

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
PROGRAMA DE MESTRADO EM ENGENHARIA DE PRODUÇÃO

A IMPORTÂNCIA DA TÉCNICA DE MODELAGEM
ORGANIZACIONAL EKD NO DESENVOLVIMENTO DE
DIAGRAMAS USE CASE

Fabiana Serralha Miranda de Pádua

Serviço de Pós-Graduação EESC/USP

EXEMPLAR REVISADO

Data de entrada no Serviço: 02.10.03

Ass: 

DEDALUS - Acervo - EESC



31100043534

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Produção.

ORIENTADOR: Prof. Dr. Edson Walmir Cazarini

SÃO CARLOS

2003



Class.	TESE - EESC
Cutt.	4344
Tombo	T180/03
Sysno	132.5510

31100043534

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

P125i

Pádua, Fabiana Serralha Miranda de
A importância da técnica de modelagem
organizacional EKD no desenvolvimento de diagramas use
case / Fabiana Serralha Miranda de Pádua. -- São
Carlos, 2003.

Dissertação (Mestrado) -- Escola de Engenharia de
São Carlos-Universidade de São Paulo, 2003.

Área: Engenharia de Produção.


Orientador: Prof. Dr. Edson Walmir Cazarini.

1. Modelagem organizacional. 2. UML. 3. Técnica
EKD. 4. Diagrama use case. I. Título.

FOLHA DE JULGAMENTO

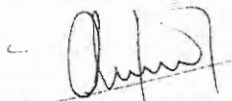
Candidata: Bacharel **FABIANA SERRALHA MIRANDA DE PADUA**

Dissertação defendida e julgada em 02-04-2003 perante a Comissão Julgadora:



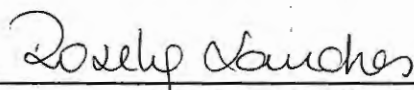
Prof. Dr. **EDSON WALMIR CAZARINI (Orientador)**
(Escola de Engenharia de São Carlos/USP)

Aprovada



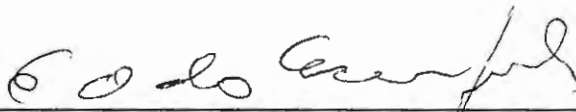
Profa. Dra. **ÉTHEL CRISTINA CHIARI DA SILVA**
(Universidade de Ribeirão Preto/UNAERP)

Aprovada

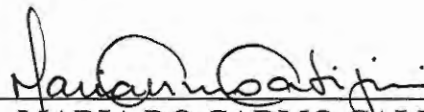


Profa. Dra. **ROSELY SANCHES**
(Instituto de Ciências Matemáticas e de Computação/USP/São Carlos)

aprovado



Prof. Doutor **EDMUNDO ESCRIVÃO FILHO**
Coordenador do Programa de Pós-Graduação
em Engenharia de Produção



Profa. Assoc. **MARIA DO CARMO CALIJURI**
Presidente da Comissão de Pós-Graduação da EESC

Dedicatória

Ao meu marido Pedro e meus filhos João Victor e Tiago

Freqüentemente meu cansaço e minhas preocupações foram sentidos e compartilhados por vocês.

Tantas foram às vezes em que me ausentei, para a realização deste trabalho. Hoje cheguei lá... E vocês estão juntos de mim, como no começo de tudo.

A alegria desta conquista também é de vocês, afinal, o amor, o estímulo e compreensão são a alma desta vitória.

Dedicatória

A minha Mãe, ao meu Avô José e a minha Avó Maria

A vocês que me ensinaram a viver com dignidade e que iluminaram os meus caminhos com carinho e dedicação para que eu caminhasse sem medo e com esperança

Agradeço a vocês tudo que sou e tudo que fizeram por mim, renunciando muitas vezes de seus sonhos para que eu pudesse realizar os meus.

Dedicatória

A minha irmã Daiúcha, e ao meu irmão Nicolas, que fazem parte do que há de mais precioso neste mundo, a família.

Agradecimentos

A Deus, que me guiou e me deu saúde, inteligência e vontade para realizar este trabalho.

Ao Prof. Dr. Edson Walmir Cazarini, pela amizade e profissionalismo durante o desenvolvimento deste trabalho.

A minha secretária Cleusa, pela dedicação com meus filhos e com a minha casa.

Aos funcionários Silvana e José Luis, e professores da Escola de Engenharia de São Carlos.

A todos os amigos que através de suas experiências contribuíram de forma direta ou indireta para que este trabalho fosse realizado.

SUMÁRIO

LISTA DE FIGURAS.....	VI
LISTA DE ABREVIACÕES	VI
RESUMO.....	VII
1.0. INTRODUÇÃO	01
<hr/>	
1.1. CONTEXTO	01
1.2. MOTIVAÇÃO DO TRABALHO	03
1.3. OBJETIVO	04
1.4. METODOLOGIA DE PESQUISA	04
1.5. ORGANIZAÇÃO DO TRABALHO	05
2.0. ENGENHARIA DE REQUISITOS	07
<hr/>	
2.1. CONSIDERAÇÕES INICIAIS.....	07
2.2. CONTEXTO E DEFINIÇÕES DA ENGENHARIA DE REQUISITOS	07
2.3. PROCESSO DE ENGENHARIA DE REQUISITOS	12
2.4. PROBLEMAS EXISTENTES NA ENGENHARIA DE REQUISITOS	16
2.5. CLASSIFICAÇÃO DE REQUISITOS	17
2.6. CONSIDERAÇÕES FINAIS	18
3.0. MODELAGEM ORGANIZACIONAL.....	20
<hr/>	
3.1. CONSIDERAÇÕES INICIAIS.....	20
3.2. CONTEXTO DA MODELAGEM ORGANIZACIONAL.....	20
3.3. O MODELO ORGANIZACIONAL.....	22
3.4. CONHECIMENTO ORGANIZACIONAL	23
3.5. O MODELO EKD	23
3.5.1. BENEFÍCIOS DO EKD	29
3.6. COMPONENTES DO EKD	30
3.6.1. MODELO DE OBJETIVOS	34
3.6.1.1. LIGAÇÃO DENTRO DO MODELO DE OBJETIVOS	36

3.6.1.	MODELO DE OBJETIVOS	34
3.6.1.1.	LIGAÇÃO DENTRO DO MODELO DE OBJETIVOS	36
3.6.1.2.	DESENVOLVIMENTO DO MODELO DE OBJETIVOS.....	38
3.6.1.3.	REFINAMENTO E OPERACIONALIZAÇÃO DOS OBJETIVOS	39
3.6.2.	MODELO DE REGRAS DO NEGÓCIO	45
3.6.3.	MODELO DE CONCEITOS	51
3.6.4.	MODELO DE PROCESSOS DO NEGÓCIO.....	55
3.6.5.	MODELO DE ATORES E RECURSOS.....	61
3.6.6.	MODELO DE REQUISITOS E COMPONENTS TÉCNICOS	66
3.6.7.	RELACIONAMENTO ENTRE SUBMODELOS.....	71
3.7.	COMPONENTES PARA AUXILIAR A MODELAGEM	75
3.8.	UTILIZAÇÃO DA ABORDAGEM EKD	76
3.9.	CONSIDERAÇÕES FINAIS	87
4.0.	ABORDAGEM <i>USE CASE</i>.....	89
<hr/>		
4.1.	CONSIDERAÇÕES INICIAIS.....	89
4.2.	UML.....	89
4.3.	MODELAGEM <i>USE CASE</i>	90
4.4.	DIAGRAMA <i>USE CASE</i>	92
4.5.	ATORES	93
4.5.1.	RECONHECENDO ATORES	94
4.5.2.	ATORES EM UML.....	94
4.5.3.	RELACIONAMENTO ENTRE OS ATORES	95
4.6.	USE CASES.....	95
4.6.1.	IDENTIFICAÇÃO DE <i>USE CASES</i>	98
4.6.2.	RELACIONAMENTO ENTRE <i>USE CASES</i>	98
4.7.	CONSTRUÇÃO DO DIAGRAMA <i>USE CASE</i>.....	99
4.8.	PROBLEMAS RELACIONADOS AO DIAGRAMA <i>USE CASE</i>.....	100
4.9.	CONSIDERAÇÕES FINAIS	101
5.0.	CONTRIBUIÇÃO DA TÉCNICA EKD NO DESENVOLVIMENTO DE DIAGRAMA <i>USE CASE</i>	103
<hr/>		
5.1.	CONSIDERAÇÕES INICIAIS.....	103
5.2.	INTRODUÇÃO	103

5.3.	A LIGAÇÃO ENTRE A ESPECIFICAÇÃO DE REQUISITOS E O	
	MODELO ORGANIZACIONAL.....	105
5.4.	DESENVOLVIMENTO DO DIAGRAMA USE CASE A PARTIR DOS	
	MODELOS EXISTENTES NA TÉCNICA EKD.....	107
5.4.1.	DIRETRIZES PARA DESCOBERTA DE ATORES	108
5.4.2.	DIRETRIZES PARA DESCOBERTA DE USE CASES	113
5.5.	APLICAÇÃO DAS DIRETRIZES.....	118
5.6.	CONSIDERAÇÕES FINAIS	121
6.0.	CONCLUSÕES.....	122
<hr/>		
7.0.	REFERÊNCIAS BIBLIOGRÁFICAS	125
<hr/>		

