito, seja através da análise de requisitos ou a partir de problemas operacionais, analisam-se os requisitos, utilizando-se as informações do problema e propõem-se as alterações nos requisitos.
- Essas alterações propostas são analisadas para que se levante quantos requisitos serão atingidos, assim como os custos e o prazo para a realização das alterações.
- As alterações são implementadas e se estabelece uma série de ações corretivas no documento de requisitos, ou constrói-se uma nova versão desse documento, devendo-se revalidar tudo.

Para o processo de análise dos requisitos, Kotonya (1998) descreve seis atividades básicas:

- 1) Identificar se as alterações solicitadas são válidas, pois às vezes os clientes se confundem com os requisitos, solicitando assim alterações desnecessárias.
- 2) Verificar se os requisitos afetados pelas alterações são descobertos.
- 3) Verificar se informações de rastreabilidade são utilizadas para identificar todos os requisitos, que podem ser afetados pelas alterações;
- 4) Propor as alterações que necessitam ser realizadas. Deve haver consulta aos clientes nesse momento, para garantir o acordo em relação às alterações.
- 5) Estimar os custos para execução das alterações, assim como o esforço despendido e o tempo a ser gasto nessa atividade. Considerar também a disponibilidade de recursos.
- 6) Contatar os clientes a fim de verificar se os custos levantados para as alterações são aceitáveis. O cliente pode achar o custo muito alto e resolver modificar a solicitação de alteração. Nesse caso, todo o processo será repetido.

O gerenciamento do ciclo de vida dos requisitos se desenvolve concomitantemente com todas as outras etapas. É uma atividade primordial

que tem como objetivo a recuperação das alterações e avaliação de impactos das ocorrências, devido a mudanças no *software*. O gerenciamento de requisitos é a atividade de compreensão e controle das mudanças nos requisitos do *software*, e ocorre em conjunto com as outras atividades da ER (SOMMERVILLE, 2003).

Justifica-se tal importância, pois, dentro da ER, essa atividade surge como um facilitador, através do qual recuperar informações irá depender da pré-definição de atributos gerenciáveis dos requisitos durante a fase de elicitación e definição dos mesmos. Essa forma de gerenciar demonstra a maturidade organizacional, e é a base para a comunicação entre os envolvidos no processo. (SEI, 1997).

As mudanças nos requisitos são vistas durante todo o processo de desenvolvimento do *software*. Assim, gerenciar os requisitos a partir de descrição em etapas anteriores estabelece a rastreabilidade dos mesmos, em ambas as direções, para frente e para trás, isto é, desde a raiz, desenvolvimento e especificação dos requisitos. Tais mudanças são inevitáveis, não significando que o processo estabelecido da ER tenha sido falho (KOTONYA, 1998).

Segundo Pinheiro (1996), o que estabelece a motivação para a rastreabilidade dos requisitos são as seguintes indicações:

- Os requisitos evoluem durante a vida de um *software*.
- Os requisitos são contextualizados e dependem de onde surgem e dos detalhes dessa situação particular.
- Os requisitos estão ligados ao processo de desenvolvimento e o gerenciamento dos objetivos do projeto é útil por todo o ciclo de vida dos requisitos.

No presente capítulo tratamos de conceitos da Engenharia de Requisitos e de técnicas para garantir a qualidade de seu produto final: documento de requisitos. No próximo capítulo, aprofundando a análise de

requisitos, vamos explorar a análise de requisitos orientada a metas, com o intuito de contextualizar os conceitos sobre metas e apresentar técnicas e métodos de suporte a orientação a metas.

3. ANÁLISE DE REQUISITOS ORIENTADA A METAS

A fase de definição dos requisitos é crítica no ciclo de vida de um *software*, pois uma compreensão completa dos requisitos de *software* é fundamental para um bem sucedido desenvolvimento do mesmo, uma vez que os requisitos formam a base para o planejamento, o acompanhamento do desenvolvimento, e a aceitação dos resultados do projeto.

Nesse contexto, um dos objetivos da ER é ultrapassar barreiras de comunicação entre os clientes, usuários e os desenvolvedores, para que os requisitos possam ser capturados e modelados corretamente. Dessa forma, técnicas podem ser utilizadas para apoiar a tarefa de especificação e elicitação de requisitos, cuja principal vantagem consiste em respeitar a perspectiva do usuário, fazendo com que a coleta de requisitos atenda aos seus desejos, tornando essencial sua participação no processo (LEITE, 2004).

Nessa direção, técnicas e metodologias incentivam o engajamento dos usuários no desenvolvimento do *software*, contando com a sua participação efetiva no sentido de trocar perspectivas e negociar requisitos do *software*. Discute-se, a seguir, a Análise de Requisitos Orientada a Metas.

3.1. CONCEITOS SOBRE METAS

A abordagem orientada a metas é uma técnica eficaz na definição dos *porquês* do *software* e identificação dos requisitos (ÁNTON, 1996), por isso é amplamente utilizada em diferentes partes da ER (YU, 1998). O argumento dessa abordagem é que as razões de um *software* determinam as diretrizes do desenvolvimento para a organização, para a qual é requerido.

Pesquisadores vêm cada vez mais reconhecendo a importância das metas no processo de ER. Tal reconhecimento motivou toda uma linha de pesquisa sobre a modelagem, especificação e raciocínio baseado em metas, para múltiplos propósitos, tais como a elaboração de requisitos, a verificação ou o gerenciamento de conflitos (LAMSWEERDE, 2001).

Segundo Lamsweerde (2001), há muito tempo as metas têm sido consideradas componentes essenciais envolvidos no processo de ER.

[...] a definição dos requisitos deve dizer *por que* o *software* é necessário, baseado em condições tanto atuais quanto previstas, *o que* pode constituir em operações internas ou externas, e também *para que* o *software* servirá, mais ainda, pode revelar de que maneira o *software* deverá ser construído.

Uma meta é simplesmente uma situação desejada. A especificação de metas, que declara os resultados desejados e não a maneira em obtê-los, permite entender as razões dos requisitos e perceber quais envolvidos no processo colaboram na sua obtenção.

As metas abordam, em diferentes níveis de abstração, os vários objetivos que o *software* deve alcançar. A ER orientada a metas preocupa-se com a utilização de recursos para elicitar, elaborar, estruturar, especificar, analisar, negociar, documentar e modificar os requisitos.

Segundo Lamsweerde (2001), as metas podem ser identificadas de forma sistemática, por meio de palavras-chave intencionais, fornecidas preliminarmente pela documentação, entrevistas etc, aos engenheiros de requisitos. Podem estar explícitas, identificadas pelos envolvidos no projeto ou estar presentes na documentação disponível aos engenheiros de requisitos, ou ainda implícitas, necessitando da elicitação de metas.

As metas são conduzidas para dar apoio à elaboração dos requisitos. Fornecem um critério completo para a especificação de requisitos, sendo que a especificação é considerada completa, se todas as metas relatadas são obtidas pela especificação; as metas propiciam razões fundamentais para os requisitos – um requisito existe por causa de algumas metas primordiais, às quais uma base é fornecida e representam as raízes para a detecção de conflitos entre requisitos.

As metas podem ser formuladas em diferentes níveis de abstração, variando de “alto nível”, que seriam preocupações estratégicas

(como “atenda um maior número de passageiros”, para um *software* de transporte ferroviário, ou “forneça um serviço de retirada de dinheiro”, para um *software* de rede de caixas eletrônicos), ou de “baixo-nível”, englobando questões técnicas (como “comando de aceleração atendido pontualmente”, para um *software* de transporte ferroviário, ou “cartão retido após três tentativas com *password* errado”, para caixas eletrônicos). O *software* ao qual a meta se refere pode ser o atual ou um *software* futuro, ambos envolvidos com o processo de ER. Sendo assim, as metas de “alto nível” referem-se, freqüentemente, a ambos os *softwares*. Para Lamsweerde (2001), há muitas razões pelas quais as metas são tão importantes no processo de ER, como as apresentadas a seguir:

- Satisfazer os requisitos é fundamental para a ER, considerando que as metas fornecem um critério preciso para a completa especificação dos requisitos. A especificação estará completa, respeitando-se as metas, se todas as metas forem comprovadamente possíveis de serem obtidas a partir da especificação e das propriedades conhecidas a respeito do domínio considerado.
- Evitar requisitos irrelevantes é também primordial para a ER. As metas fornecem um critério preciso para a *pertinência* dos requisitos. Um requisito será pertinente com base nas metas do domínio considerado, ou seja, baseado no contexto ou na abrangência do *software*, ou ainda na problemática em questão. Dessa forma, utiliza-se a especificação de meta para se comprovar uma meta.
- Explicar os requisitos para os envolvidos no processo é outra questão importante, pois as metas provêm a razão para os requisitos, de uma maneira similar àquela do projeto de metas, nos processos de projeto. Um requisito aparece por causa de alguma meta que fornece uma base para tal. Mais especificamente, a árvore de refinamento de metas fornece ligações de rastreabilidade dos objetivos estratégicos de “alto nível”, para os requisitos técnicos de “baixo nível”, ou seja, uma árvore divide-se em várias ramificações. Assim, uma árvore de refinamento de metas, parte de uma meta geral (por exemplo, segurança). Essa meta vai sofrendo detalhamento

em “metas menores”, que, ao final, convergem para a meta geral. Isso leva á rastreabilidade das metas do *software*. Em particular, para os *softwares* de aplicação de negócios, podem ser usadas metas para relacionar o *software* ao contexto organizacional e de negócio.

- O refinamento de metas fornece um mecanismo natural para estruturar documentos de requisitos complexos, para uma maior elegibilidade.
- Os engenheiros de requisitos deparam-se com muitas alternativas a serem consideradas durante o processo de elaboração dos requisitos. Estudos revelam que o refinamento de metas alternativo, ou seja, o detalhamento das metas em “metas menores”, fornece o nível de abstração correto, com o qual se envolvem para validar as escolhas feitas ou sugerir alternativas. Esse refinamento de metas alternativo permite propostas alternativas de *software*, a serem exploradas.
- Gerenciar os conflitos entre diversos pontos de vista, é uma outra questão importante. As metas reconhecidas fornecem as raízes que detectam conflitos entre os requisitos, resolvendo os mesmos posteriormente.
- Separar a informação estável da mais volátil, é outra importante preocupação para gerenciar a evolução dos requisitos. Um requisito representa uma maneira particular de atingir uma meta específica. Será mais provável, portanto, que o requisito evolua numa direção diferente para atingir a mesma meta. Quanto maior o nível em que a meta está, mais estável será. Conseqüentemente, diferentes versões do *software* freqüentemente compartilham um conjunto de metas de “alto nível”; o *software* atual e o futuro correspondem a refinamentos alternativos de metas comuns e, portanto, podem ser integrados num único modelo.
- As metas identificam o trajeto dos requisitos para apoiá-los juntamente com cenários. Dessa forma se estabelece a formação de um processo sistemático de elaboração de requisitos.

Geralmente as metas são mais estáveis que os requisitos. Em suma, requisitos “implementam” as metas, da mesma forma como os

programas implementam as especificações do projeto, onde uma meta é um objetivo que o *software* deve alcançar.

A identificação das metas não é necessariamente uma tarefa fácil. Às vezes estão expostas pelos envolvidos no processo ou num material preliminar disponível aos engenheiros de requisitos. Frequentemente estão implícitas, de forma que a eliciação das metas tem de ser utilizada (LAMSWEERDE, 2001).

A análise preliminar do *software* é uma fonte importante para a identificação das metas. Tal análise resulta, frequentemente, em uma lista com os problemas e deficiências que podem ser formulados de forma precisa.

Uma vez que um conjunto de metas e requisitos é obtido e validado, muitas outras metas podem ser identificadas por meio de *refinamento* e *abstração*, ou seja, por intermédio do detalhamento da meta, em consonância com uma visão mais resumida, focada nos aspectos mais importantes dessa meta, por meio das perguntas *como* e *por quê*, em relação às metas e requisitos já disponíveis, respectivamente. Outras metas podem também ser atingidas, na resolução de conflitos entre as próprias metas, ou dos obstáculos para alcançá-los.

Discute-se que os modelos de metas devam ser construídos durante os primeiros estágios do processo de ER. Uma base para esse argumento é o papel desempenhado pelas metas no processo. Quanto antes elas forem identificadas e validadas, melhor. No entanto, não implica em nenhum tipo de processo de elaboração de requisitos do tipo cascata. Como os requisitos “implementam” as metas, é inevitável na identificação de metas, a elaboração de requisitos. Porém, as metas podem, às vezes, serem identificadas bem tardiamente no processo de ER – especialmente quando as questões, *como* e *por quê*, sobre detalhes técnicos ou cenários, inicialmente são desprezadas.

Ao dar importância desnecessária às metas, os requisitos são afetados, resultando em inconsistência de tempo decorrido entre a

especificação do *software* e o seu comportamento real. Dessa forma, a falta de antecipação de circunstâncias excepcionais pode conduzir a requisitos irrealis, impossíveis ou incompletos. Como consequência, o *software* desenvolvido a partir desses requisitos resultará inevitavelmente em falhas, algumas vezes com consequências críticas ao ambiente.

Dada a formação e os vários tipos de metas, a modelagem é utilizada para dar suporte aos esquemas de raciocínio heurísticos, ou seja, baseado na experiência do próprio indivíduo, para se chegar a alguma conclusão, onde se utiliza alguma ferramenta durante a ER. As metas são geralmente modeladas por meio de *características intrínsecas*, como seus tipos e atributos, e pelas ligações com outras metas e outros elementos do modelo de requisitos.

O principal propósito da modelagem de metas e da especificação é dar suporte ao raciocínio baseado em metas para os subprocessos da ER, tais como, a elaboração, consistência e completude na checagem de requisitos, a seleção alternativa, o gerenciamento da evolução e outros. As metas também se referem a outros tipos de questões: as funcionais, associadas aos serviços a serem prestados; e as não-funcionais, associadas à qualidade de serviço, como segurança, precisão, desempenho, entre outras.

3.2. TÉCNICAS E MÉTODOS DE SUPORTE À ORIENTAÇÃO A METAS

O emprego de métodos e técnicas formais é um meio efetivo de garantir a qualidade dos requisitos, uma vez que esses métodos produzem especificações precisas, que podem ser rigorosamente validadas. A seguir discutem-se algumas técnicas que dão suporte à elaboração de requisitos baseados em metas.

3.2.1. ORIENTAÇÃO A METAS UTILIZANDO KAOS

Para a Orientação a Metas utilizando KAOS (Aquisição de Conhecimento para Especificação Automatizada) pode-se dizer que as metas são vistas como um conceito central da aquisição de requisitos. A metodologia

KAOS visa dar suporte a todo o processo na elaboração de requisitos das metas de alto nível, a fim de alcançar os requisitos, objetos e operações a serem estabelecidas pelos vários agentes, no *software* proposto. A metodologia fornece uma linguagem de especificação, um método elaborado e ferramenta de suporte (LAMSWEERDE, 2000).

Uma meta é um objetivo que o *software* proposto deve atingir e que captura um conjunto de comportamentos desejáveis do *software*. Conexões *E* relacionam uma meta a um conjunto de submetas (chamado refinamento). Isso significa, que satisfazer todas as submetas no refinamento é uma condição suficiente para satisfazer a meta. Conexões *Ou* relacionam uma meta a um conjunto alternativo de refinamentos. Isso significa que satisfazer um dos refinamentos é uma condição suficiente para satisfazer a meta. A estrutura de refinamento de meta para um *software* pode ser representada por um gráfico acíclico direto *E* (*and*) e *Ou* (*or*).

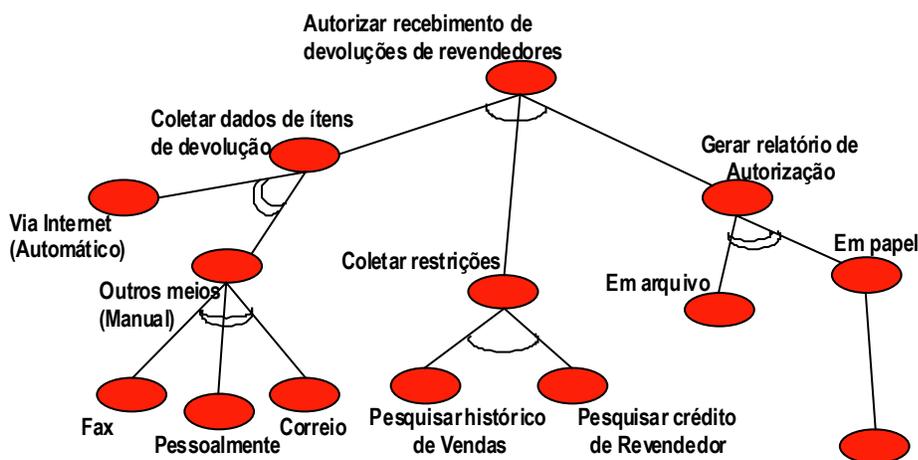


FIGURA 7 – DECOMPOSIÇÃO DE CADA META FUNCIONAL EM UMA HIERARQUIA DE RELAÇÕES E/OU (LAMSWEERDE, 2000)

A Figura 7 representa as conexões *E* (*and*) e *Ou* (*or*) de refinamento de metas. A meta principal é Autorizar recebimento de devoluções de Revendedores, onde conexões *E* (arco simples) indicam que para alcançar a meta principal todas as submetas precisam ser alcançadas, e as conexões *Ou* (arco duplo) indicam que, para alcançar a meta principal, ao menos uma submeta precisa ser alcançada.

Metas estão relacionadas a objetos aos quais estes se referem. Uma meta pode adicionalmente ser caracterizada por um atributo de prioridade, cujos valores especificam a extensão para as quais as metas são obrigatórias ou opcionais.

Os objetivos da metodologia KAOS são a modelagem formal de requisitos em termos de metas, restrições, hipóteses, objetos, ações, eventos, agentes, utilizando-se de técnicas de refinamento de metas, detecção de resolução de conflitos e atribuições de responsabilidades a agentes.

Metas, nesse contexto, são classificadas de acordo com a categoria de requisitos que conduziram sobre os agentes em questão. Dessa forma, as metas funcionais resultam em requisitos funcionais. Igualmente, metas não-funcionais resultam em requisitos não funcionais. Por exemplo, Metas Precisas são metas não funcionais que se preocupam em manter a consistência, entre o estado do objeto no ambiente e o estado de suas representações no *software*.

Assim, pode-se dizer que a metodologia KAOS preocupa-se com as metas da organização, independentemente da sua estrutura. A relação de dependência entre os atores e agentes, parte das metas para definir os relacionamentos entre seus conceitos (ações, agentes, entidades), utilizando métodos formais.

A metodologia KAOS possui passos bem definidos, levando em consideração metas particulares e a utilização de requisitos não funcionais, focando a partir da ER o processo de desenvolvimento do *software*, sem perder de vista os objetivos organizacionais.

3.2.2. ABORDAGEM CREWS-L'ECRITOIRE

A abordagem CREWS-L'Ecritoire utiliza um acoplamento bidirecional meta-cenário, permitindo a movimentação a partir de metas para cenários e vice-versa. Considerando o aspecto meta-cenário, à medida que é descoberta uma meta, esta é operacionalizada por um cenário. Utilizando o

relacionamento meta-cenário na direção reversa, ou seja, de cenário para meta, a abordagem guia o processo de elicitação de requisitos, descobrindo novas metas por meio da análise de cenários textuais (ROLLAND, 1998).

De acordo com Regnell (1996), a interação do *software* focaliza as relações entre *software* e seus usuários, necessárias para executar os serviços relacionados ao *software* no nível contextual. Regnell (1996), destaca em seu trabalho, a noção de pedaços de requisitos (RC), as estratégias de descoberta de metas por intermédio dos relacionamentos *AND*, *OR* e de refinamento entre RCs.

Dessa forma, essa abordagem propõe acoplar metas e cenários, numa decomposição “*top-down*” em diferentes níveis de abstração. Um RC é um par <meta, cenário>, integrado por meio do relacionamento *AND/OR* e de refinamento, organizado em três níveis de abstração: contextual, interação do *software* e interno do *software*, onde cada nível corresponde a um tipo de RC.

Em nível conceitual, são identificados os serviços que o *software* deve prover para uma organização e suas razões. O nível de interação do *software* focaliza as interações entre o *software* e seus usuários, necessárias para executar os serviços associados ao *software* em nível conceitual. O nível interno do *software* focaliza o que o *software* pode internamente executar. A Figura 8 apresenta o modelo de processo da abordagem CREWS-L'Ecritoire, destacando o direcionado com intenções (nós) e estratégias (setas). (ROLLAND, 1998).



FIGURA 8 – MODELO DE PROCESSO DA ABORDAGEM CREWS-L'ECRITOIRE (ROLLAND, 1998)

3.2.3. MÉTODO DE ANÁLISE DE REQUISITOS BASEADO EM METAS - GBRAM

Em Ánton (1996), é apresentado o método GBRAM (*Goal-Oriented Requirements Analysis Method*), que pressupõe que as metas não tenham sido previamente documentadas ou explicitamente elicitadas a partir dos envolvidos no processo. Nessa teoria, o analista deve trabalhar a partir de todas as fontes de informações disponíveis, como fluxos de processos ou de informação, declarações textuais de necessidades, e/ou fontes adicionais de informação como *transcripts* (cópia) de entrevistas com os envolvidos no processo, para determinar as metas do *software* desejado.

O Método GBRAM, conforme apresentado na Figura 9, envolve duas fases: a fase de análise e a de refinamento de metas, tendo como produto final o DRS (Documento de Requisitos do *Software*), que provê uma comunicação não ambígua entre envolvidos, além de apoiar a evolução e validação de requisitos.

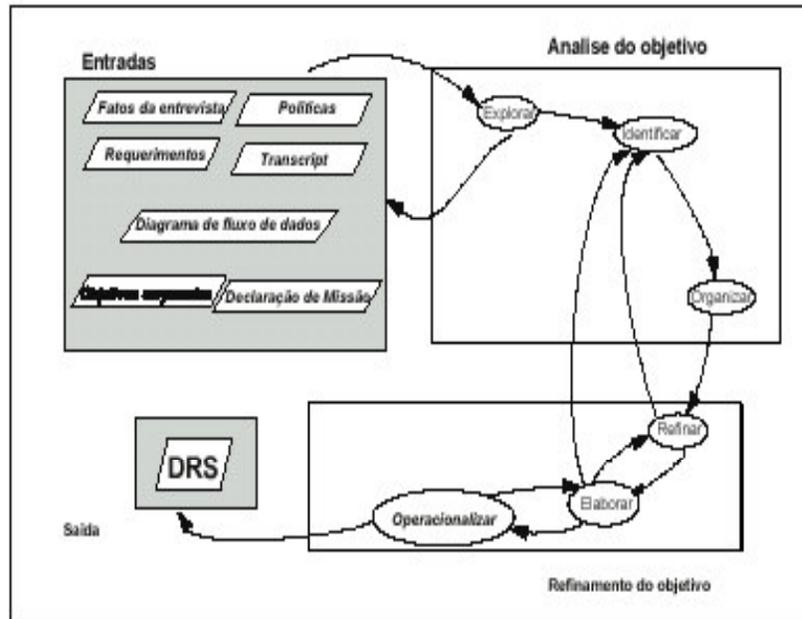


FIGURA 9 – ATIVIDADES DO MÉTODO GBRAM (ÁNTON, 1996)

As atividades de análise de metas podem ser definidas como:

1) Explorar a documentação existente para a identificação inicial de metas; 2) Identificar as metas, envolvidos e seus agentes responsáveis, 3) - Organizar as metas, de acordo com as relações de dependência e classificação das mesmas, conforme as condições alvo.

Para as atividades de refinamento de meta, estas podem ser descritas como: 1) Refinamento de um conjunto de metas com a diminuição do seu tamanho, 2) Elaboração de cenários para descobrir metas e requisitos escondidos, 3) Operacionalização das metas em requisitos operacionais.

Nesse contexto, na fase de elaboração das metas, são identificados os obstáculos que se apresentam, considerando as possíveis falhas das metas, e como estas podem ser bloqueadas. Os obstáculos são levantados através de perguntas como "de quais outras metas ou condições esta meta depende?", "pode o agente responsável pela falha da meta realizar a meta?", "pode a falha de outra meta causar completamente o bloqueio desta meta?", "se esta meta é bloqueada, quais são as conseqüências?".

Sendo especificados os obstáculos de metas, os engenheiros de requisitos devem considerar os cenários possíveis que são prováveis para

cada obstáculo. Cenários permitem levantar descrições comportamentais de um *software* e seu ambiente, facilitando uma compreensão dos problemas atuais pelos engenheiros de requisitos, o que auxilia na antecipação de situações futuras, explicando por quê é necessário e unificando critérios (RIDAO, 2000).

Na identificação de cenários, são sugeridas por esse método as seguintes perguntas: "o que acontece se esta meta não é executada?", "por que esta meta não foi executada?", "quais as circunstâncias sob as quais este obstáculo ocorre?", "por que este obstáculo ocorre?".

Concluída a especificação de metas, essa informação deve ser operacionalizada e traduzida em linguagem natural de requisitos no DRS. Operacionalizadas as ações descritas pelos envolvidos, e extraídas a partir da documentação disponível, são relacionadas de volta às metas.

Dessa forma, as metas operacionalizadas, agentes responsáveis, envolvidos no processo, restrições, obstáculos e cenários, são mapeados em ações firmadas em um conjunto de esquemas de metas (descreve os relacionamentos entre metas, e agentes em termos de eventos que causam uma mudança de estado). Esse conjunto de esquemas de metas é mapeado para um documento de requisitos de *software*, incorporando toda a informação adquirida durante a análise e elaboração das metas.

3.2.4. CASOS DE USO DIRIGIDO A METAS – GDUC

Lee e Xue (1999) propõem uma técnica que estende casos de uso com metas, de forma a possibilitar o tratamento dessas questões. Denominada GDUC (*Goal Driven Use Cases*), a técnica prevê:

- A utilização de casos de uso, para estruturar modelos de casos de uso e derivar casos de uso;
- A diferenciação entre metas flexíveis e rígidas, para o tratamento de requisitos não funcionais imprecisos;
- O registro da informação das metas nos casos de uso;

- Análise das interações entre os requisitos, por meio da investigação da relação entre metas e casos de uso.

De acordo com Mylopoulos (1999), metas representam requisitos não funcionais, decisões de projeto e argumentos a favor e contra outras metas. Decisões de projeto influenciam positiva ou negativamente uma meta particular. O processo proposto é composto de três passos:

- 1) Identificar os atores através da investigação de todos os possíveis usuários que interagem com o *software* diretamente;
- 2) Identificar as metas utilizando-se um esquema de classificação baseado em aspectos;
- 3) Construir os modelos de casos de uso.

Uma pessoa pode representar diversos papéis em relação ao *software*, sendo que para cada um dos papéis identifica-se um ator específico. Um bom ponto de partida para a identificação dos atores é analisar o *software* e a quem deve auxiliar.

A representação das metas em diagramas de casos de uso tem o propósito de estabelecer uma correlação entre as metas e os requisitos do *software*, considerando que casos de uso são conjuntos de ações que atores (entidades externa ao *software*) realizam em um *software* buscando uma meta particular.

Para cada meta modelada, um diagrama de casos de uso representa a funcionalidade do *software* (o que o *software* deve fazer?) em prol da meta. Para a identificação das metas a partir das descrições do domínio, os autores sugerem sua identificação, baseada em um esquema de classificação dos requisitos. Para essa classificação, cada meta pode ser analisada segundo três aspectos: **competência, perspectiva e conteúdo**.