LISTA DE FIGURAS

FIGURA 01- Custo do Software.....	08
FIGURA 02- Erros e Custo de Correção.....	09
FIGURA 03- Custo de Correção de Erros.....	09
FIGURA 04- Conteúdo da Estrutura do EKD.....	25
FIGURA 05- Uma visão da modelagem EKD.....	27
FIGURA 06- Submodelos que compõe o Modelo Organizacional.....	34
FIGURA 07- Fração de um Modelo de Objetivos contendo relacionamento AND.....	37
FIGURA 08- Fração de um Modelo de Objetivos contendo relacionamento OR.....	37
FIGURA 09- Notações do Modelo de Objetivos.....	38
FIGURA 10- Matriz de Classificação de Objetivos.....	42
FIGURA 11- Matriz de Correlação de Objetivos.....	42
FIGURA 12- Parte de um Modelo de Objetivos do caso de uma Biblioteca.....	44
FIGURA 13- Estrutura de Regras do Negócio.....	47
FIGURA 14- Regras referem-se a Entidades do Modelo de Conceitos e são apoiadas por Processos do Modelo de Processos do Negócio.....	48
FIGURA 15- Interação entre o Modelo de Regras do Negócio e componentes do Modelo Objetivos.....	50
FIGURA 16- Exemplo de Relacionamento Binário.....	52
FIGURA 17- O conceito “cliente” pode ser especializado de formas diferentes.....	53
FIGURA 18- Exemplo de relacionamento Part Of.....	54
FIGURA 19- O processo de verificação do endereço cliente não é decomposto.....	57
FIGURA 20- O processo de verificação do endereço cliente é decomposto.....	58
FIGURA 21- Um exemplo de Modelo de Processos do Negócio.....	59
FIGURA 22- Notações dos Componentes do Modelo de Atores e Recursos.....	64
FIGURA 23- Um exemplo de Modelo de Atores e Recursos de uma Biblioteca.....	66
FIGURA 24- Exemplo de um Modelo de Requisitos e Componentes Técnicos.....	70
FIGURA 25- Relações entre Submodelos.....	71
FIGURA 26- Ligações entre Submodelos do caso da Biblioteca.....	74
FIGURA 27- Diagrama <i>Use Case</i>	92

FIGURA 28- Representação do ator em diagrama <i>Use Case</i>	95
FIGURA 29- Representação de <i>Use Cases</i>	97
FIGURA 30- Notações para <i>Use Cases</i> em UML.....	99
FIGURA 31- Ligação entre o Modelo Objetivos e o Modelo de Processos do Negócio	115
FIGURA 32- Ligação entre o Modelo de Regras do Negócio e o Modelo de Processos do Negócio	116
FIGURA 33- Ligação entre o Modelo de Processos do Negócio e o Modelo de Atores e Recursos	118
FIGURA 34- Diagrama <i>Use Case</i> para um Sistema de Biblioteca.....	120

LISTA DE ABREVIACÕES E SIGLAS

AOO	Análise Orientada a Objeto
DER	Diagrama Entidade Relacionamento
DFD	Diagrama de Fluxo de Dados
EKD	Enterprise Knowledge Development
MAR	Modelo de Atores e Recursos
MC	Modelo de Conceitos
MER	Modelo Entidade Relacionamento
MRCT	Modelo de Requisitos e Componentes Técnicos
MO	Modelo de Objetivos
MRN	Modelo de Regras do Negócio
UML	Unified Modeling Language
OMG	Object Management Group

Resumo

PÁDUA, F.S.M. (2003). **A Importância da Técnica de Modelagem Organizacional EKD no Desenvolvimento de Diagramas *Use Case***. 130 f. (Mestrado em Engenharia de Produção) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos.

Tipicamente, Diagramas *Use Case* têm sido utilizados para capturar os requisitos funcionais do sistema a ser desenvolvido. Contudo o desenvolvimento de sistemas ocorre dentro de contextos onde processos organizacionais estão estabelecidos. Portanto, é preciso capturar os requisitos organizacionais para definir como o sistema pretendido irá satisfazer os objetivos da organização, por que ele é necessário e quais as implicações das alternativas para as várias partes envolvidas. Infelizmente UML e Diagramas *Use Case* não estão equipados para modelar os requisitos organizacionais, que em geral são indispensáveis para a eficácia do sistema. Neste sentido, a técnica de Modelagem Organizacional EKD facilita a compreensão do ambiente empresarial e é reconhecida como uma atividade valiosa para a Engenharia de Requisitos. Através da técnica de Modelagem Organizacional EKD são representados dentre outros, processos, informações, atores, objetivos organizacionais, políticas e requisitos funcionais e não funcionais. No entanto, os requisitos organizacionais definidos a partir da técnica EKD precisam ser relacionados aos requisitos funcionais representados pelo diagrama *Use Case*. Como resultado desta integração, os requisitos necessários para uma solução computacional de qualidade poderão ser melhores elicitados e especificados. Dessa forma, o presente trabalho tem como objetivo, analisar a técnica de Modelagem Organizacional EKD, mostrando sua importância no desenvolvimento de Diagramas *Use Case*.

Palavras-chaves: Modelagem Organizacional, UML, Técnica EKD, Diagramas *Use Case*.

Abstract

PÁDUA, F.S.M. (2003). **The Importance of the Technique of Organizational Modeling EKD to Development Use Cases Diagrams**. 130 f. Dissertação (mestrado em Engenharia de Produção) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos.

Use Cases Diagrams have been used for capturing system functional requirements. However, system development occurs in a context where organization process are will established. Therefore, need to capture organizational requirements to define how the system fulfills the organization goals, why it is necessary, what are the possible alternatives and what are the implications to the involved parts. Unfortunately, UML and Use Cases Diagrams not are will equipped for modeling organizational requirement, that in general, are indispensable for the efficacy of the system. Nevertheless, organizational requirements must related to functional requirements represented as Use Case Diagrams. This way the techniques of Organizational Modeling EKD, facilitates the understanding of the business environment and it is recognized like a valious activity by the Requirements Engineering. Through the Technical EKD are represented among other, processes, information, actors, organizational goals, strategies, politics and requirements functional and not-functional. In this sense, the present work objects, to analyze the Technique of Organizational Modeling EKD, showing its importance to the development Use Cases Diagrams.

Key Words: Organizational Modeling, Technical EKD, UML, Use Case Diagrams.

Introdução

1.1 – Contexto

O desenvolvimento de software cada vez mais complexo, passível de certificação e com menor custo possível, têm-se tornado um desafio constante para a comunidade de Engenharia de Software. Diversas técnicas, metodologias e ferramentas vêm sendo propostas com o intuito de suportar e auxiliar a produção de software de qualidade.

Neste contexto, uma das etapas mais críticas está relacionada à Engenharia de Requisitos. Segundo JACKSON (1995), a principal função da Engenharia de Requisitos é gerar especificações que descrevam de forma não ambígua, consistente e completa o comportamento do universo do domínio do problema.

Freqüentemente requisitos de software são inadequadamente elicitados, analisados e especificados, o que resulta na produção de um sistema que não satisfaz as necessidades do cliente.

Segundo PRESSMAN (1995), mesmo que o sistema sendo bem projetado ou codificado se a análise e especificação de requisitos forem pobres, o cliente ficará desapontado e levará o fracasso ao desenvolvedor.

Desta forma, uma das maiores preocupações da comunidade acadêmica e industrial tem sido o desenvolvimento de novas técnicas de suporte as atividades de Engenharia de Requisitos.

Diversas abordagens para elicitar, analisar e especificar tem surgido desde as primeiras técnicas utilizadas com esse fim. No início a análise estruturada, diagrama de fluxo de dados (DFD's) e modelagem entidade-relacionamento (ME-R) procuravam atentar para análise de requisitos, porém as informações capturadas nesses modelos descrevem apenas entidades, funções, fluxo de dados e estados do sistema não expressando as razões envolvidas no processo, ou seja, o porquê fazer uma determinada ação ou tomar uma decisão. As notações gráficas utilizadas por essas ferramentas são muito simples e não permitem análises precisas, avaliação de alternativas para os processos do negócio, entendimento dos objetivos e estratégias, o que inviabiliza uma visão do contexto mais próxima da realidade da organização.

Atualmente, técnicas tais como cenários, *Use Cases*, etnografias, prototipação e outras são apontadas como alternativas para suportar a atividade de Engenharia de Requisitos.

Dentre estas abordagens, técnicas baseadas em cenários tem recebido uma atenção especial. Uma das abordagens de cenários, que tem-se destacado é *Use Cases*.

Esta técnica é parte integrante e central da linguagem UML, (BOOCH *et al*, 1999), a qual é padrão de linguagem de modelagem para o desenvolvimento de software orientado a objetos.

A importância do diagrama *Use Case* na análise orientada a objeto é destacado por diversos autores como PRADO (1999), JACOBSON *et al* (1999), LARMAN (1999), FOWLER & SCOTT (1997), como sendo de suma relevância para compreender o âmbito do problema a ser modelado.