- 1) Competência indica o grau de satisfação de uma meta:

- uma meta rígida, que precise ser completamente satisfeita, descreve um requisito mínimo para o *software* a ser desenvolvido.

- uma meta flexível, descreve uma propriedade desejável do *software* a ser desenvolvido e pode ser parcialmente satisfeita.

Uma meta flexível ("*soft-goal*") depende de uma meta rígida ("*rigid-goal*"), isto é, existe um certo relacionamento entre metas flexíveis e metas rígidas.

2) Perspectiva classifica as metas com relação à sua aplicabilidade, podendo ser específico do ator, ou específico do *software*. Desta forma, as metas específicas do ator referem-se aos seus objetivos com relação ao uso do *software*; as metas específicas do *software* são requisitos aplicáveis aos serviços que oferece.

3) Conteúdo classifica as metas em duas categorias: aquelas que se relacionam aos aspectos funcionais do *software*, e aquelas relacionadas a seus aspectos não funcionais. As metas funcionais são atingidas através da realização de uma seqüência de operações, e as metas não funcionais são definidas como sendo as restrições que qualificam a meta funcional relacionada.

Para a construção dos modelos de casos de uso, define-se que os mesmos são vistos como processos associados às metas que precisam satisfazer, otimizar ou manter. Esse processo de construção de modelos de casos de uso divide-se em três atividades: identificar os casos de uso, de forma a capturar os requisitos mínimos, identificar extensões de casos de uso, de forma a construir um modelo mais completo, e refinar um modelo de caso de uso, de forma a melhorar sua reusabilidade.

Para otimizar ou manter metas flexíveis, criam-se casos de uso estendidos². Dessa forma, o curso básico de um caso de uso estendido é otimizar ou manter sua meta flexível, enquanto que um curso alternativo descreve o que deve ser feito se houver falha na otimização ou manutenção da meta flexível.

² Casos de uso estendidos são concebidos para satisfazer metas originais, de forma a modelar os requisitos mínimos do usuário, levando em conta diferentes tipos de metas.

A fim de atingir uma meta específica do *software*, cria-se o caso de uso estendido, para satisfazer a metas específicas do *software* e, portanto, aplicável globalmente a quaisquer de seus atores. Para satisfazer a metas não funcionais, o caso de uso estendido serve como uma restrição aplicável ao caso de uso original. O método proposto avalia as interações entre metas em três passos:

- 1) Análise dos relacionamentos entre casos de uso e metas, por intermédio da pesquisa sobre os efeitos da ocorrência dos casos de uso nas metas;
- 2) Exploração das interações entre metas em nível dos casos de uso;
- 3) Derivação das interações entre metas em nível do *software*.

De acordo com os autores, vários benefícios podem ser obtidos com a aplicação do método proposto:

- Funciona como um mecanismo de estruturação para facilitar a derivação de especificações a partir de casos de uso;
- Preenche a lacuna entre descrições do domínio e os requisitos do *software*, isto é, as interações entre requisitos funcionais e não funcionais;
- Facilita o tratamento de requisitos flexíveis e a análise entre os diversos requisitos conflitantes.

Neste capítulo, apresentamos a análise de requisitos orientado a metas, tratando dos conceitos sobre metas e das técnicas e métodos de suporte a orientação a metas. No próximo capítulo apresentamos a Modelagem *i**, objeto central do trabalho, onde detalhamos os dois modelos que a técnica oferece.

4. TÉCNICA DE MODELAGEM I*³

O uso de técnicas de modelagem apropriadas é crucial para o sucesso no desenvolvimento de sistemas complexos. Assim, as técnicas de análises de sistemas e de projetos têm focalizado, primeiramente, a questão dos requisitos funcionais. Dado o ambiente tecnológico de hoje e considerando a necessidade de novas técnicas adequadas ao tratamento de confiabilidade, risco e vulnerabilidade, à frente do processo de análise, projeto e desenvolvimento de sistemas, elegeu-se a técnica de modelagem i*. Essa técnica torna-se recurso fundamental para o entendimento do ambiente organizacional e suas metas, tendo como foco o relacionamento estratégico entre todos os atores envolvidos (YU, 2000) (LIU, 2002).

Dentro de uma rede de atores, a modelagem i* interliga análises não funcionais com análises funcionais do sistema de informação. Os atores apresentam liberdade de ação, porém operam dentro de uma rede de relacionamentos sociais onde são expressas dependências recíprocas de um ator para outro, com o objetivo de atender as metas, cumprir as tarefas e utilizar adequadamente os recursos disponibilizados (PEDROZA, 2004).

A relação de dependência entre os atores é intencional, baseada em conceitos fundamentais, tais como meta, habilidade e compromisso, crença, etc. Os atores estabelecem ligações estratégicas, constituindo uma rede de relações que permitem avaliar os relacionamentos sociais em termos das oportunidades que oferecem e da vulnerabilidade que podem trazer (YU, 1998) (LIU, 2003).

Muitos requisitos não funcionais são difíceis de serem quantificados ou caracterizados, tal qual ocorre com critérios de flexibilidade e de escalabilidade, dentre outros. Essas qualidades não funcionais interagem mutuamente e de maneira complexa, dificultando seu explicitamento. Em tal complexidade de relações de dependências e influências mútuas, a técnica i* oferece um nível de modelagem mais elevado para que se alcancem

³ Pronuncia-se i-star: *Intentional Strategic Actor Relationships* .

configurações alternativas dos relacionamentos organizacionais em busca da melhor avaliação e análise dos requisitos (YU, 2001).

O conceito *softgoal* em *i** decorre de uma técnica estabelecida originalmente para se lidar com requisitos não funcionais em engenharia de *software*. Nesse contexto, as contribuições podem ser positivas e negativas, consideradas parciais ou suficientes para atingir uma *softgoal*. Contribuições à confiabilidade são sistematicamente elaboradas e analisadas, utilizando um sistema de raciocínio qualitativo. As qualidades não funcionais de um sistema estabelecem o nível em que estão cumpridas suas funções que têm a ver com agilidade, baixo custo e qual o nível de correção das funções executadas. Para a modelagem, a técnica *i** oferece dois modelos (YU, 1998) (YU, 2000), (WEBSTER, 2005):

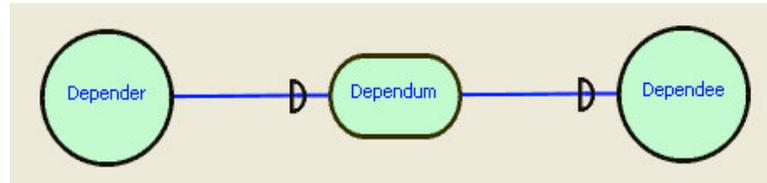
- *Strategic Dependency Model (SD)* - Modelo de Dependência Estratégica que descreve uma rede de relacionamentos entre atores.
- *Strategic Rationale Model (SR)* - Modelo Estratégico Detalhado que descreve e apóia o raciocínio que cada ator tem sobre seus relacionamentos com os demais atores de forma mais detalhada (LIU, 2003).

4.1. MODELO DE DEPENDÊNCIA ESTRATÉGICA - SD

As relações entre os atores podem ser modeladas graficamente utilizando modelos baseados em atores. O modelo de dependência estratégica representa redes de relacionamentos entre os atores do sistema, que consistem em um grupo de nós e ligações, cada nó representando um ator e cada ligação indicando a dependência entre dois atores, para que o primeiro consiga atingir alguma meta (WEBSTER, 2005).

Destaca-se a relação entre dois atores e um objeto, como mostra a Figura 10. O ator subordinado e dependente, em inglês *dependor*, o ator do qual há dependência, *dependee* e, finalmente o objeto no qual o relacionamento de dependência se concentra que é denominado *dependum*. Embora, à primeira vista tal relação mostre-se complexa, evidencia, porém, a

meta a ser conquistada, pois somente com a extinção da dependência de um ator *dependor*, por um objeto *dependum*, a ser disponibilizado pelo ator *dependee*, ocorre a realização da meta. Originalmente, não se conseguiria a realização da meta, simplesmente por decorrências de facilidades ou qualidade, pois não estavam destacados a relação de dependência e os atores



envolvidos na mesma.

FIGURA 10 – NOTAÇÃO GRÁFICA DO RELACIONAMENTO DE DEPENDÊNCIA

Estritamente dentro da relação de dependência, um ator depende do outro para realizar uma dada condição porque uma meta foi estabelecida. O ator *dependee* tem liberdade para escolher como deve atingir a meta. Porém, se falhar em concretizar o objeto – *dependum*, estará afetado adversamente em sua habilidade de concretizar suas metas.

A relação de dependência entre atores se divide em dependências de tarefa, de recurso, de meta e, finalmente, de *softgoal*, tal como as descrições abaixo:

- **Dependência de tarefa:** é estabelecida pela dependência que um ator tem do outro para que as ações sejam realizadas, a fim de que os resultados sejam atingidos. A descrição das tarefas especifica a direção das ações.
- **Dependência de recurso:** retrata a dependência de um ator sobre o outro, por meio da disponibilidade de uma entidade (recurso), a fim de atingir uma necessidade estabelecida.
- **Dependência de meta:** evidencia-se pela dependência de um ator sobre o outro para que um resultado especificado (meta) seja alcançado. Assim, a dependência de meta é definida entre os atores como sendo a entidade cujo enfoque é obter a condição para que os resultados sejam atingidos.

- **Dependência de *softgoal*:** é usado para modelar atributo de qualidade, onde não existem, a priori, critérios claros para satisfação, mas que são julgados pelos atores como sendo suficientemente atingidos.

Na construção do modelo são adotados símbolos de representação gráfica para indicar relacionamentos, atores e processos. A seguir, a Figura 11 estabelece a simbologia utilizada na técnica i* para tal representação de dependência estratégica.

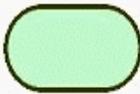
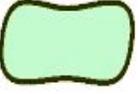
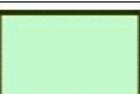
Representa Metas		Representa <i>Softgoal</i>	
Representa Tarefas		Representa Atores	
Representa Recursos		Representa ligação de dependência	

FIGURA 11 – SIMBOLOGIA PARA O MODELO DE DEPENDÊNCIA

Apresenta-se, a seguir, a Figura 12, que demonstra os mediadores aplicáveis para as relações de dependência, conforme identificados de forma distinta na Figura 11.

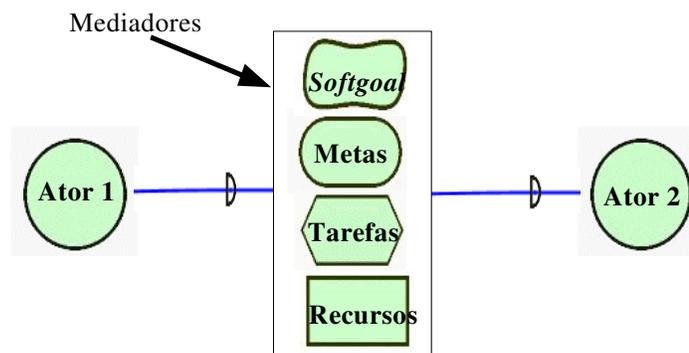


FIGURA 12 – MEDIADORES NAS RELAÇÕES ENTRE ATORES

4.2. RELACIONAMENTOS COMPLEXOS ENTRE ATORES

Na técnica i^* o termo ator é usado genericamente para se referir a qualquer unidade a qual dependências intencionais possam ser atribuídas. O modelo de dependência estratégica pode ser estendido, refinando a noção do ator, em noções de funções, posições e agentes, cada um deles constituindo-se em ator num sentido mais especializado (YU, 2000).

Um **agente** é um ator com manifestações físicas e concretas. Tais como manifestadas por um ser humano, tem dependências que se aplicam independente da função em que esteja atuando. O termo agente é aplicado, preferencialmente à pessoa, por generalização, mas se refere tanto a humanos como a agentes artificiais (*hardware, software*).

Uma **função** é uma caracterização abstrata do comportamento de um ator social, ou seja, as dependências são associadas a uma função, quando estas se aplicam independentemente de quem atua na função.

A **posição** é intermediária, está entre a função e o agente, em termos abstratos. Representa um grupo de funções normalmente realizadas por um agente. As posições podem cobrir funções, agentes podem ocupar posições e podem também atuar em funções diretamente.

4.3. MODELO ESTRATÉGICO DETALHADO - SR

O Modelo Estratégico Detalhado, como sugere a nomenclatura, esclarece detalhes dos pontos estabelecidos nos modelos, focando nos atores para modelar relacionamentos internos intencionais. Os elementos intencionais (metas, tarefas, recursos e softgoal) aparecem nesse modelo não somente como dependências externas, mas também como elementos internos, organizados numa hierarquia de relacionamento de meio-fim e de decomposição de tarefas (YU, 2000) (WEBSTER, 2005).

Quando um elemento é expresso como uma meta, significa que pode haver várias alternativas, por meio das quais se pode cumpri-la, ou seja, por meio da decomposição de subtarefas, por exemplo. Uma tarefa

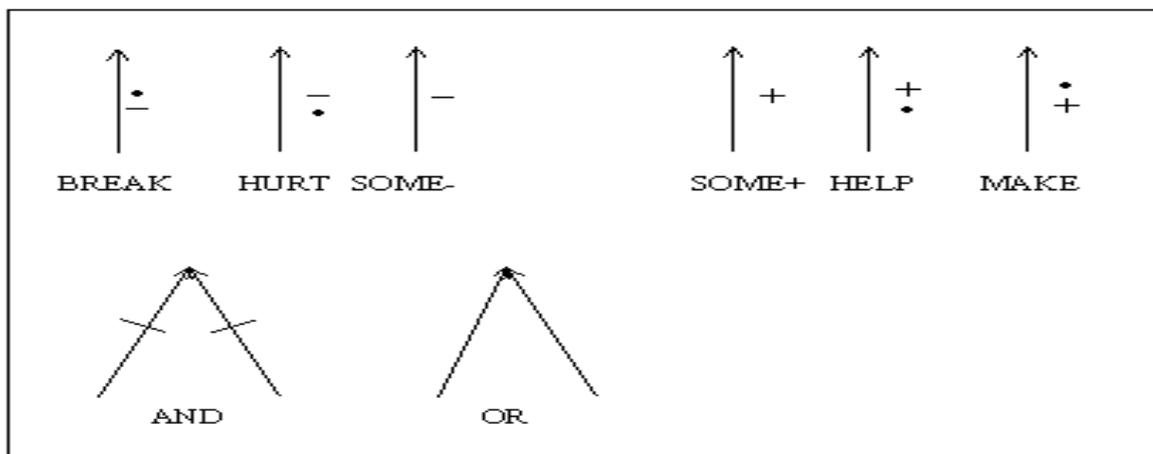
especifica uma maneira particular de fazer as coisas em termos de decomposição de submetas, recursos e *softgoal*.

Nesse modelo, para se satisfazer uma *softgoal* outras formas detalhadas de expressar a sua condição são identificadas no Quadro 1, como contribuições positivas ou contribuições negativas, utilizando-se algumas palavras-chave para sua representação e denominação.

Contribuições Positivas	Contribuições Negativas
HELP – a contribuição é positiva, porém não suficiente.	HURT – a contribuição é negativa, mas não suficiente.
MAKE - a contribuição é positiva e suficiente.	BREAK – a contribuição é negativa e suficiente.
SOME⁺ – a contribuição é positiva, mas a extensão da contribuição é desconhecida.	SOME - a contribuição é negativa, mas a extensão da contribuição é desconhecida.

QUADRO 1 – PALAVRAS CHAVE PARA EXPRESSAR *SOFTGOAL*

Têm-se ainda, no Quadro 2, as palavras, *AND* (significando que se todas as submetas são atingidas, a meta principal é suficientemente atingida) e *OR* (significando que a meta superior será suficientemente atingida, se quaisquer das suas submetas forem atingidas).



QUADRO 2 – SIMBOLOGIA DAS CONTRIBUIÇÕES POSITIVAS E NEGATIVAS

No presente capítulo apresentamos a Modelagem *i**, onde detalhamos a definição e aplicação dos dois modelos que a técnica oferece.

Para que seja possível apresentar resultados contundentes e contribuir efetivamente com o presente estudo, cujo foco é apresentar um método prático para a aplicação da técnica i^* , a seguir no Capítulo 5, desenvolver-se-á o estudo de caso.

5. CONTEXTUALIZAÇÃO DO ESTUDO DE CASO

A proposta que se apresenta está centrada num ambiente distribuído, cuja integração de funcionalidades na comunicação e eficiência é o fator principal. Trata-se do desenvolvimento de um *software* para gerenciamento e organização de um simpósio, destacando suas atividades e integrações no contexto da ER, onde o universo de informações e atividades, com prazos determinados, têm de ser eficazmente atendidos e atualizados para otimizar o gerenciamento do evento. A organização de um simpósio apresenta uma multiplicidade de atores e ações complexas, portanto consideramos ser representativa para o estudo de caso, sobre o qual dispúnhamos de confiável fonte de informação.

A abordagem é orientada a metas, que se destaca pela definição dos *porquês* do sistema e identificação dos requisitos, considerando a dependência de tarefas existentes, que precisam ser realizadas para a execução de um certo objetivo (ÁNTON, 1996). Elegeu-se a técnica de modelagem *i**, que traz modelos de dependência como mais apropriada para modelagem das atividades desse sistema.

No desenvolvimento desse estudo de caso, teve-se como foco avaliar as dificuldades e facilidades encontradas em se adotar *i** como técnica de modelagem de requisitos orientada a metas. Tem-se, ainda, motivação adicional para a escolha do tema, ao propor-se um guia de orientação para a aplicação do *i**, durante a análise de requisitos, de forma a garantir clareza na atuação e identificação dos atores envolvidos, estabelecendo a rede de ligações dos processos e relacionamentos entre os mesmos, considerando complementarmente os desdobramentos e refinamentos do processo.

Para a modelagem dos diagramas de dependência, utilizou-se a ferramenta OME⁴ (*Organizational Modeling Environment*), que se constitui numa ferramenta de análises, estabelecida na abordagem orientada a metas e desenvolvida pelo Laboratório de Gerência do Conhecimento da Universidade

⁴ <http://www.cs.toronto.edu/km/ome/>

de Toronto. Atualmente a ferramenta OME apóia modelagens i*, NFR- *Non Functional Requirements* e GRL *Goal-oriented Requirement Language*.

Implementando por meio da modelagem, a partir da técnica i*, houve o direcionamento a dois objetivos iniciais - distintos e complementares na aplicação da técnica:

- Identificação de relacionamentos decorrentes de uma rede de relacionamentos entre atores. O recurso técnico para tais identificações decorre da aplicação do Modelo de Dependência Estratégica (*Strategic Dependency Model - SD*).
- Identificação e descrição para fomentar uma base de sustentação ao raciocínio que cada ator tem sobre seus relacionamentos com os demais atores, de forma detalhada. O Modelo Estratégico Detalhado (*Strategic Rationale Model – SR*) cumpre tal necessidade.

5.1. MODELO DE DEPENDÊNCIA ESTRATÉGICA - SD

Assumindo o Modelo de Dependência Estratégica, baseado nos papéis e objetivos oriundos das relações de dependência entre atores, a primeira atividade realizada foi identificar os atores que atuariam no contexto.

Partiu-se de informações coletadas por meio de entrevistas com o coordenador do evento, identificaram-se diversos atores e quais as atividades a serem realizadas pelos mesmos. Prosseguindo na metodologia, procedeu-se a construção de uma matriz para estabelecer as relações de dependências, papéis e objetivos oriundos dessas relações. O Quadro 3 apresenta tal matriz que expressa formalmente as relações entre os atores identificados no sistema do simpósio:

1. Coordenador,
2. *Software* do simpósio,
3. Avaliador,
4. Participante ouvinte,
5. Participante autor,
6. Banco.

ATORES	COORDENADOR	SOFTWARE DO SIMPÓSIO	AVALIADOR	PARTICIPANTE OUVINTE	PARTICIPANTE AUTOR	BANCO
COORDENADOR		⁵ (T) Controlar recebimento dos artigos(Coord/Sof) (M) Divulgação dos resultados dos artigos (Coord/Sof)	(S) Manter bom cadastro (Coord/Aval)		(T) Enviar resultado dos artigos (Coord/Aut)	(T) Verificar pgto das inscrições (Coord/Bco)
SOFTWARE DO SIMPÓSIO	(M) Encerramento das inscrições (Soft/Coord)		(T) Distribuir artigos (Soft/Aval)	(T) Gerar boleto (Soft/Ouv) (T) Imprimir boleto (Soft/Ouv) (T) Gerar código partic. (Soft/Ouv) (M) Inscrição aceita (Soft/Ouv)	(T) Gerar boleto (Soft/Aut) (T) Imprimir boleto (Soft/Aut) (T) Gerar código partic. (Soft/Aut) (M) Inscrição aceita (Soft/Aut)	
AVALIADOR		(T) Informar notas dos artigos (Aval/Soft)			(S) Comentários dos artigos (Aval/Aut)	
PARTICIPANTE OUVINTE		(T) fazer inscrição (Ouv/Soft)				
PARTICIPANTE AUTOR		(T) fazer inscrição (Aut/Soft) (T) Submeter artigo (Aut/Soft)				
BANCO						

QUADRO 3 – MATRIZ DAS RELAÇÕES DE DEPENDÊNCIA PARA SISTEMA DO SIMPÓSIO

Em decorrência da relação dos atores e de suas atividades, de forma geral, classificam-se metas, recursos, tarefas e as *softgoal*. Assim, determinada a relação entre atores, iniciou-se a descrição das atividades a fim de classificar os mediadores no contexto do *software*.

Nesse modelo, a relação entre os atores se estabelece por meio de alguma atividade de ligação da qual decorre, também, a relação de dependência. Essa estrutura de relacionamentos configura-se como o elemento essencial em todo o processo de análise de requisitos, repetindo-se, enquanto estrutura. É identificável, nas micro-relações, aquelas mais básicas e fundamentais e, organizadas num processo complexo, constituem as macro-relações que são resultantes de inúmeros processos de micro-relações. Utilizou-se o recurso de linguagem para melhor expressar esse conceito, por exemplo: um tijolo, peça básica numa construção civil que, reunidos a inúmeros outros tijolos, pode assumir formas arquitetônicas distintas e complexas.

⁵ (T) Tarefa, (M) Meta, (S) *Softgoal*, (Ator que depende/Ator que satisfaz a dependência).

Na Figura 13, o modelo de dependência estratégica descreve as relações de dependência entre os seis atores (participante ouvinte; participante autor; coordenador; avaliador; *software* do simpósio e o banco) que estão intimamente ligados. O ator “Participante Ouvinte” é dependente do ator “*Software* do Simpósio” para alcançar a meta “Inscrição Aceita”, o ator “Coordenador” é dependente do “*Software* do Simpósio” para realizar a tarefa “Controlar recebimento de Artigos”, na meta “Divulgação dos resultados dos artigos”. Nesse modelo a relação de dependência ocorre entre ator–atividade–ator.

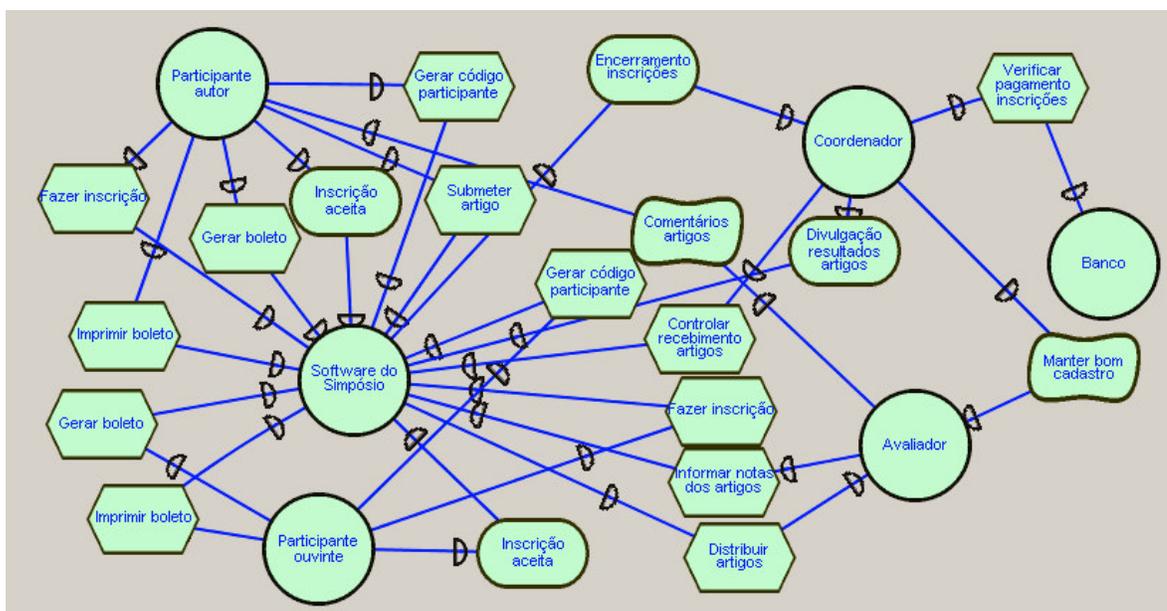


FIGURA 13 – MODELO DE DEPENDÊNCIA ESTRATÉGICA PARA SISTEMA DE SIMPÓSIO

Modelou-se nessa etapa todas as atividades identificadas na matriz e cuja principal função é estabelecer a relação de dependência entre os atores, independentemente das atividades, se “metas, tarefas, recursos ou *softgoal*”.

5.2. MODELO ESTRATÉGICO DETALHADO - SR

Esse modelo descreve de forma mais detalhada o modelo anterior, apresentando as decomposições dos elementos (metas, tarefas, recursos e *softgoal*), de acordo com os relacionamentos internos organizados numa hierarquia de relacionamentos de meio-fim e de tarefa-decomposição. Ao

se estabelecer uma “meta”, significa que várias alternativas para cumpri-la por meio da decomposição podem ocorrer, como exemplo por “subtarefas”.

A Figura 14 mostra o ator “Avaliador”. Nesse caso a meta “Nota dos Artigos” pode ser alcançada pela contribuição de dependência das tarefas “Listar Artigos para Avaliação”; “Inserir Notas dos Artigos” e “Incluir Comentários”. Contudo para cumprir a tarefa “Inserir Notas dos Artigos” é necessário uma decomposição de tarefas “Informar senha para acesso” e “Informar e-mail” para que assim seja alcançada a meta.