Apesar do consenso e do reconhecimento de cenários como uma ferramenta importante no processo de Engenharia de Requisitos pode-se apontar algumas carências da técnica principalmente no que diz respeito à inclusão de aspectos inerentes ao ambiente organizacional no qual o software está inserido.

Segundo SANTANDER & CASTRO (2000), quando usuários desejam desenvolver um software, grande parte dos casos não se tem uma idéia concreta do que os mesmos desejam. Geralmente o que se tem são intenções e desejos de facilitar a execução de atividades no ambiente organizacional.

Muitos problemas associados ao desenvolvimento de software podem-se iniciar nesta fase, pois detectar o que é realmente relevante para o usuário, levando em consideração os objetivos organizacionais não é uma tarefa trivial.

Entender a organização é importante para a Engenharia de Requisitos, porque muitos problemas na definição de requisitos são originados devido ao pouco conhecimento das atividades da organização e de seus sistemas de informação.

Portanto, torna-se fundamental capturar os requisitos organizacionais para definir como o sistema pretendido irá satisfazer os objetivos da organização, porque ele é necessário, quais as alternativas existentes, quais as implicações das alternativas para as várias partes envolvidas.

Infelizmente, técnicas de modelagem Orientada a Objetos, especificamente UML não estão equipadas para modelar os requisitos organizacionais. Torna-se necessário o uso de uma outra técnica, tal como EKD para representar tais aspectos. Nosso principal argumento é que a técnica de Modelagem Organizacional – EKD através de seus submodelos direciona o processo de conhecimento do ambiente, sendo adequada para auxiliar a definição de requisitos organizacionais.

Os requisitos organizacionais não devem ser considerados como uma simples descrição da funcionalidade do sistema, pois tratam do domínio onde o sistema está inserido. Nesse contexto, segundo ALENCAR (1999), a Modelagem Organizacional facilita a compreensão do ambiente e é reconhecida como crucial para a Engenharia de Requisitos.

1.2. Motivação

Quando desenvolvemos sistemas nós usualmente necessitamos ter um amplo entendimento do ambiente organizacional e seus objetivos.

É bastante aceito que o trabalho de Engenharia de Requisitos pode ser melhorado se modelarmos aspectos organizacionais visando entender melhor as intenções (objetivos do negócio) e motivações organizacionais. Nesse sentido, várias propostas objetivando a Modelagem Organizacional têm sido apontadas tais como: a técnica de ORDIT; técnica de BUBENKO (1993); técnica i^* de YU (1995); a modelagem de FURLAN (1997) e a técnica EKD (2001).

As maiorias dessas modelagens objetivam fornecer recursos de forma a modelar as intenções, relacionamentos e motivações entre os membros da organização. A partir desses modelos permite-se compreender melhor o ambiente organizacional bem como os objetivos, restrições, regras gerais, relações humanas e de trabalho entre os participantes da organização. No entanto, os requisitos organizacionais definidos a partir da Modelagem Organizacional precisam ser relacionados aos requisitos funcionais representados pelo diagrama *Use Case*.

Como resultado esperado desta integração, os requisitos necessários para uma solução computacional de qualidade poderão ser melhores elicitados e especificados.

No presente trabalho será tratado a técnica EKD proposta por BUBENKO, que tem seus trabalhos mencionados como referências bibliográficas por outros autores de Modelagem Organizacional. Os motivos pelos quais optou-se em estudar a técnica EKD foi pela facilidade de seus submodelos serem entendidos e analisados por profissionais não especificamente ligados a área de desenvolvimento de software, por apresentar uma notação gráfica simples e ainda por fornecer um conjunto de diretrizes para sua introdução e aplicação.

1.3. Objetivo

O presente trabalho tem como objetivo identificar de que forma o desenvolvimento de um Modelo Organizacional, através da técnica EKD, pode complementar e servir de fonte de informação para o desenvolvimento de *Use Cases*.

A pesquisa consiste em estabelecer uma correlação entre os submodelos do EKD e os elementos componentes em *Use Cases* com a finalidade de obter a especificação de requisitos. Assim, fatores do ambiente organizacional inicialmente não considerados em *Use Cases* poderão ser integrados ao mesmo.

Para atingir este objetivo será necessária uma revisão bibliográfica sobre:

1. Modelagem Organizacional e a Técnica EKD;
2. Engenharia de Requisitos;
3. *Use Cases*.

1.4. Método de Pesquisa

No presente trabalho será empregada inicialmente a Pesquisa Bibliográfica, com o objetivo de recolher informações e conhecimentos sobre o problema levantado em relação ao diagrama *Use Case*.

Segundo CERVO & BERVIAN (1983), a Pesquisa Bibliográfica procura explicar um problema a partir de referências teóricas publicadas em documentos, buscando-se conhecer e analisar as contribuições culturais ou científicas do passado existentes sobre um determinado assunto, tema ou problema. Os objetivos da pesquisa bibliográfica são:

- Evitar duplicidade de pesquisa
- Determinar a contribuição da pesquisa para base de conhecimento
- Obter fundamentação teórica para o desenvolvimento do projeto.

Nas investigações em geral, nunca se utiliza apenas um método, mas quantos forem necessários ou apropriados para determinado caso. Segundo LAKATOS & MARCONI (2001), independente do método a ser utilizado, pode-se definir método como um conjunto de atividades sistemáticas e racionais que, com maior segurança e economia permitem alcançar o objetivo – conhecimentos válidos e verdadeiros, traçando o caminho a ser seguido, detectando erros e auxiliando as decisões do cientista. Neste sentido para atingir o objetivo proposto será utilizado o método Hipotético-Dedutivo.

Para BUNGE (1974) *apud* LAKATOS & MARCONI (2001), as etapas desse método são:

a) Colocação do problema:

- Reconhecimento dos fatos: exame, classificação, preliminar e seleção dos fatos que, com maior probabilidade, são relevantes no que respeita a algum aspecto;
- Descoberta do problema: encontro de lacunas ou incoerências no saber existente;
- Formulação do problema: colocação de uma questão que tenha alguma probabilidade de ser correta; em outras palavras, redução do problema a um núcleo significativo com probabilidades de se solucionado e de apresentar-se frutífero, com o auxílio do conhecimento disponível.

b) Construção de um modelo teórico:

- Seleção dos fatores pertinentes: invenção de suposições plausíveis que se relacionem a variáveis supostamente pertinentes;
- Invenção das hipóteses centrais e das suposições auxiliares: proposta de um conjunto de suposições que sejam concernentes a supostos nexos entre as variáveis.

c) Dedução de conseqüências particulares

- Procura de suportes racionais: dedução de conseqüências particulares que, no mesmo campo, ou campos contíguos, possam ter sido verificadas;
- Procura de suportes empíricos: tendo em vista as verificações disponíveis ou concebíveis, elaboração de predições ou retrodições, tendo por base o modelo teórico e dados empíricos.

Nesta fase metodológica, foram encontradas as sugestões que determinaram, na conclusão deste trabalho, as alterações no Diagrama *Use Case*.

d) Teste das hipóteses

- Execução da prova: realização das operações planejadas e nova coleta de dados;
- Esboço da prova: planejamento dos meios para por aa prova as predições e retrodições;
- Elaboração dos dados: procedimentos de classificação, análise, redução e outros, referentes aos dados empíricos coletados;
- Inferência da conclusão: à luz do modelo teórico, interpretação dos dados já elaborados.

e) Adição ou introdução das conclusões na teoria:

- Comparação das conclusões com as predições e retrodições: contraste dos resultados da prova com as conseqüências deduzidas do modelo teórico, precisando o grau em que este pode, agora, ser considerado confirmado ou não.
- Reajuste do modelo: caso necessário, eventual correção ou reajuste do modelo;
- Sugestões para trabalhos posteriores: caso o modelo não tenha sido confirmado, procura dos erros ou na teoria ou nos procedimentos empíricos, caso contrário – confirmação -, exame de possíveis extensões ou desdobramentos, inclusive em outras áreas do saber.

1.5. Organização do Trabalho

O trabalho está organizado em 6 capítulos, divididos da seguinte maneira.

No capítulo 1, é apresentada a definição do trabalho. É discutido inicialmente o contexto em que o trabalho se insere, a motivação pela escolha do tema e a definição do objetivo.

No capítulo 2, são apresentados os conceitos e principais processos da Engenharia de Requisitos.

No capítulo 3, são apresentados os conceitos relativos a Modelagem Organizacional e a Técnica de Modelagem Organizacional EKD.

No capítulo 4 é realizada uma descrição de UML e dado o enfoque à abordagem *Use Case*

No capítulo 5 apresentamos algumas diretrizes que visam auxiliar o desenvolvimento de *Use Cases* a partir da Modelagem Organizacional.

No capítulo 6 são apresentadas as conclusões obtidas e sugestões para trabalhos futuros nesta área de pesquisa.

2. Engenharia de Requisitos

2.1. Considerações Iniciais

O interesse crescente pelo estudo de formas de melhorar a definição de sistemas de software levou a criação de uma nova área de pesquisa relacionada à Engenharia de Software. Deu-se a essa área o nome de Engenharia de Requisitos.

O objetivo desta denominação é ressaltar que o processo de definir requisitos de software é uma atividade extremamente importante e indispensável das outras atividades da Engenharia de Software.

Atualmente a Engenharia de Requisitos tem sido identificada como uma fase crucial no ciclo de vida do desenvolvimento de software, por tratar de conhecimentos não apenas técnicos, mas também gerenciais, organizacionais, econômicos e sociais.

Este capítulo tem como objetivo definir os processos oferecidos na Engenharia de Requisitos para uma correta definição e análise de requisitos.

2.2. Contextualização e principais definições da Engenharia de Requisitos

O objetivo do desenvolvimento de software é a criação de sistemas de software que correspondam às necessidades de clientes /e usuários. Uma completa especificação dos requisitos do software é essencial para o sucesso do seu desenvolvimento.

Estudos recentes mostram que existe um grande número de Sistemas de Informação que não são apropriados para as necessidades de seus usuários.

Segundo Castro (1995), o nível de aceitação dos sistemas de informação comerciais é da ordem de 40%, enquanto para os sistemas de tempo real o índice de aceitação sobe para 75%. Na maioria das vezes, os motivos de tal insucesso se devem ao fato de que os requisitos não foram bem definidos e/ou entendidos.

Para o autor vários são os motivos para se estudar Engenharia de Requisitos, que variam desde razões sociais a motivos econômicos.

Do ponto de vista jurídico desejamos que o documento de requisitos funcione como um acordo contratual entre clientes e fornecedores de software. Portanto, é necessário que existam técnicas apropriadas para a especificação de sistemas. A equipe responsável pelo desenvolvimento de software deve ter obrigação de inquirir sobre os requisitos dos seus clientes, visando a uma melhor compreensão do sistema a ser desenvolvido. Assim, é necessário um melhor conhecimento das técnicas de aquisição de requisitos.

Existem fortes razões econômicas para se estudar a Engenharia de Requisitos, pois embora a etapa de análise represente um pequeno custo no desenvolvimento de manutenção de software (figura 1) é de vital importância que ela seja bem realizada.

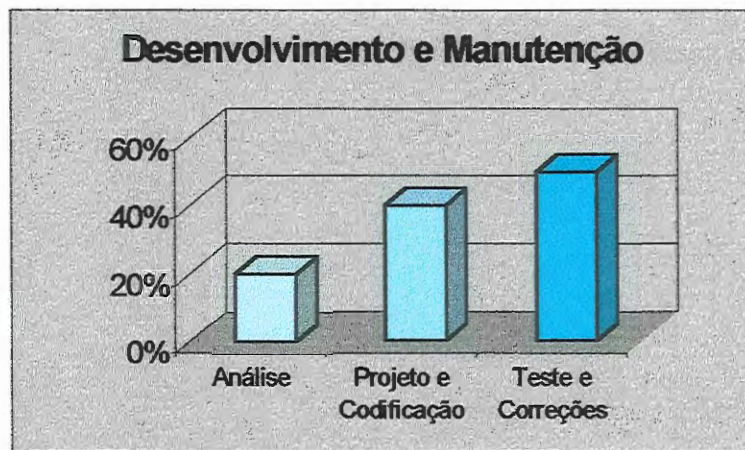


Figura 01 – Custo do Software

Fonte: Castro (1995), p.05.

Segundo Castro (1995) cerca de 40% dos erros cometidos ocorrem na fase de especificação do sistema (figura 02). Como eles são os mais caros de consertar, a correção dos problemas oriundos da fase de especificação atingem a um patamar de 66% do custo total da correção dos erros do projeto.

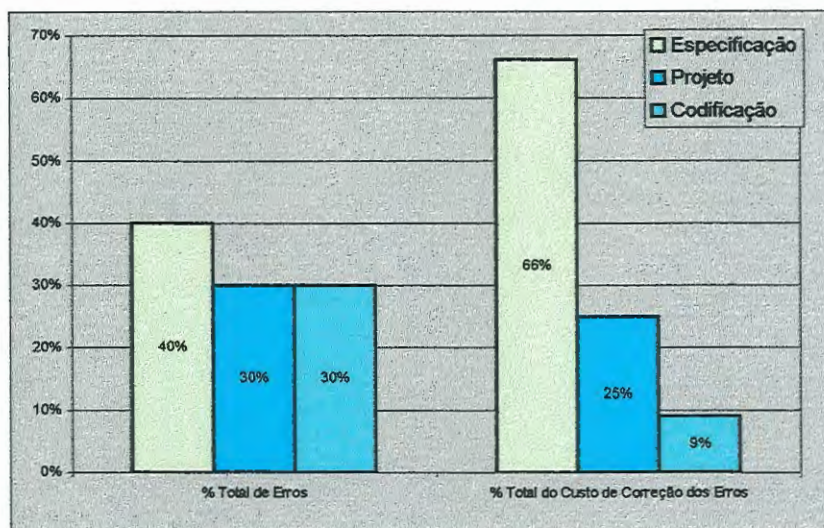


Figura 02 – Erros e Custo de Correção

Fonte: Castro (1995), p.06.

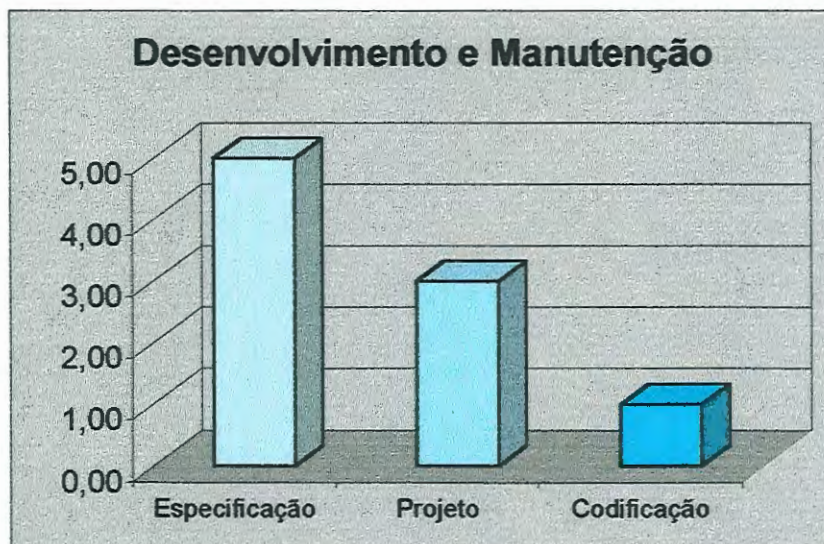


Figura 03 – Custo de Correção de erros

Fonte: Castro (1995), p.07.

Quanto mais tarde for detectada uma decisão errada mais caro será corrigi-la. Conforme figura 03, a correção de um erro de especificação é cerca de cinco vezes mais cara do que um erro de codificação.

Muitos dos problemas oriundos de sistemas mal desenvolvidos são causados por uma má definição de requisitos nos estágios iniciais do processo de desenvolvimento, resultando na produção de um sistema que não satisfaz as necessidades do cliente. Como consequência diversas tarefas do ciclo de desenvolvimento de software devem ser refeitas, o que eleva seu custo de manutenção.

De acordo com DE BERTOLI (1999), vários fatores dificultam a realização adequada na definição de requisitos, entre eles:

- Dificuldade dos desenvolvedores entenderem com exatidão o problema a ser resolvido;
- Dificuldade de comunicação entre os envolvidos no processo;
- Dificuldade de acomodar as mudanças que surgem durante o desenvolvimento;
- Muitas funções desempenhadas são pouco estruturadas, ou até mesmo não estruturadas, envolvendo “*Know How*” para sua realização;
- Os fluxos de informações nem sempre são óbvios e de fácil percepção e o comportamento humano é dinâmico, difícil de entender e explicar, não sendo possível medi-lo ou formalizá-lo.

Diante dos vários problemas encontrados na fase de definição de requisitos, a Engenharia de Requisitos surge na tentativa de solucionar ou minimizar os mesmos.

A Engenharia de Requisitos segundo CHICHINELLI (2002), procura sistematizar o processo de definição de requisitos, sistematização essa necessária devido a complexidade dos sistemas atuais.

Portanto, para melhorar o entendimento sobre Engenharia de Requisitos torna-se necessário a definição de alguns conceitos tais como: Requisitos e Engenharia de Requisitos. Nesta seção apresentaremos tanto a definição do IEEE como a de diversos autores.

O glossário de engenharia de software do IEEE (1990) define requisitos como:

1. Uma condição ou capacidade necessária a um usuário para resolver um problema ou alcançar um objetivo;
2. Uma condição ou capacidade que deve ser satisfeita ou possuída por um sistema ou componente do sistema para satisfazer um contrato, um padrão, especificação ou outros documentos impostos formalmente.
3. Uma representação documentada de uma condição ou capacidade como em (1) ou (2).

Para LEITE (1994), requisito é a condição necessária para obtenção de certos objetivos, ou para preenchimento de certo objetivo.

Requisito para MACAULAY (1996), simplesmente pode ser definido como “algo que um cliente necessita”. Entretanto do ponto de vista do Engenheiro de Software, requisitos pode ser definido como “algo que necessita ser projetado”.

Já ALENCAR (1999), define requisitos como um conjunto de descrições de como o sistema pretendido deve se comportar, ou um conjunto de propriedades, de atributos do sistema e/ou limitações do próprio processo de desenvolvimento de software.