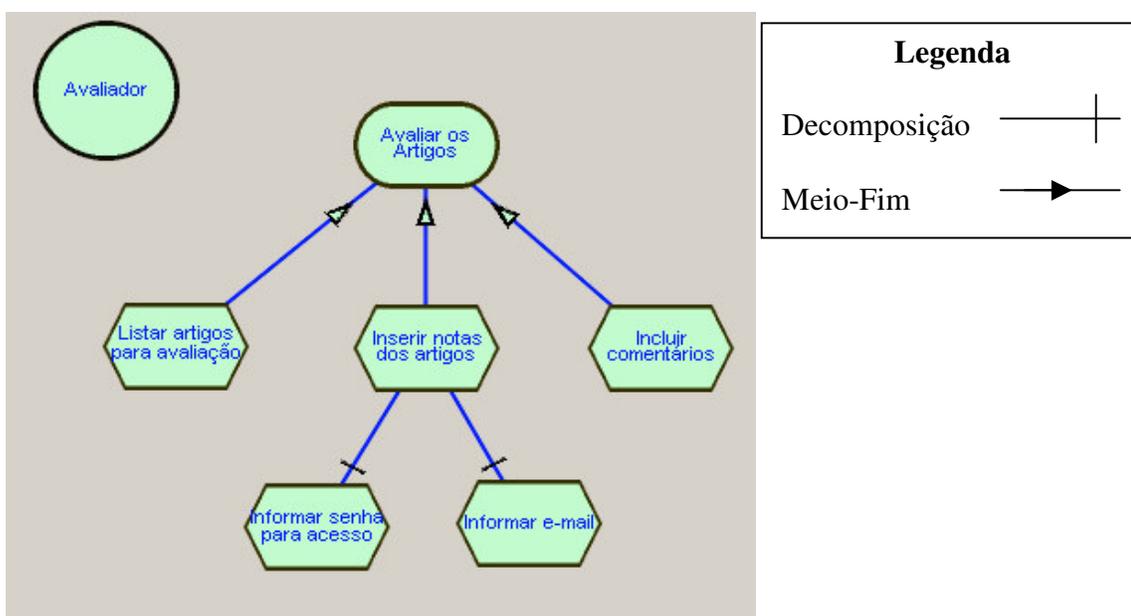


FIGURA 14 – MODELO ESTRATÉGICO DETALHADO PARA O ATOR “AVALIADOR”

Na Figura 15 o ator “Participante Autor”, tem como meta “Fazer Inscrição” que é realizada pelas tarefas e subtarefas:

- “Preencher Formulário”, ao mesmo tempo tem que “gerar código do participante”, sendo assim uma sub tarefa;
- Para contribuir com a realização da meta, a tarefa “Imprimir Boleto” é decomposta pelas subtarefas “Confirmar Inscrição após pagamento do Boleto” e “Rejeitar Inscrição após 7 dias pelo não Pagamento”;

- A tarefa “Alterar dados da Inscrição” só é possível por meio da realização da subtarefa “Informar Senha”,
- Assim a tarefa “Submeter Artigo” contribui para atingir a meta “Fazer Inscrição”, pois no ato da inscrição deve-se submeter o artigo.

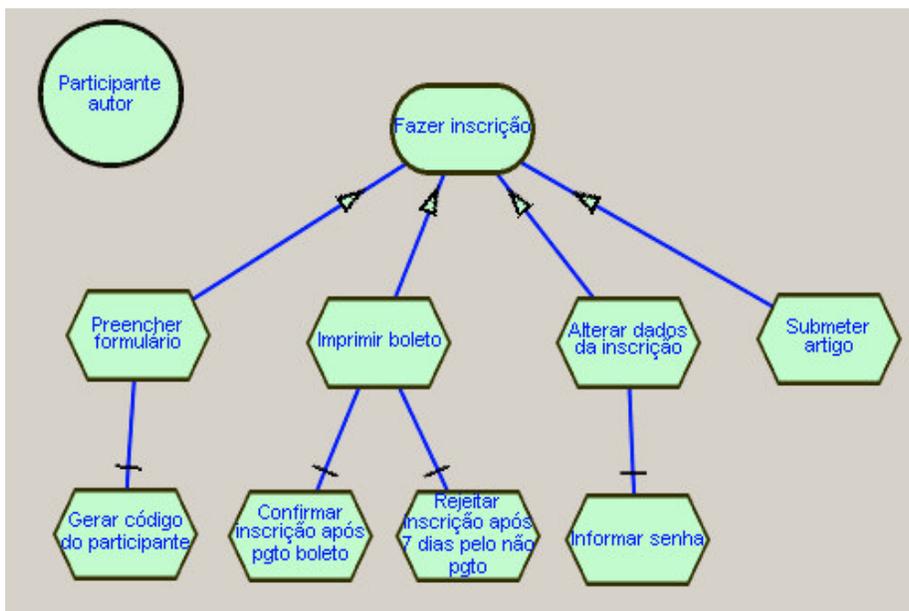


FIGURA 15 – MODELO ESTRATÉGICO DETALHADO PARA O ATOR “PARTICIPANTE AUTOR”

Na Figura 16 o ator “Participante Ouvinte”, tem como meta “Fazer Inscrição” que é realizada pelas tarefas e subtarefas:

- “Preencher Formulário”, ao mesmo tempo tem que “gerar código do participante”, sendo assim uma subtarefa;
- Para contribuir com a realização da meta a tarefa “Imprimir Boleto” é decomposta pelas subtarefas “Confirmar Inscrição após pagamento do Boleto” e “Rejeitar Inscrição após 7 dias pelo não Pagamento”;

- A tarefa “Alterar dados da Inscrição” só é possível por meio da realização da subtarefa “Informar Senha”.

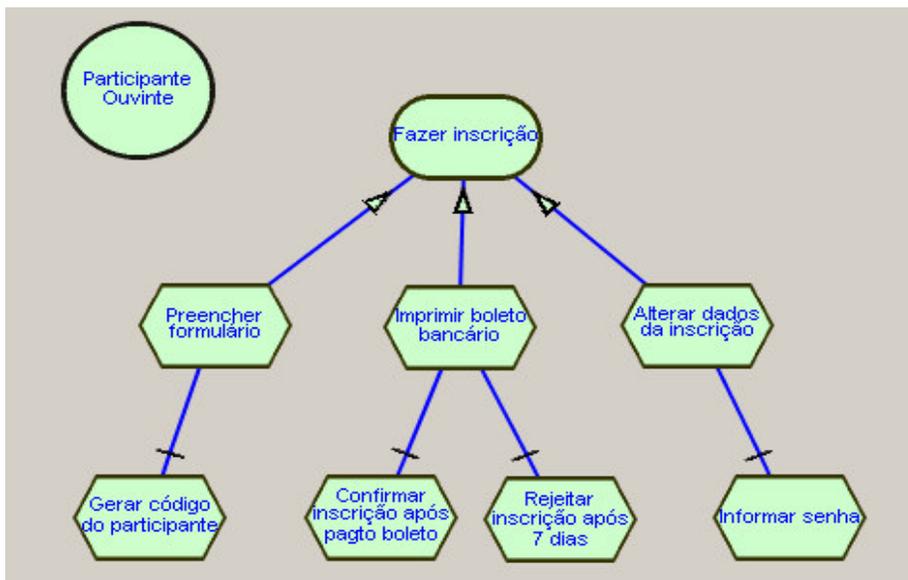


FIGURA 16 – MODELO ESTRATÉGICO DETALHADO PARA O ATOR PARTICIPANTE OUVINTE

Ilustra-se na Figura 17 o ator “*Software* do Simpósio”, que tem como meta a “Administração das Transações do Simpósio”. Para realizar essa meta, a tarefa “Controlar recebimento dos Artigos” só pode ser atingida se as submetas “Relatar a lista dos artigos ao Coordenador” e “Encaminhar artigos aos avaliadores” forem cumpridas. O sistema deve fazer uma distribuição dos artigos entre os avaliadores cadastrados no sistema e relatar ao coordenador a lista dessa distribuição. Para a tarefa “Informar ao Coordenador”, duas subtarefas foram identificadas “Relação de Inscritos” e “Encerramento das Inscrições”. Nesse momento o sistema deve gerar, após a data de encerramento das inscrições, a relação dos inscritos, assim o coordenador terá uma fonte de dados para extrair quaisquer comparações a fim de subsidiar as suas atividades para a elaboração do evento. Para a tarefa “Divulgar os Resultados dos Artigos”, a subtarefa “Via e-mail encaminhar Notas e Comentários aos Autores”, permite que o Coordenador, num dado momento, acione a opção divulgação dos resultados, que fará com que o sistema envie um e-mail para os autores, notificando-os sobre a aceitação ou rejeição dos artigos, assim como envie a nota final do artigo e os comentários realizados pelos avaliadores sobre os artigos.

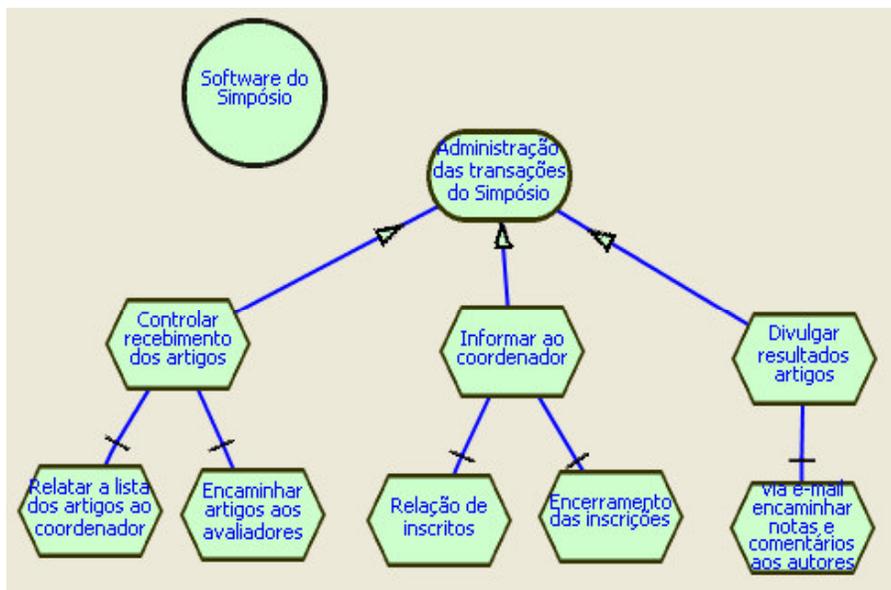


FIGURA 17 – MODELO ESTRATÉGICO DETALHADO DO ATOR “SOFTWARE DO SIMPÓSIO”

Na Figura 18, o ator “Coordenador” tem como meta “Gerenciar o Simpósio” estando dependendo das tarefas “Criar Categorias para Participantes”, “Manter Cadastro de Avaliadores”, “Divulgar Resultados dos Artigos”, estas sem nenhuma subtarefa a realizar. Mas ainda, a tarefa “Acessar Cadastro dos Participantes” decompõe-se nas subtarefas “Emitir crachás”, “Emitir Lista de Presença” e “Emitir Certificado”, que se decompõe em uma subtarefa “Incluir texto para o Certificado”. O sistema permitirá, assim, que o Coordenador crie quantas categorias de participantes achar necessário constando nome da categoria e valor correspondente. Acessando os dados dos participantes pode emitir crachás, gerar lista de presença e certificado de participação do evento. A manutenção de um cadastro de avaliadores deve ser mantida pelo Coordenador onde os dados nome do avaliador, senha, instituição, país, e-mail devem ser alimentados pelo mesmo para consultas e relatórios.

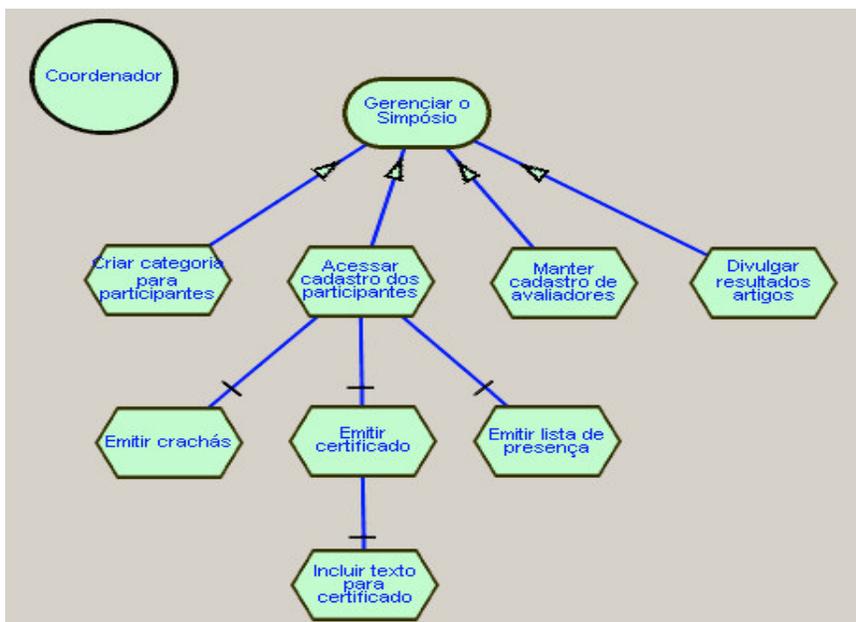


FIGURA 18 – MODELO ESTRATÉGICO DETALHADO DO ATOR “COORDENADOR”

Na Figura 19 o ator “Banco” tem como meta principal “Verificar pagto das inscrições”. Para essa relação existe o link de contribuição para a tarefa “Acessar sistema do Banco”, que se decompõe em duas subtarefas “Informar senha” para acesso às informações e a subtarefa “Obter extrato para Conferência”.