Segundo a visão dos autores citados acima, podemos concluir que requisitos são objetivos ou restrições estabelecidas por clientes e usuários do sistema que definem as diversas propriedades do sistema. Os requisitos do software são, obviamente, aqueles dentre os requisitos do sistema que dizem respeito a propriedades do software.

A Engenharia de Requisitos, segundo MACAULAY (1996), pode ser definida como um processo sistemático de desenvolvimento de requisitos através de processos iterativos de análise do problema, documentação das observações resultantes em uma variedade de formatos de representação e de checagem da precisão do entendimento obtido.

O objetivo desta denominação é ressaltar que o processo de definir requisitos de software é uma atividade extremamente importante e independente das outras atividades de Engenharia de Software. Ela requer fundamentação e processos próprios e que devem ser planejados e gerenciados ao longo de todo o ciclo de vida do software.

Para BOEHM (1989), a Engenharia de Requisitos é uma disciplina cujo objetivo é desenvolver uma especificação completa, consistente e não ambígua servindo de base para um acordo entre todas as partes envolvidas no processo de desenvolvimento do software, onde se pactue de forma concisa, “o quê” o produto irá fazer.

Para DAVES (1993), a Engenharia de Requisitos corresponde à atividade de entendimento das necessidades do usuário no contexto do problema a ser resolvido, bem como as limitações impostas na solução.

Após as definições apresentadas, na próxima seção será caracterizado o processo de Engenharia de Requisitos.

2.3. Processo de Engenharia de Requisitos

O processo de Engenharia de Requisitos tem sido dividido de várias formas, segundo as visões de diversos autores como KOTONYA & SOMMERVILLE (1998), LEITE (1994) e LOUCOPOULOS & KARAKOSTAS (1995).

Para LEITE (1994), o processo de Engenharia de Requisitos é dividido em três atividades: Elicitar, Modelar e Analisar. Para o autor, produzir um documento de requisitos completo e consistente – produto da Engenharia de Requisitos implica na necessidade inicial de entender melhor o contexto em que o problema se situa, ou seja, quais são os objetivos do produto a ser desenvolvido, as tarefas/atividades fundamentais e os limites do desenvolvimento.

Já para LOUCOPOULOS & KARAKOSTAS (1995), o processo de Engenharia de Requisitos envolve três atividades principais: Elicitação, Especificação e Validação que correspondem as três preocupações importantes na Engenharia de Requisitos: Compreensão, Descrição e Concordância de um Problema.

Segundo KOTONYA & SOMMERVILLE (1998), o processo de Engenharia de Requisitos é um conjunto estruturado de atividades que devem ser seguidas para Extrair requisitos, Validá-los e Mantê-los. Nas atividades previstas devem constar o levantamento de requisitos, a análise e sua validação.

O processo deve oferecer diretrizes sobre a programação das atividades, definição de responsabilidades, entradas, saídas de cada uma das atividades, entre outras.

De acordo com os autores, no processo de Engenharia de Requisitos não basta apenas buscar informações sobre a funcionalidade dos sistemas, mas é preciso entender melhor o domínio da aplicação e considerar aspectos relativos à própria estrutura da organização.

Para KOTONYA & SOMMERVILLE (1998), os processos de Engenharia de Requisitos são muito variáveis. Partindo de processos poucos estruturados até processos sistemáticos, baseados na aplicação de alguma metodologia de análise.

Quatro fatores fundamentais influenciam o grau de estruturação do processo adotado para a Engenharia de Requisitos: maturidade técnica, envolvimento disciplinado, cultura organizacional e domínio da aplicação. Por melhor que seja um processo é quase impossível defini-lo como único para todos os casos.

O modelo de processo mais utilizado na Engenharia de Requisitos envolve quatro atividades:

1- Elicitação de Requisitos

Nesta atividade o Engenheiro de Requisitos procura capturar os requisitos do sistema, buscando também obter um conhecimento do domínio do problema.

Segundo BUBENKO (1993), apesar de na prática ser freqüentemente utilizado, o termo “captura de requisitos” é errôneo porque indica que os requisitos “já estão lá” e que tudo que é necessário são dispositivos para extrair os requisitos dos usuários. Na prática, segundo o autor, essa não é a realidade, são necessários conceitos, métodos e ferramentas que podem ajudar a analisar a situação, para projetar (ou reprojetar) processos de negócios, e para desenvolver e definir os requisitos do sistema de informação passo a passo junto com usuários, sem perder detalhes e a visão do todo.

Algumas equipes ao iniciar o desenvolvimento de sistemas acham que a elicitação de requisitos é fácil de se obter, portanto, não dedicam tempo e recursos suficientes para sua execução. Embora a tarefa pareça relativamente simples, o processo de elicitação de requisitos requer uma análise criteriosa da organização, compreendendo: a) a definição do alvo e da abrangência do domínio da aplicação, b) o entendimento do foco no problema a resolver (o quê, para quê e para quem), c) a identificação de processos do negócio e, principalmente, d) o conhecimento da informação do cliente relativa as suas necessidades ou desejos e exigências. A seguir vamos explicar cada item do processo de elicitação de requisitos:

a) Domínio da Aplicação ou Ambiente

O primeiro passo em análise de problemas, segundo JACKSON (1995), é estruturar e analisar o domínio da aplicação. O princípio de relevância do domínio declara que “Tudo o que é relevante para os requisitos, deve aparecer em alguma parte do domínio da aplicação”. Existe uma importante consequência deste princípio. O domínio da aplicação não é limitado a partes do mundo que estão diretamente ligadas ao domínio da máquina. É mais abrangente, pois se refere ao negócio.

O domínio da aplicação é onde os requisitos particulares dos clientes são encontrados. Se o domínio da aplicação não for identificado corretamente, não se está apto para focalizar os requisitos dos clientes.

A identificação do ambiente ou domínio da aplicação onde está inserido o problema que se quer tratar é o passo inicial para o planejamento das atividades de descobrimento inerentes ao contexto organizacional

b) Problema a ser Resolvido

Um problema ao ser tratado por pessoas agrega sempre características inerentes ao fator humano do querer, do saber, do poder e, principalmente, da comunicação e do entendimento do requisito.

Ainda persiste na área de Engenharia de Software a tentativa de visualizar a tecnologia como solução de problema, sem antes focar intensivamente o esforço em definição ao entendimento do problema e na negociação de eventuais conflitos de interesses pela solução.

A falta de habilidade para discutir problemas tem sido uma das mais flagrantes deficiências da teoria e da prática em software. Muitos escritores sobre métodos de desenvolvimento afirmam oferecer uma análise de problema, quando de fato oferecem somente um contorno de solução, deixando o problema inexplorado e inexplicável (JACKSON, 1995).

O problema no contexto de extração de requisitos é a razão principal para o entendimento, a especialização e o domínio do conhecimento. Identificar a designação do que é problema, qual é a definição do problema, quem tem o problema e qual a essência do problema sob o ponto de vista de quem o tem, caracterizam a complexidade do processo. Fatos como determinar o que está errado e o que pode ser feito acerca do erro ou como identificar o problema em função da diferença entre o que é desejado pelo cliente e o que é percebido pelo engenheiro de software agrega um esforço considerável ao processo de descobrimento.

Enfim, é necessário distinguir claramente um processo de definição do problema (conhecimento dos requisitos) de um processo de solução do problema (aplicação de ferramentas de software como solução).

c) Contexto do Negócio

Processos organizacionais e fatores políticos geralmente podem influenciar nos requisitos. Nas organizações, a luta pelo poder e relações de influência entre diferentes pessoas dificultam um denominador comum quanto à propriedade da informação e à definição da melhor forma de administrá-la. Cabe aí uma negociação cuidadosa e a delimitação de fronteiras, com o estabelecimento de objetivos, o entendimento do histórico e da estrutura organizacional e da própria organização do conhecimento.

Um fator expressivo de conflito também ocorre em ambientes descentralizados geograficamente, em relação ao desenvolvimento dos processos organizacionais. O desconhecimento das peculiaridades regionais pelo poder centralizado, as dificuldades de comunicação e a falta de padronização no tratamento da informação tornam mais complexo o

processo de validação de requisitos, quando estes tendem a ser necessidades comuns ou até mesmo particulares.

d) Informação do Cliente

A fonte de informação cliente é composta por uma variedade de requisitos, expressa de acordo com a posição que a pessoa ocupa na organização e o seu nível de interesse e de comprometimento na identificação de problemas e na procura de solução.

O posicionamento do cliente como consumidor e/ou produtor como formador de opinião para definição do problema em estudo é o passo inicial do processo, especialmente no esclarecimento do nível de exigência do requisito sob o enfoque de grandeza do que é essencial para a organização, acima dos interesses particulares. É claro que o ponto de vista particular do cliente é importante, mas no contexto organizacional este pensamento pode ser diluído em confronto com os demais posicionamentos sobre o requisito, considerando-se a necessidade de se obter a melhor representatividade de opiniões para a definição do problema.

2- Análise de Requisitos

O objetivo desta fase é a obtenção de uma especificação que seja consistente e completa. Os requisitos são analisados detalhadamente e os representantes de cada grupo de usuários devem negociá-los de forma a chegar a um consenso. Esta fase é necessária, devido às divergências que são inevitáveis por limitações como: orçamentos, prazo, inconsistências oriundas de informações fornecidas por diferentes fontes e diferentes perspectivas de cada um dos grupos de usuários (a maneira como cada grupo imagina o sistema).