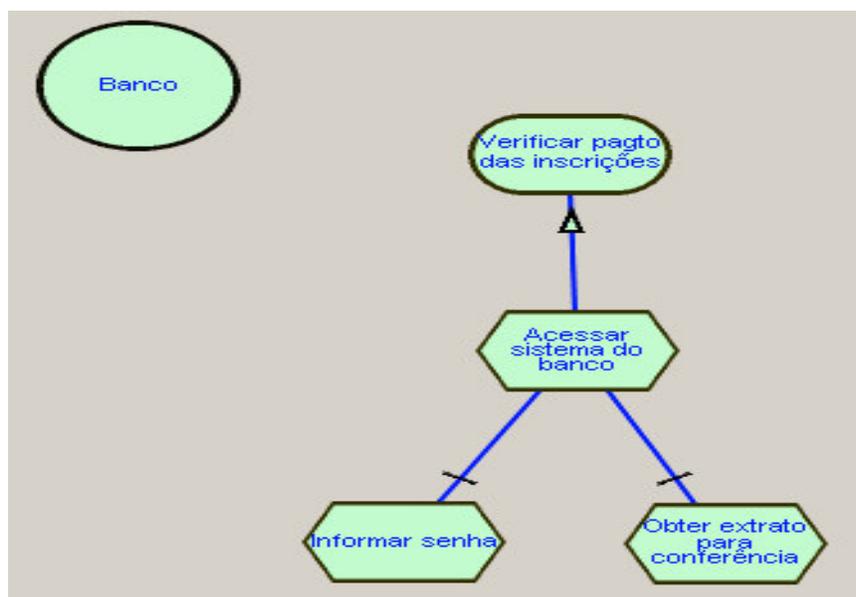


FIGURA19 – MODELO ESTRATÉGICO DETALHADO DO ATOR “BANCO”

Para se obter um nível ainda mais detalhado, os relacionamentos internos, metas, tarefas, recursos e *softgoal* são organizados

numa hierarquia de relacionamentos de meio-fim e de tarefa-decomposição. Esse modelo traz também as relações de dependências externas.

Durante a análise e projeto do sistema relataram-se novos elementos (metas, recursos, *softgoal*) que foram sistematicamente refinados até que sejam operacionalizados e implementados.

Na Figura 20 demonstra-se a estrutura unificada de cada ator, apresentada individualmente nas figuras anteriores deste capítulo, onde se nota que não se trata de uma estrutura de árvore rígida. Elaboraram-se os relacionamentos entre os seis atores: coordenador, *software* do simpósio, avaliador, participante ouvinte, participante autor e banco. Por exemplo, “Coordenador” tem uma meta interna “Gerenciar o Simpósio”. Quando um elemento é expresso como uma meta, significa que pode haver várias alternativas por meio das quais se pode cumpri-la. O “Coordenador” pode “Criar categorias para os participantes”, “Acessar cadastro dos Participantes”, “Manter cadastro dos avaliadores” ou ainda “Divulgar resultados dos artigos”. Estas são representadas como tarefas. Uma tarefa especifica uma maneira particular de agir, em termos de decomposição de subtarefas, submetas, recursos e *softgoal*.

Esse modelo estratégico detalhado (SR) permite descobrir, incluir novas atividades e relacionamentos, os quais retratem as necessidades dos usuários e a posterior implementação desses requisitos, sendo, portanto, elemento fundamental para o sucesso na construção do *software*.

O SR proporciona uma forma mais abrangente de relacionar os atores, as atividades e suas necessidades, pois permite constatar novos recursos e *softgoal*, que traduzem mais eficazmente o que o usuário, num primeiro momento, não enxerga sobre as necessidades para a construção do *software*.

A construção do modelo SR, a partir da literatura pesquisada, não prescreve como este deve ser construído, ou seja, não possui um processo e regras de como estabelecer as relações e inclusões de atividades

entre os atores, apenas mostra a possibilidade de refinar as atividades e as necessidades que se apresentam, nesse momento, para atingir sua meta.

Para a modelagem podem-se inserir os recursos necessários a fim de se atingirem as metas e cumprirem as tarefas, como mostra a Figura 20 que identifica a necessidade do recurso “Internet” como fundamental à realização das ações para a participação no evento, pois provê a ferramenta para que, de qualquer lugar e a qualquer momento, os interessados possam fazer parte desse evento. Destacamos o “*Software*” e “Banco” que desempenham dois papéis nesse sistema: o de ator, que estabelece uma relação de dependência com os demais atores, e também o de recurso, para que as ações dos atores sejam efetivadas. Novas metas são identificadas com o enfoque de obter a condição para que a meta principal seja atingida.

Neste estudo de caso enfrentamos certas dificuldades para a aplicação da modelagem i^* . Durante a experiência identificamos o lapso entre as afirmações teóricas e sua implementação prática. Assim, as dificuldades e o desenrolar das soluções para as mesmas foram transformadas na contribuição que é desenvolvida no capítulo a seguir.

No presente capítulo apresentamos a aplicação prática do modelo proposto, consideramos uma situação real, o gerenciamento e organização de um simpósio como base para esta aplicação da técnica. No próximo capítulo é feita análise da aplicação da técnica i^* e são apresentados os benefícios da aplicação, as dificuldades encontradas e um relato da contribuição para a pesquisa que consiste na elaboração de um guia de utilização do i^* .

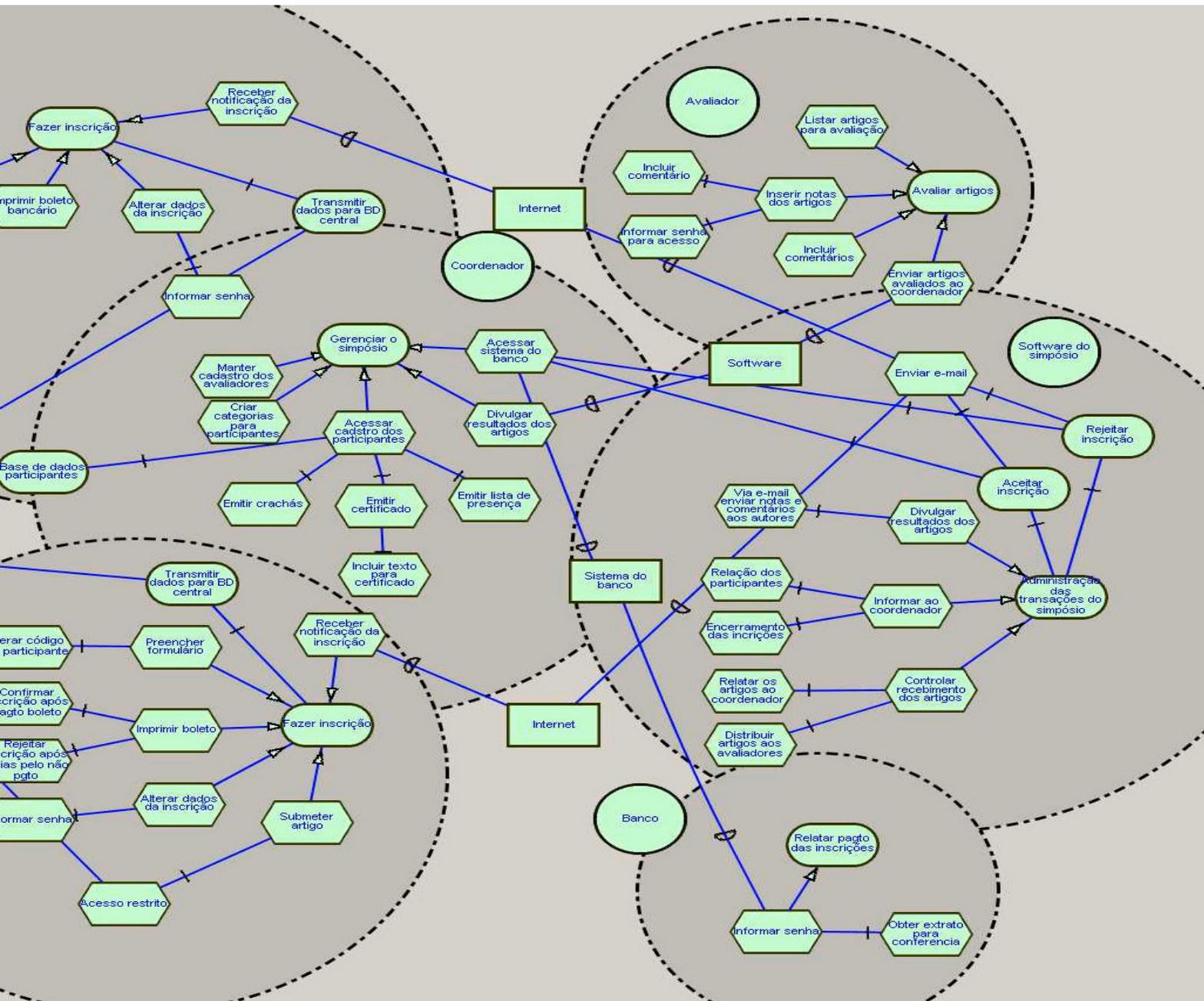


FIGURA 20 – MODELO ESTRATÉGICO DETALHADO DO SISTEMA DE SIMPÓSIO

6. CONTRIBUIÇÕES PARA UTILIZAÇÃO DA TÉCNICA I*

Nas primeiras incursões do presente estudo de caso, foi possível que se tomasse ciência das dificuldades envolvidas no trabalho do Engenheiro de Requisitos. Um trabalho de extrema importância que não se desenvolve apoiando-se exclusivamente em ferramentas automatizadas. De fato, a experiência profissional é fator diferencial no sucesso da eliciação de requisitos, que se expressa em atitudes flexíveis frente às dificuldades imediatas para a identificação dos dados de entrada.

Assim, entendeu-se ser uma contribuição relevante a elaboração de um procedimento, tal como um guia, para dissolver as dificuldades na aplicação da técnica i*. A partir da experiência de uso da técnica no estudo de caso apresentado no Capítulo 6, pôde-se elaborar um procedimento heurístico, a medida em que se buscou o apoio em métodos que não são tão rigorosos, mas que expressam o conhecimento humano, permitindo um encaminhamento satisfatório.

No capítulo 2 já se destacou que o fator humano no desenvolvimento de um *software* é algo existente no processo de ER, apresentando características de volatilidade não previsíveis e complexas. Até aquele momento apresentaram-se posições elaboradas a partir de contribuições teóricas que, embora estivessem fundamentadas experimentalmente, não faziam parte da vivência pessoal que se obtivera até então.

6.1. FLUXOGRAMA DE APLICAÇÃO DA TÉCNICA I*

A eliciação de requisitos é um processo “extrair de”, que no estudo de caso significou extrair, a princípio, do usuário, as informações para prover maior facilidade de compreensão acerca da construção do sistema. Entretanto um sistema é composto de diferentes características e entendê-las em conjunto é igualmente importante. Se por um lado é preciso separar as características por outro é preciso compô-las.

Assim, para diminuir essa complexidade, permitindo o entendimento e análise de uma única característica por vez, estabeleceu-se um fluxograma (Figura 21), que traz as seis etapas identificadas durante a aplicação da técnica i*:

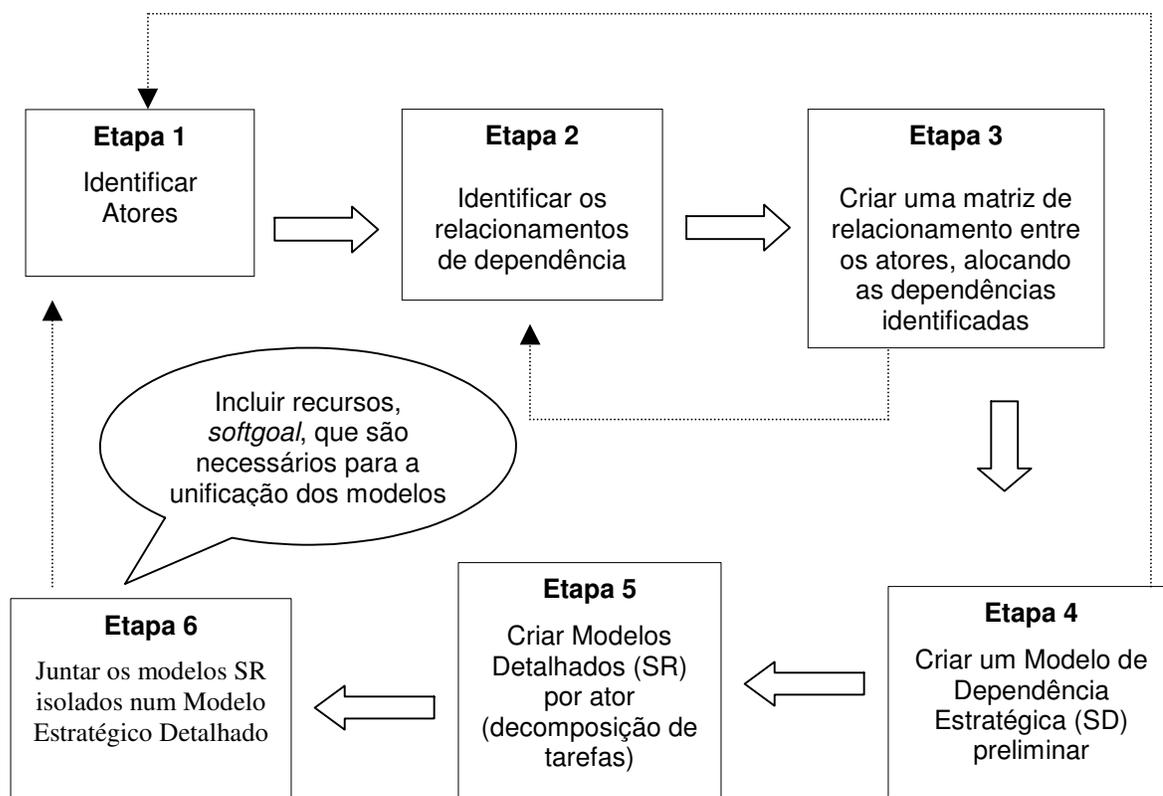


FIGURA 21 – FLUXOGRAMA DAS ETAPAS PARA APLICAÇÃO DA TÉCNICA I*

Esse fluxograma representa as etapas que são realizadas continuamente, ou seja, ao caminhar para a finalização da etapa, simultaneamente está se iniciando a próxima etapa. Entretanto, para efeitos de representação gráfica há uma separação que, também no procedimento prático, permite que se avalie e re-elabore a mesma.

Na etapa 3, para a construção da matriz identificamos que é preciso retornar à etapa 2 para esclarecer e/ou identificar novas relações de dependência. Na etapa 4, é necessária a reelaboração a partir da etapa 1, pois, ao modelar o SD podemos encontrar novos atores e/ou mediadores. Para unificar os modelos SR na etapa 6, é necessário retornar à etapa 1, pois neste momento, para se configurar a unificação, serão demandados novos mediadores (*softgoal*, metas, recursos, tarefas) associados aos atores.

Com tal seqüência de ações, pretende-se atingir plenamente os objetivos demandados pelos usuários e das áreas envolvidas na organização. Porém, deve-se estabelecer uma linha objetiva junto às necessidades do usuário e desenvolver o sistema enquanto processo viável, segundo critérios de compreensão da realidade e dos requisitos.

Vale ressaltar que o Engenheiro de Requisitos deve realizar ensaios preliminares, ao cumprir cada etapa, a fim de identificar e conferir as relações de dependência e tarefas a serem descritas na aplicação da técnica *i**, indo um pouco mais além na elicitacão dos requisitos. A seguir, descrevemos as seis etapas apresentadas no fluxograma, Figura 21.

6.1.1. IDENTIFICAR ATORES

A identificação dos atores é o passo inicial. Cumpre lembrar que os atores podem ter, ou ser, parte de um todo, assumindo o papel de pessoas, funções, hardware, *software*, etc. A identificação se dá através de entrevistas, leitura de documentos e inspeção visual nas áreas envolvidas. Um ator é uma entidade ativa relacionada com uma dependência, a qual satisfaz para outrem ou é satisfeita por outro. Saliente-se que, para se definir quem são e o que são atores, no contexto de *i**, deve-se explorar as relações de dependências intencionais que possam ser atribuídas ao termo ator.

Uma forma prática de identificar um ator é inicialmente considerar as entidades *e*, dentre estas, distinguir quais se relacionam e são mediadas por alguma função. Se essa condição existir, identifica-se um ator, mas se a entidade não for mediada, poderá ser o mediador por meio do qual poder-se-ão identificar ou, até mesmo, criar os atores necessários. Ainda, é importante que se destaque que um ator não se relaciona, necessariamente, com todos os demais, exigindo apenas uma única relação para sua existência.

No primeiro momento de elicitacão de requisitos os recursos de informática estarão ausentes, é imprescindível atencão à fala das pessoas, além de registro metódico de informações para o subsídio de análises e validações posteriores. Quanto mais complexo for o sistema, recomendam-se

algumas repetições dessa etapa para a construção consistente e, naturalmente, para a correta compreensão dessa demanda.

6.1.2. IDENTIFICAR OS RELACIONAMENTOS DE DEPENDÊNCIA

No capítulo 4, indica-se a estrutura básica de relacionamento entre atores através de um mediador. Com tal estrutura, o Engenheiro de Requisitos deve mapear cada relação distinta entre atores e indicar o mediador aplicável, sejam *softgoal*, metas, tarefas ou recursos, conforme apresentado na Figura 22.

Estando claros os atores e as atividades por estes realizadas, o Engenheiro de Requisitos poderá avançar e definir as relações de dependência entre todos os atores identificados, ou seja, aquele que é o *dependee* e o que é o *depender*.

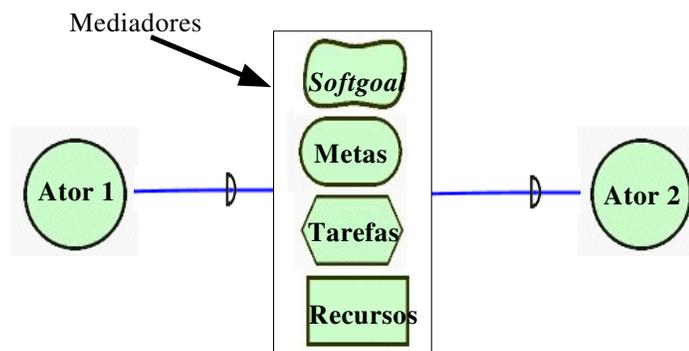


FIGURA 22 – MEDIADORES NAS RELAÇÕES ENTRE ATORES

6.1.3. CRIAR UMA MATRIZ DE RELACIONAMENTO ENTRE OS ATORES, ALOCANDO AS DEPENDÊNCIAS IDENTIFICADAS.

A Matriz de relacionamentos deve avaliar a correlação entre todos os atores e mediadores identificados. Assim, podem ser destacadas as relações de dependência que foram caracterizadas como metas a serem alcançadas, as tarefas a serem realizadas, os recursos a serem disponibilizados, ou *softgoal* identificadas a fim de satisfazer necessidades estabelecidas pelos atores. Considerando a matriz, na intersecção entre dois

atores haverá um mediador apenas quando existir o relacionamento entre os mesmos.

A verificação de atendimento ao que é demandado através do mediador proporciona a avaliação de qualidade. Portanto, o atributo de qualidade é definido pela ação efetivada num mediador e, se for verificável e cumpriu-se a exigência, então se caminha para o desenvolvimento satisfatório do sistema. Porém, se ocorrer um impedimento, isso denotará um problema a ser resolvido e que poderá estar relacionado ao mediador, aos atores envolvidos, ambos e, até mesmo, com outros atores ou mediadores que até então não foram identificados.

Vale salientar que às vezes identifica-se um mediador e não se identificam os atores que se relacionam por meio desse mediador. Toda entidade significará um ator ou um mediador que devem estar representados num relacionamento. Com a avaliação desses relacionamentos, pode-se identificar a ausência de atores ou mediadores que deverão ser construídos.

A seguir, apresenta-se no Quadro 4, um exemplo de matriz das relações de dependência. Nesse exemplo, de uma construção civil, apresentam-se os quatro atores identificados (o proprietário da casa, o engenheiro, o pedreiro e o comerciante de materiais de construção). Estabelecem-se as relações de dependência entre os atores e identificam-se os mediadores destes relacionamentos. Por exemplo, o ator “Engenheiro” estabelece a relação de dependência com o ator “Dono da Construção” por meio do mediador estabelecido como M (M = meta), que é “Fazer a planta da construção”.

ATORES	PROPRIETÁRIO DA CASA	ENGENHEIRO	PEDREIRO	COMERCIANTE MATERIAL DE CONSTRUÇÃO
PROPRIETÁRIO DA CASA		(T) Solicita planejamento da construção (T) Prover recursos financeiros para a construção	(M) Fazer o alicerce da construção (S) Manter boa qualidade do trabalho	(M) Comprar materiais da construção (S) Manter boa relação financeira
ENGENHEIRO	(M) Fazer a planta da construção (T) Acompanhar todo o trabalho da construção		(T) Distribuir artigos (Soft/Aval)	
PEDREIRO	(M) Construir a casa (T) Verificar os materiais para a construção	(M) Informar o andamento da construção (T) planejar solicitação de materiais		(T) Verificar a entrega dos materiais
COMERCIANTE MATERIAL DE CONSTRUÇÃO	(M) Entregar todos os materiais solicitados (S) Manter prazo acordado		(M) Enviar as notas fiscais dos materiais	

Legenda: (M) Metas (T) Tarefas (S) Softgoal

QUADRO 4 – EXEMPLO DE MATRIZ PARA AS RELAÇÕES DE DEPENDÊNCIA

6.1.4. CRIAR UM MODELO DE DEPENDÊNCIA ESTRATÉGICA (SD) PRELIMINAR

Ao concluir-se a matriz na etapa 3 estabelecem-se às condições para constituir-se o Modelo de Dependência Estratégica. Ao mapear as relações entre atores através dos mediadores, há que se classificar cada mediador e, para fazê-lo, tem-se que compreendê-lo. Tais informações já permitem que se desenvolva o primeiro nível de representação gráfica do i^* , que mapeia as relações por meio da simbologia explicitada no capítulo 5 (o SD).

Entretanto, neste momento, passa-se a necessitar de recursos computacionais. Conforme foi abordado no estudo de caso, os dados obtidos foram processados por meio da Ferramenta OME que apóia modelagem em i^* , permitindo a visualização num modelo SD, conforme apresentado na Figura 13 do estudo de caso, capítulo 6.

6.1.5. CRIAR MODELOS DETALHADOS (SR) POR ATOR (DECOMPOSIÇÃO DE TAREFAS)

A partir do modelo SD, que é um dos propostos pela técnica i^* , verificou-se que tal modelo apenas destaca as relações de dependência entre

atores. Para uma análise mais crítica e apurada dessas relações, identificou-se a necessidade da decomposição de tarefas. Essa decomposição indica como satisfazer uma determinada entidade e, por outro lado, a razão pela qual as subtarefas são necessárias. Nesse contexto, o outro modelo proposto pelo i^* , o modelo SR, permite a avaliação e destaque das tarefas. Ao compreendê-las é possível reclassificá-las, se for necessário. Além disso, num grau de entendimento mais elevado, tem-se a tendência de decompor-se todas as tarefas complexas. Essa decomposição de tarefas é um recurso interessante, quando aplicável, pois permite que se otimizem os processos e corrijam falhas que se originam em pequenas coisas, mas que provocariam impactos significativos no sistema como um todo. Um ponto adicional da decomposição é avaliar relações de dependências secundárias onde uma parte da tarefa pode depender de um subsídio que é originário de um ator ainda não mapeado na relação em análise. Para essa etapa, também, é necessária a utilização de um *software* específico para se construir o modelo SR, conforme destacado anteriormente, utiliza-se a ferramenta OME.

6.1.6. JUNTAR OS MODELOS SR ISOLADOS NUM MODELO ESTRATÉGICO DETALHADO

A etapa final reúne todas as informações obtidas numa forma gráfica unificada. Ela é interessante para a verificação de aspectos macro, embora, todos os relacionamentos estejam ali descritos. A visualização ampla nos permite avaliar mentalmente todos os processos e movimentos previstos no sistema, nos dando uma compreensão do todo e da regularidade e interdependência das ações parciais. É uma figura complexa, desenvolvida a partir das etapas anteriores, com recursos gráficos e de *software*; produz consistência no conhecimento elaborado.

O modelo detalhado unificado, assim denominado, é o sistema vivo. Permite que cada uma de suas evoluções particulares possa ser, mentalmente exercitadas. Essa unificação é o próprio modelo SR, detalhado e acrescido de relacionamentos internos e organizados numa hierarquia de relacionamentos de meio-fim e de decomposição de tarefas. Cabe salientar que para essa unificação não existe um processo rígido estabelecido para a

inserção de novos elementos, conforme verificado na literatura pesquisada. As necessidades identificadas devem possibilitar a atribuição de valores qualitativos aos relacionamentos, a formalização e, além disso, pensar sobre estes relacionamentos como o cerne dessa unificação.

Não é incomum perceber que existem outros desdobramentos que podem conduzir a uma revisão do modelo. Porém, essa reinicialização ocorre de forma espiral, não se tratando apenas de inserção, mas de uma evolução do processo de elaboração na composição desse modelo. É um processo viabilizado a partir da ferramenta OME, embora sejam necessários ensaios manuais para alocar as necessidades apontadas. Este estágio da modelagem requer sensibilidade e conhecimento acerca do que o sistema se propõe.

Neste capítulo realizamos uma análise da aplicação da técnica i^* e apresentamos os benefícios, as dificuldades encontradas e elaboramos um guia de utilização da técnica, como contribuição para a pesquisa.

A seguir, no capítulo de conclusão, elaboramos o fechamento na dimensão do trabalho de pesquisa, apresentando considerações finais com base no estudo de caso desenvolvido, além de destacar possibilidades de ligações do presente trabalho com futuros desenvolvimentos.

7. CONCLUSÃO

Durante o presente estudo buscou-se destacar de forma precisa, a partir da vivência com o estudo de caso, um guia para conduzir o engenheiro de requisitos no processo de elicitação de requisitos orientada a metas, usando a técnica i*. A aplicação prática enfocou um estudo de caso, levando-se em consideração a premissa básica: em toda organização, a diversidade cultural e de ações levam a uma adequação de técnicas e de ferramentas para o desenvolvimento do trabalho.

Identificou-se que o processo de ER envolve desenvolvimento, descobrimento e o embasamento do documento de requisitos, enquanto fatores fundamentais para uma proposta eficiente de construção de sistemas, e que são vinculados ao entendimento dos requisitos.

A identificação de usuários, suas necessidades e como supri-las são atributos a serem desenvolvidos pelo engenheiro de requisitos - um mediador entre o usuário e suas necessidades. Num certo momento o engenheiro de requisitos identifica e produz informações que, em outro, organizará metodologicamente para o processo de desenvolvimento do sistema.

Considera-se que a contribuição do presente trabalho, ao tema Engenharia de Requisitos, foi apresentada por meio do guia desenvolvido no capítulo 6. Com a elaboração desse guia, conseguiu-se traçar um panorama dos problemas enfrentados no processo de utilização da técnica i*. A técnica de modelagem i*, enquanto mecanismo que ainda se mantém em evolução, requer delineamentos mais precisos na abordagem das relações de dependência entre atores. Enquanto ferramenta metodológica, esse guia destina-se também a traduzir aspectos atribuídos à subjetividade, pois os desafios não podem ser superados de uma forma puramente tecnológica.

O estudo de caso revelou dificuldades para a identificação de mediadores, sejam estes metas, tarefas, recursos ou *softgoal*. Essas dificuldades com os mediadores refletem dificuldades com os atores, ou seja,

ocorrem ambigüidades para a classificação dos mediadores que somente poderão ser tratadas pela ação do engenheiro de requisitos.

A definição conceitual que se estabeleceu no capítulo 4, quando se identificou os mediadores (Figura 12), deve ser utilizada para apontar estruturas que refletem os micro-relacionamentos e, então, elaborar a construção mais refinada e completa de todas as relações perceptíveis por meio da ferramenta OME, no modelo estratégico detalhado. Tal desenvolvimento somente será realizável quando o engenheiro de requisitos tratar todas as ambigüidades, ou impasses, que podem surgir no fluxo de relacionamentos apontados.