O engenheiro de requisitos durante a análise deve ser capaz de detectar e resolver inconsistências.

3- Modelagem e Documentação de Requisitos

Os requisitos são documentados e formalizados em forma de modelos conceituais que descrevem estaticamente e dinamicamente aspectos do problema e do domínio da aplicação, cujo entendimento deverá ser comum a todos os usuários.

A estrutura estática dos conceitos é representada por redes de objetivos, que encapsulam atributos. Estruturas dinâmicas descrevem o comportamento do sistema de acordo com estímulos externos e o passar do tempo.

4-Validação dos Requisitos

Quando validamos uma especificação, estamos garantindo que ela de fato retrata as necessidades do cliente. Se a validação não for adequada, mal entendidos e erros serão propagados para as fases de projeto e implementação. Nesta atividade deve-se identificar possíveis problemas nos requisitos, antes que o documento produzido sirva de base para o desenvolvimento do sistema. Portanto, deve haver uma cuidadosa avaliação dos requisitos, com ênfase em sua consistência e completitude.

Independente do processo de Engenharia de Requisitos utilizado o resultado é um documento de requisitos do software. Este documento algumas vezes chamado de especificação de requisitos de software é a declaração oficial do que é requerido dos desenvolvedores do sistema.

Geralmente um documento de requisitos de software utiliza modelos que descrevem estática e dinamicamente o que o sistema deve fazer. Estes modelos expressam os requisitos descritos no documento de requisitos, possibilitando um maior entendimento do domínio da aplicação, servindo para determinar se a especificação está completa, consistente e precisa.

2.4. Problemas Existentes na Engenharia de Requisitos

A atividade de Engenharia de Requisitos requer uma intensa atividade de comunicação. Durante a comunicação, problemas de omissão e má interpretação podem causar dificuldades entre o engenheiro e o cliente.

Freqüentemente informações obtidas de usuários conflitam com requisitos descritos anteriormente por outras pessoas. Nestes casos é preciso negociar uma solução para o impasse. A qualidade de negociação depende de um bom entendimento e de uma análise rigorosa.

Grandes sistemas de software usualmente possuem uma clientela bastante heterogênea. Diferentes clientes desfrutam de variadas prioridades e diferentes requisitos possuem diversos graus de importância

Difícilmente quem encomenda (responsável pelo pagamento) o sistema será o usuário (principal) do sistema. É comum a imposição de requisitos, que são devidos à questão organizacional ou limitações financeiras da empresa. Estas solicitações podem entrar em

conflito com os requisitos dos usuários. Portanto as visões parciais dos clientes muitas vezes não são consistentes entre si.

A evolução dos sistemas é outro aspecto de suma importância. Independente do estágio de desenvolvimento de um sistema, mudanças ocorrerão nos requisitos. Portanto, devemos abordar a questão da coordenação das mudanças dos requisitos, seu impacto em outras partes do software e como corrigir erros de especificação de forma que efeitos colaterais sejam minimizados ou evitados.

Muitas vezes as mudanças são realizadas apenas no código de implementação, causando divergências entre o sistema de software implementado e sua especificação de requisitos.

A produção de uma boa especificação de requisitos não é tarefa fácil tendo em vista os problemas expostos acima

Assim, entre as metas da Engenharia de Requisitos estão:

- Criação de técnicas de comunicação que visem facilitar a aquisição de informação;
- Desenvolvimento de técnicas e ferramentas que resultem em especificações de requisitos adequadas e precisas;
- Consideração de alternativas na especificação de requisitos.

2.5. Classificação de Requisitos

Os requisitos de software são obviamente aqueles dentre os requisitos do sistema que dizem respeito às propriedades do software. Tradicionalmente, os requisitos de software são separados em requisitos funcionais e requisitos não funcionais.

Contudo, trabalhos recentes identificam mais uma categoria muito importante, os chamados requisitos organizacionais (KOTONYA & SOMMERVILLE (1998), DAVIS (1993), YU (1995) e BUBENKO (1993)).

Os requisitos funcionais são a descrição das diversas funções que clientes e usuários querem ou precisam que o software ofereça. Eles definem a funcionalidade desejada do software e descrevem as transformações do sistema ou de seus componentes que transformam entradas para produzir saídas. Por exemplo: O software deve emitir relatórios de compras a cada 15 dias.

A especificação de um requisito funcional deve determinar “o que” se espera que o software faça, sem a preocupação de “como” ele faz.

Os requisitos não funcionais são as qualidades globais de um software, como manutenibilidade, usabilidade, desempenho, custos entre outras. Normalmente são descritos de maneira informal, controversa, e difíceis de validar. Por exemplo: a base de dados deve ser protegida para acesso apenas de usuários autorizados.

De forma geral, a diferença entre requisitos funcionais e não funcionais está no fato dos primeiros descreverem o que o sistema deverá fazer enquanto os outros fixam restrições sobre como os requisitos funcionais serão implementados (ALENCAR, 1999).

Os requisitos funcionais e os requisitos não funcionais são derivados dos requisitos organizacionais, que por sua vez são definidos em função do domínio da aplicação.

Os requisitos organizacionais dizem respeito às metas da empresa, suas políticas estratégicas adotadas, os empregados da empresa com seus respectivos objetivos, os processos, informações, requisitos não funcionais, enfim representam a estrutura da organização.

2.6. Considerações Finais

O sucesso do processo de desenvolvimento de software depende significativamente da definição de requisitos.

Uma definição precisa do que produzir é condição determinante para abonar o sucesso de todo o processo de produção do software. A definição apropriada dos requisitos depende de um preciso conhecimento do ambiente e das razões pelas quais o software é proposto.

Muito dos problemas oriundos de sistemas mal desenvolvidos são causados por falhas ocorridas durante a definição de requisitos, resultando na produção de um sistema que não satisfaz as necessidades do cliente.

A definição de requisitos não é uma atividade desenvolvida de forma simples e direta. É uma atividade interativa com presença do elemento humano e de muita comunicação sendo necessário à integração de pessoas, ferramentas e informações.

As técnicas tradicionalmente aplicadas no desenvolvimento de software tratam aspectos relacionados à funcionalidade do sistema, sem considerarem aspectos mais amplos como objetivos da organização, regras do negócio, políticas, estratégias, entre outros.

De acordo com a literatura pesquisada, não basta apenas buscar informações sobre a funcionalidade dos sistemas, no processo de Engenharia de Requisitos, mas é preciso considerar aspectos mais amplos como objetivos da organização, regras do negócio, políticas, estratégias,

entre outros. Estes aspectos são extremamente importantes para um entendimento completo e correto dos requisitos sendo necessário, portanto, considerá-los.

Neste contexto, é importante ressaltar a necessidade de se utilizar uma outra técnica para entender melhor o domínio da aplicação e os aspectos relativos a estrutura da organização. Dessa forma pretende-se no capítulo 3 (três) realizar uma revisão bibliográfica sobre a técnica de Modelagem Organizacional – EKD e no capítulo 4 (quatro) revisar a abordagem *Use Case*.

Tanto o *Use Case* como o EKD, são técnicas que permitem focar os requisitos de software, porém sob perspectivas diferentes.

O Modelo *Use Case* é um modelo criado com o fim de representar *use cases* (casos de uso) do sistema e conseqüentemente os requisitos do mesmo. Já a Modelagem Organizacional, implicitamente permite que as informações consideradas da organização possam ser utilizadas tendo um enfoque voltado para a descrição do sistema.

3. Modelagem Organizacional

3.1. Considerações Iniciais

O mercado que envolve as organizações é bastante dinâmico. Para que as organizações sejam mais competitivas elas precisam estar dispostas a se modificarem continuamente e só poderão fazer isso se seus softwares forem capazes de se adaptarem rapidamente. Esse raramente é o caso, e muitas organizações ficam estagnadas por softwares imutáveis.

A Modelagem Organizacional propõe soluções voltadas para esse problema fazendo uma definição de requisitos voltada para o sistema de informação da organização, seus objetivos, atividades, conceitos e atores. Dessa forma se obtém uma melhor integração entre as atividades da organização e o sistema de informação, proporcionando um sistema que realmente atenda as necessidades do cliente.

Neste sentido, este capítulo tem como objetivo apresentar a técnica de Modelagem Organizacional – EKD, como uma técnica que facilita a compreensão do ambiente empresarial, tornando-se desta forma uma atividade fundamental para a Engenharia de Requisitos.

3.2. Contextualização da Modelagem Organizacional

Segundo PÁDUA (2001), existe um relacionamento natural e lógico entre o negócio e os requisitos do sistema, sendo assim, o Modelo Organizacional é parte importante do desenvolvimento de software para captura e especificação dos requisitos, nos quais a determinação explícita dos objetivos, problemas, conceitos, atividades, processos e atores direciona para um sistema que atende às reais necessidades do cliente, além de diminuir custos de manutenção.

A definição de requisitos é uma fase crítica no desenvolvimento de sistemas. Erros não identificados nessa fase são caros de corrigir e difíceis de serem rastreados em fases futuras.

Para o sucesso do desenvolvimento de sistemas é necessário se ter certas propriedades na definição de requisitos. Entre estas podemos citar: requisitos os mais completos possíveis, corretos, não ambíguos, consistentes, modificáveis, de fácil entendimento e rastreáveis.

Entender a organização é importante para Engenharia de Requisitos porque muitos problemas na definição de requisitos são originados devido ao pouco conhecimento das atividades da organização e de seu Sistema de Informação. Exemplos desses problemas são requisitos conflitantes, limitações aparentemente impostas por requisitos, e retorno ao processo de desenvolvimento de software para entender melhor os requisitos.

Cada organização tem sua missão, seus objetivos e seus processos próprios e é importante dar atenção a modelagem desses processos.

Em uma organização as informações são originadas de várias fontes como pessoas, documentos e softwares. Pessoas executam atividades, possuem objetivos dentro da organização e para isso lidam com um conjunto de informações. Documentos possuem informações variadas como ordens, solicitações e comunicados. No código dos softwares existem várias informações da organização como regras, limitações e fórmulas de cálculos. Muitas informações também estão presentes na base de dados utilizados pelos softwares.

A elaboração do Modelo Organizacional deve acontecer de forma planejada e estruturada. A meta é obter requisitos abrangendo o máximo possível de fontes de informações. Depois de elaborado o Modelo Organizacional, ele próprio servirá de fonte de informação para diferentes grupos de usuários.

Segundo SOUZA & CASTRO (1998) a Modelagem Organizacional separa e representa o conhecimento em submodelos (partes), facilitando a leitura e a compreensão por diferentes grupos de usuários. A partir da compreensão verifica-se consequência benéfica para a definição de requisitos porque quando se entende o que se lê surgem naturalmente críticas e sugestões ao que está se representando, proporcionando um Modelo Organizacional completo e consistente.

ALENCAR (1999) destaca os seguintes objetivos da Modelagem Organizacional:

1. Fornecer um objetivo que seja uma representação compartilhável e reusável da cadeia de fornecimento de informação e conhecimento;
2. Suportar tarefas da cadeia de fornecimento, pela habilitação de respostas a questionamentos que não estão explicitamente representados no modelo;
3. Definir os objetivos de maneira precisa, de forma que sejam consistentemente aplicados, através dos domínios e interpretados pelos usuários;
4. Suportar visualização do modelo, de forma intuitiva, simples e consistente.

Existem alguns métodos na literatura com o objetivo de definir requisitos a partir da visão da organização. Neste trabalho nos concentramos em especial nos modelos existentes na

técnica de BUBENKO – EKD. No nosso departamento já temos trabalhos da modelagem de BUBENKO e da técnica I* de YU.

A técnica de modelagem de BUBENKO (1993) e o método EKD foram apresentados no trabalho de PÁDUA (2001), proporcionando uma boa revisão bibliográfica para os pesquisadores.

A técnica I* foi apresentada no trabalho de CHICHINELLI (2002), enfatizando a contribuição da técnica ao processo de Engenharia de Requisitos.

3.3. O Modelo Organizacional

A modelagem organizacional iniciou-se nos anos 80 pelo projeto Plandata e refinado pelo SISU¹ (*Swedish Institute for Development* – Instituto Sueco para o Desenvolvimento de Software) no final os anos 80. A grande contribuição foi à noção de considerar componentes organizacionais de uma especificação, por exemplo, os objetivos (intenções de um negócio, além dos tipos de componentes do modelo tradicional, como entidade, relacionamentos e processos). O uso dessa abordagem em muitas aplicações diferentes durante os últimos dez anos mostrou que a razão do sucesso não era apenas o Modelo Organizacional, mas também o gerenciamento apropriado do processo do negócio e engenharia de requisitos. A idéia do Modelo de Negócios do SISU era, no final, estendida para o Modelo Organizacional e mais tarde desenvolvida no projeto ESPRIT F3 (From Fuzzy to Formal). A modelagem organizacional F3 foi então elaborada pelo projeto ESPRIT ELKD e está agora sendo aplicada no projeto ESPRIT ELEKTRA (*Electrical Enterprise Knowledge For Transforming Applications*). A estrutura de modelagem ELEKTRA é denotada EKD (*Enterprise Knowledge Development*) que inclui a Modelagem Organizacional como parte (BUBENKO *et al.*, 1998).

O projeto ELEKTRA visa principalmente aplicação do método EKD para problemas de gerenciamento de mudanças dentro de organizações da Grécia e Suécia, gerando um conjunto de práticas genéricas de forma a aplicá-las em outras companhias. Para ROLLAND *et al.* (2000), a pesquisa do projeto ELEKTRA visa a criação de uma base de conhecimento para o

¹ O Instituto Sueco de Desenvolvimento de Software foi fundado em 1984 e realizou mais de 100 projetos de análise de sistemas e negócios, nos quais a metodologia de modelagem organizacional foi usada.

gerenciamento de mudança. Um dos objetivos do projeto era: capturar as melhores práticas do negócio para reusá-las em situações similares em outras companhias de eletricidade.

3.4. Conhecimento Organizacional

As descrições estruturadas tornam possível ver como diferentes partes de uma organização interagem. Pode-se traçar descrições e verificar quais processos estão relacionados, e como contribuem para um particular objetivo. Inversamente, é possível verificar se existem objetivos para os quais nenhum processo está contribuindo. Percebe-se que as descrições estruturadas permitem a realização de diferentes tipos de análises e discussões sobre o que tem sido dito e escrito sobre a organização. Esses modelos podem ser efetivamente usados para representar e discutir como projetar e melhorar processos organizacionais, e como desenvolver estratégias futuras. Entretanto, os modelos são limitados, uma vez que eles apenas apresentam percepções subjetivas de como as coisas são.

Segundo BUBENKO *et al.*, (2001), um grande benefício da modelagem é o efeito nos participantes. Em um projeto bem sucedido, os efeitos podem ser:

- Melhora no entendimento das partes essenciais da organização;
- Encontro de soluções para problemas práticos ou
- Consenso sobre questões que no início eram difíceis de serem definidas.

3.5. O Modelo EKD

O EKD é uma metodologia que fornece uma forma sistemática e controlada de analisar, entender, desenvolver e documentar uma organização e seus componentes, usando a Modelagem Organizacional (ROLLAND, 2000).

O objetivo ao se usar o EKD é prover uma descrição clara e não-ambígua de

- Como a organização funciona atualmente;
- Quais são os requisitos e as razões para a mudança;
- Quais alternativas deveriam ser criadas para encontrar esses requisitos;
- Quais são os critérios e argumentos para avaliação dessas alternativas.

Segundo KIRIKOVA (2000), o EKD pode ser usado em situações diferentes e com propósitos diferentes, como nas seguintes situações:

- Na engenharia de requisitos para definição e especificação de requisitos.
- Na análise do negócio para detecção do problema.
- Na reengenharia de processos do negócio para definição de novos sistemas de negócio.
- No gerenciamento de conhecimento organizacional ou aprendizagem organizacional para formar a base de propagação e ampliação de conhecimento.

A família de modelos EKD destina-se a responder as questões: O que, Como, Onde, Quem, Quando e Por que. De acordo com BUBENKO *et al.* (2001), o conteúdo básico da estrutura EKD (figura 04), inclui:

- a) um conjunto de técnicas de descrição;
- b) a participação de *stakeholders*¹; e
- c) um conjunto de diretrizes para o trabalho.

O conjunto de técnicas de descrições fornece um conjunto de modelos que é usado para descrever o sistema a ser analisado ou construído, e a organização na qual ele será operado. Esse conjunto de técnicas é usado pelos desenvolvedores do sistema. As técnicas de descrições sozinhas não serão muito valorosas, sem um envolvimento direto dos atuais clientes, usuários finais, gerentes, proprietários, entre outros. Um fator crítico de sucesso de um projeto que inclui a aplicação de EKD na construção de um sistema de informação ou reestruturação da organização é a participação e o envolvimento dos *stakeholders*.

Para KIRIKOVA (2000), o EKD prescreve o desenvolvimento da especificação de requisitos como um escopo de atividades do grupo de trabalho.

O resultado desse trabalho é o Modelo Organizacional que reflete a organização e seus requisitos. Uma característica essencial da metodologia é a possibilidade de ver como cada requisito é relacionado aos objetivos, atividades, pessoas e particularidades da organização.