É importante destacar que na construção metodológica não se estabeleceram índices para avaliações de progresso, eficiência, desempenho, qualidade ou qualquer outro índice numérico que possa refletir uma avaliação quantitativa das relações de dependência. Entretanto, houve interesse na avaliação dos relacionamentos, a partir da estrutura básica, que apresenta a ligação entre dois atores, numa relação de necessidade a ser satisfeita.

A efetividade de mediação, por meio do cumprimento das tarefas, do alcance das metas, da obtenção dos recursos ou da concretização de *softgoal* permite viabilizar a estrutura de relacionamentos, configurando-se uma rede de ligações estabelecendo a evolução do sistema. A avaliação da mediação está colocada em termos absolutos, em representação binária: o mediador atendeu ou o mediador não atendeu o que lhe era demandado.

Como trabalhos futuros sugerimos o desenvolvimento de uma avaliação quantitativa de desempenho dos mediadores, a partir do porcentual de sucesso obtido em cada mediação identificada num determinado sistema. Essa proposta baseia-se no contexto do estudo de caso, onde se observou a possibilidade de avaliar o desempenho relativo para cada mediador, bem como, estabelecer um índice mínimo de desempenho que admitisse a realização da mediação desejada.

Por exemplo, num índice quantitativo, 78% das mediações apontadas entre os atores obtiveram êxito e outras 22% não obtiveram êxito.

Assim, a cada mediação que não foi realizada com sucesso, há um decréscimo. Para a aplicabilidade dessa avaliação é necessário o desenvolvimento de conceitos para sustentar avaliações numéricas e, conseqüentemente, possibilitar a interpretação do que os números, numa relação de escala, significam nos contextos do desenvolvimento do sistema e da eliciação dos requisitos.

Enfim, com tal apontamento, considera-se que novos estudos de casos devem ser incentivados, especialmente para o desenvolvimento de métricas que admitam, inclusive, níveis de atendimentos parciais para cada mediador em particular. Registra-se tal observação como encaminhamento para trabalhos futuros, contemplando desafios de compreensão associados ao tema, que devem ser significativos no mundo acadêmico e, naturalmente, devem favorecer os profissionais que se formam nesse âmbito.

REFERÊNCIAS BIBLIOGRÁFICAS

- ÁNTON, A.I. “**Goal-Based Requirements Analysis**”. Proceedings of Second IEEE International Conference on Requirements Engineering, Colorado Springs, USA, April, pp. 136-144, 1996.
- BATISTA, E.AP., CARVALHO, A.M. B. R “**Uma Taxonomia Facetada para Técnicas de Elicitação de Requisitos**”. Anais do WER03 - Workshop em Engenharia de Requisitos, Piracicaba-SP, Brasil, Novembro 27-28, pp 48-62, 2003.
- BERRY, D. M.; LAWRENCE, B. **Requirements Engineering**. 1ed. USA : IEEE *Software*, April, pp. 26-29, 1998.
- BREITMAN, K. K. e LEITE, J. C. S. P., “**Suporte Automatizado à Gerência da Evolução de Cenários**”, Workshop de Engenharia de Requisitos - WER98, 1998.
- BREITMAN, K. K. e LEITE, J. C. S. P., “**Processo de Software Baseado em Cenários**”, II (Ibero-American) Workshop on Requirements Engineering – Buenos Aires, September, pp. 95-105, 1999.
- BREITMAN, K. K., “**Evolução de Cenários**”. Tese de Doutorado apresentada ao Departamento de Informática da PUC/RJ, Maio, 2000.
- BROOKS JR, F.P. “**No Silver Bullet: Essences and Accidents of Software Engineering**” IEEE Compute. No 4, pp.10-19, 1987.
- BOOCH, G., “**Object-Oriented Analysis and Design with Application**”, Benjamin/Cummings, 1994.
- BOOCH, G., RUMBAUGH, J. e JACOBSON, I, “**The Unified Modeling Language User Guide**”, Addison Wesley, 1999.
- CARVALHO, A. M. B. R e CHIOSSI, T. C. S. “**Introdução à Engenharia de Software**”. Editora da Unicamp, Campinas-SP, 2001.
- DAWSON, L.; SWATMAN, P. A.: “**The use of object-oriented models in requirements engineering: a field study**”. ICIS International Conference on Information Systems. Proceeding of the 20th international conference on Information Systems, pp. 260-273, 1999.
- ENGESTROM, Y., “**Learning by Expanding**”, Helsinki: Orienta-Konsultit, 1987.
- FOURNIER, R. “**Guia Prático para Desenvolvimento e Manutenção de Sistemas Estruturados**”. Makron Books, 1994.
- GAUSE, D. C e WEINBERG, G. M. “**Are Your Lights On? How to Figure Out What the Problem Really Is**” 1ed. USA : Dorset House Publishing Co. Inc., 1990.

GOGUEN, J. e LINDE, C. "**Techniques for Requirements Elicitation**" In IEEE Intern. Symp. On Requirements Engineering, I San Diego. Proceedings. IEEE Computer Society Press. January, pp. 152-164, 1993.

INSFRÁN, E., PASTOR, O., WIERINGA, R. "**Requirements Engineering-Based Conceptual Modelling**" Requirements Engineering Publisher: Springer London. Vol. 7, Nr. 2, pp. 61-72, June 2002.

JACOBSON, I., CHRISTERSON, M., JONSSON, P., e GUNNAR, O., "**Object-Oriented Software Engineering: A Use Case Driven Approach**", Addison-Wesley. 1992.

JACKSON, M. "**Software Requirements and Specifications: A Lexicon of Praticce, Principles and Prejudices**" 1ed. USA, Massachusets: Addison-Wesley, Reading. pp. 228. 1995.

JACKSON, M. "**Problems and Requirements**" ISRE'95 Second International Symposium on Requirements Engineering. (York, England, UK). IEEE CSP, Los Alamitos, CA. Proceedings. March, pp. 2-8, 1995.

KOTONYA, G. e SOMMERVILLE, I. "**Requirements Engineering Processes e Techniques**" John Wiley and Sons, 1998.

KUUTI, K., "**Activity Theory as a Potential Framework for Human-Computer Interaction**", Context and Consciousness - Activity Theory and Human-Computer Interaction, MIT Press, pp. 17-44, 1996.

LAMSWEERDE, A. V. e LETIER, E. "**Handling Obstacles in Goal-Oriented Requirements Engineering**" IEEE Transactions on *Software Engineering*, vol. 26, nº 10. Outubro, 2000.

LAMSWEERDE, A. V. "**Goal-Oriented Requirements Engineering: A Guided Tour**" Proceedings of the 5th IEEE International Symposium on Requirements Engineering, pp. 249-263, 2001.

LEE, J. e XUE, N. "**Analyzing User Requirements By Use Cases: A Goal-Driven Ap-proach**". IEEE *Software* 16(4), Julho/Agosto, 1999.

LEITE, J. C. S. P, HADAD, G, DOORN, J., Kaplan G. "**A Scenario Construction Process**" Requirements Engineering Journal, Springer-Verlag, Vol. 5, N. 1, pp. 38-61, 2000.

LEITE, J. C. S. P. Yu, Y., MYLOPOULOS, J. "**From Goals to Aspects: Discovering Aspects from Requirements goal Models**" Proceedings of the 12th IEEE International Conference on Requirements Engineering : IEEE Computer Society, pp. 38-47, 2004.

LIN, Liu, YU, E., MYLOPOULOS, J. "**Security and Privacy Requirements Analysis within a Social Setting**" In: Proceedings of the International Conference on Requirements Engineering (RE'03). pp. 151-161. Monterey, California, September, 2003.

LIN, Liu, YU, E., MYLOPOULOS, J. "Analyzing Security Requirements as Relationships Among Strategic Actors" In: Proceedings of 2nd Symposium on Requirements Engineering for Information Security (SREIS'02). Raleigh, North Carolina, October 16, 2002.

MACAULAY, L. A. "Requirements Engineering." 1ed. Great Britain: Springer-Verlag London Limited, 202 p. 1996.

MARTINS, L. E. G. e DALTRINI, B. M., "An Approach to *Software Requirements Elicitation Using Precepts from Activity Theory*", Proceedings of the 14th IEEE International Conference on Automated *Software Engineering*, 1999.

MARTINS, L. E. G. e DALTRINI, B.M. "Utilização dos Preceitos da Teoria da Atividade na Elicitação dos Requisitos de *Software*". Anais do XIII Simpósio Brasileiro de Engenharia de *Software*, pp.63-76, Outubro 1999.

MARTINS, L. E. G., "Uma Metodologia de Elicitação de Requisitos de *Software Baseada na Teoria da Atividade*". Tese de Doutorado apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas. Agosto, 2001.

MYLOPOULOS, J., CHUNG, L. e NIXON, B., "Representing and Using Nonfunctional Requirements: A Process-Oriented Approach", IEEE Trans. on *Software Engineering*, Vol. 18 No. 6, pp. 483-497, June 1992.

MYLOPOULOS, J. C. e Yu, E. "From Object-Oriented to Goal-Oriented Requirements Analysis". Communications of the ACM, Vol.42,Nº1, pp. 31-37,Jan 1999.

NUSEIBEH, B. e EASTERBROOK, S., "Requirements Engineering: a Roadmap", International Conference on *Software Engineering – Proceedings of the conference on the future of Software Engineering 2000*. IEEE Computer Society Press. pp. 5-9, June 2000.

PEDROZA, F.P., ALENCAR, F.M.R., CASTRO, J., SILVA, F.R.C., SANTANDER, V.F.A. "Ferramentas para Suporte do Mapeamento da Modelagem i* para a UML: eXtended GOOD - XGOOD e GOOSE. [WER 2004](#): 164-175.

PINHEIRO, F. A. C.; GOGUEN, J. A. "An Object-Oriented Tool for Tracing Requirements" 1ed. USA : IEEE *Software*, p 52-64, Março, 1996.

POTTS, C., TAKAHASHI, K. e ANTON, A.I., "Inquiry-Based Requirements Analysis." IEEE *Software*, vol. 11, no. 2, February, 1994.

Pressman, R. S. "*Software Engineering: a Practitioner's Approach*". 5 th ed. McGraw Hill, 860 p., 2000.

REGNELL, B., ANDERSSON, M., BERGSTRAND, J. “**A Hierarchical Use Case Model with Graphical representation**”. IEEE International Symposium and Workshop on Engineering of Computer-Based Systems, Germany, March 1996.

RIDAO, M. DOORN, J. e LEITE, J. C. S. P., “**Uso de Patrones en la Construcción de Escenarios**” WER00 - Workshop em Engenharia de Requisitos. Julho, 2000.

ROLLAND, C., SOUVEYET, C. ACHOUR, C. Ben. **Guiding goal modelling using scenarios**. IEEE Transactions on *Software Engineering*, special issue on Scenario.Management, 1998.

RUMBAUGH, J., “**Getting Started: Using Use Cases to Capture Requirements**”, J. Object-Oriented Programming, pp. 8-12, Sept. 1994.

SUTCLIFFE, A. G., MAIDEN, N. A. M., MINOCHA, S. e MANUEL, D., “**Supporting Scenario-Based Requirements Engineering**”, IEEE Transactions on *Software Engineering*, vol. 24, no. 12, December, 1998.

SOMMERVILLE, I. SAWYER, P. “**Requirements Engineering – a good practice guide**” Wiley, 1997.

SOMMERVILLE, I. “**Engenharia de Software**”, 6ª edição. Pearson Education do Brasil, 2003.

THOMAS, P., OLIVEROS, A. ”**Elicitación de Objetivos a partir de Escenarios**” Anais do WER05 - Workshop em Engenharia de Requisitos, Porto, Portugal, Junho 13-14, pp 50-62, 2005.

YU, E. S. K. and MYLOPOULOS, J. C. “**Using goals, rules, and methods to support reasoning in business process Reengineering**”. Information Systems: Collaboration Technology Organizational Systems and Technology, Proceedings of the Twenty-Seventh Hawaii International Conference, vol. 4, pp. 234-243, 1994.

YU, E. S. K. “**Why Goal-Oriented Requirements Engineering**”. *Proceedings of Fourth International Workshop of Requirements Engineering: Foundation for Software Quality*. pp.15-22, June 1998.

YU, E. S. K. “**Strategic Modelling for Enterprise Integration**”. 14th World Congress of International Federation of Automatic Control, July, 1999.

YU, E. S. K, LIU, L. “**Modelling Trust in the i* Strategic Actors Framework**”. *Proceedings of the 3rd Workshop on Deception, Fraud and Trust in Agent Societies*. pp. 137-147, June, 2000.

YU, E. S. K, LIU, L. LI, Y. “**Modelling Strategic Actor Relationships to Support Intellectual Property Management**”. 20th International Conference on Conceptual Modeling, pp. 164-178, November, 2001.

YU, E. S. K. **“Agent-Oriented Modelling: *Software Versus the World*”** Workshop Proceedings. pp. 206-225, 2001.

YU, Y. LEITE, J. C. S. P., MYLOPOULOS, J. **“From Goals to Aspects: Discovering Aspects from Requirements Goal Models”** 12th IEEE International Requirements Engineering Conference. pp.38-47, 2004.

WEBSTER, I., AMARAL, J., CYSNEIROS, L. M. **“A Survey of Good Practices and Misuses for Modelling with i* Framework”** Anais do WER05 - Workshop em Engenharia de Requisitos, Porto, Portugal, Junho 13-14, pp 148-160, 2005.

ZAVE, P. **Classification of Research Efforts in Requirements Engineering.** ISRE'95 Second International Symposium on Requirements Engineering. (York, England, UK) 1ed.USA : IEEE CSP, Los Alamitos, CA. Proceedings, pp. 214-216. Março, 1995.

ZORMAN, L., **“Requirements Envisaging by Utilizing Scenarios (Rebus)”**, PhD. Dissertation, University of Southern California, 1995.