¹ *Stakeholders*: esse termo foi introduzido para designar todos os envolvidos no projeto, diretamente ou indiretamente, ou que tenham interesse no resultado do projeto

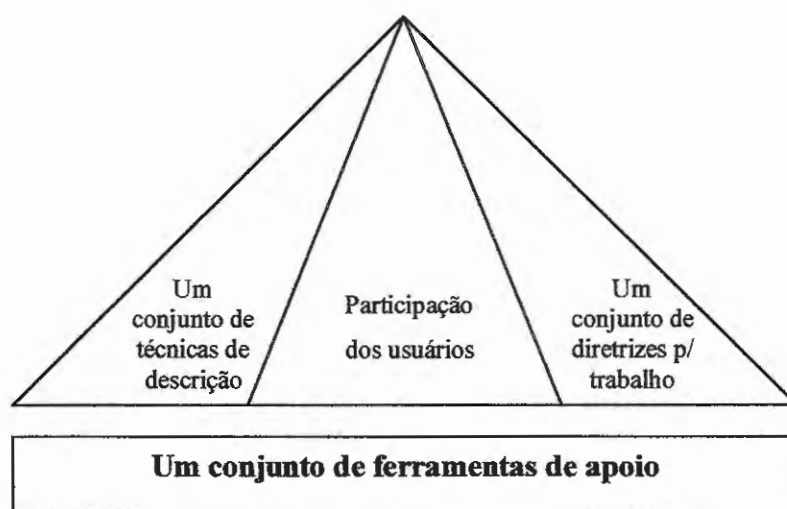


Figura 04 – Conteúdo da estrutura do EKD

Fonte: BUBENKO *et al.* (2001), p 21.

De acordo com ROLLAND *et al.* (2000), durante o desenvolvimento do conhecimento organizacional, desenvolvedores e *stakeholders* se deparam com situações que requerem à tomada de decisões para atingir suas intenções. O EKD ajuda o processo de conhecimento organizacional através de seus modelos que proporcionam várias visões da organização.

Segundo LOUCOPOULOS *et al.* (1998), o EKD pode ser visualizado em três planos ou níveis:

1. Objetivos Organizacionais;
2. Processos Organizacionais e;
3. Sistemas de Informação.

Ao usar o EKD é possível iniciar em um nível e mover-se para outros níveis, dependendo da situação. Por exemplo, se a organização não é bem conhecida, ou existe a necessidade de documentação apropriada da empresa, o ponto mais apropriado é o nível de Processos Organizacionais. Realizar a modelagem nesse nível mostrará uma fotografia clara dos processos organizacionais em termos de tarefas que os atores executam para completar suas obrigações na organização; como essas tarefas podem interagir cooperativamente para atingir seus objetivos; quais os tipos de atividades que eles precisam empenhar em suas obrigações; quais objetivos são necessários para essas atividades (recursos físicos, objetos de informação); e

finalmente; quais regras impõem os processos organizacionais. Essa é uma abordagem *bottom-up* em um nível organizacional.

Uma visão mais estratégica é a *top-down* que se inicia com os objetivos organizacionais e depois procede com a modelagem de como esses objetivos podem ser atingidos nos processos organizacionais. Essa é uma situação de engenharia tradicional, onde os requisitos e visão deveriam ser definidos no plano dos objetivos do negócio, define-se as características operacionais que ocasionarão, por sua vez modelos do plano de sistemas de informação. Outra situação é quando existe a necessidade de entender o sistema de informação já implantado, as tecnologias usadas e como esses sistemas estão relacionados com o processo organizacional. Nesse caso uma abordagem de modelagem reversa será adotada para mapear a construção do sistema de informação nos conceitos dos processos organizacionais (BUBENKO *et al.*, 1998).

De acordo com BUBENKO *et al.* (1998), o termo “processo organizacional” precisa de alguma elaboração nesse estágio. A visão tradicional de processos de negócio é a visão funcional.

Uma visão funcional de uma organização considera funções do negócio relacionado de uma forma hierárquica. Essa mesma hierarquia é construída através de uma decomposição severa de funções de negócio. Tipicamente, a ligação é feita através de sucessivos detalhamentos de funções em sub-funções, sub-sub-funções até um nível de detalhe que esteja de acordo com as idéias e opiniões dos analistas. Funções são ligadas via fluxo de dados. Essa não é a visão no EKD que considera o conceito de “processo organizacional” como composto de quatro componentes organizacionais chaves:

1. As tarefas que são executadas pela organização para encontrar os objetivos dos processos;
2. As atividades envolvidas em cada tarefa;
3. Os objetivos que são envolvidos juntos com sua evolução da criação para extinção (com o contexto de um processo organizacional);
4. As regras que determinam os componentes de processo

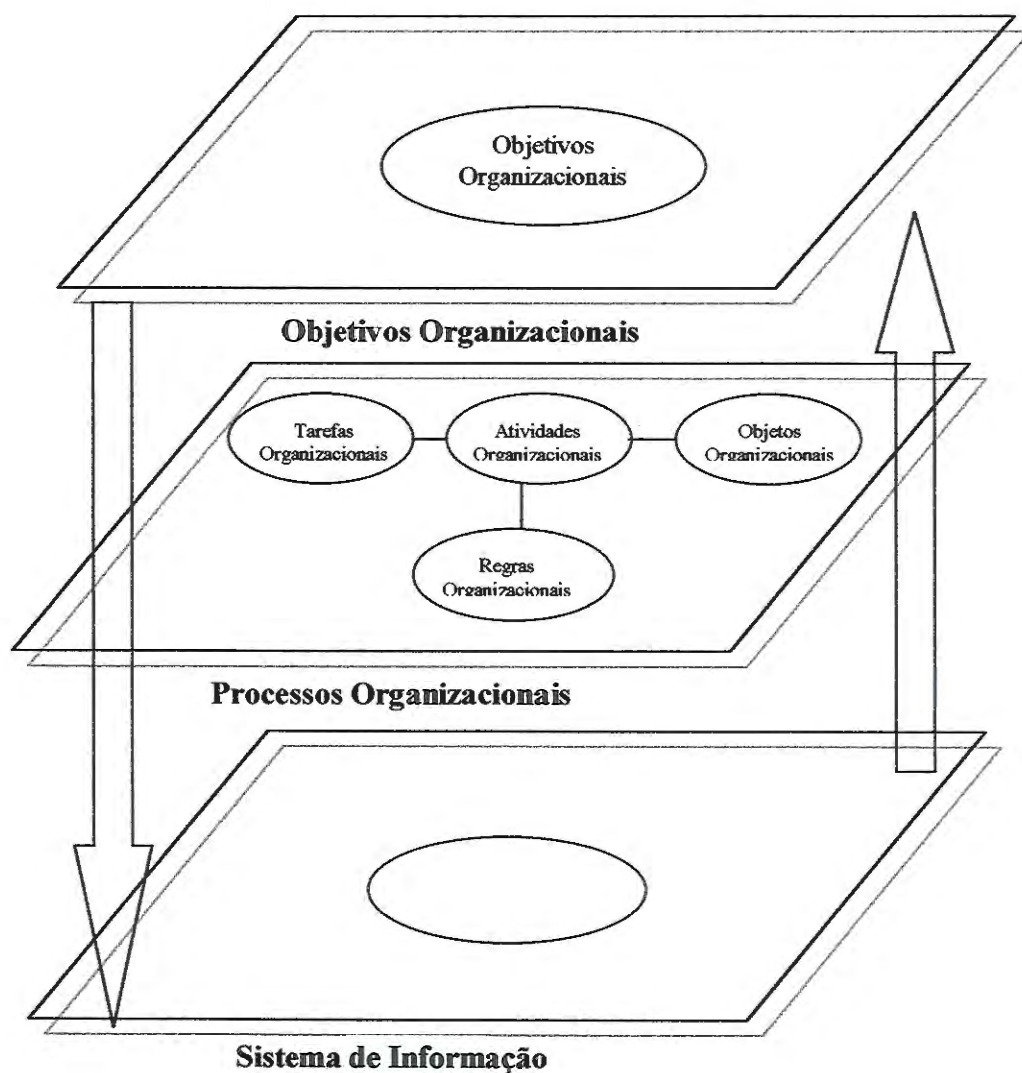


Figura 05 – Uma visão da Modelagem EKD

Fonte: BUBENKO *et al.* (1998), p.3.

Em outras palavras, segundo BUBENKO *et al.* (1998), um processo organizacional pode ultrapassar uma divisão funcional, o que proporciona uma visão verdadeiramente dinâmica da organização. Os modelos usados no método EKD enfocam os seguintes pontos:

1. Modelo de Objetivos das situações AS_IS e TO_BE (como é e como será), porque é importante conhecer o que a organização correntemente trabalha e o que pretende atingir no futuro.

2. Modelo de Atores e suas tarefas, porque na realidade, em um nível operacional, pessoas trabalham juntas para completar os requisitos de um processo do negócio. Os comportamentos detalhados da tarefa de um indivíduo e suas interações forneceriam um critério calculável para os processos do negócio.
3. Modelo de objetos envolvidos no processo e usado pelos atores para cumprir suas tarefas na organização, porque tal conhecimento levará à definição dos recursos e sistema de informação que são necessários para apoiar a forma que os atores completam suas tarefas.
4. Modelo de Regras do Negócio relacionado aos processos de negócio, porque condições diferentes vão requerer a invocação de processos de negócio diferentes. São as regras do negócio que determinam os estados permissíveis dos objetos do negócio e a interação entre tarefas diferentes.

De forma resumida, os modelos EKD são usados para entender, analisar, melhorar e consertar algum sistema. Os *stakeholders* têm objetivos que precisam ser encontrados pelo sistema. Sistemas são compostos de partes de interface e interdependentes, que trabalham juntas para realizar uma função útil. As partes do sistema podem ser uma combinação de coisas, incluindo pessoas, informações, softwares, equipamentos, produtos ou matéria prima. Na realidade, modelos operacionais deveriam descrever o que o sistema faz, controla, baseia-se, o que usa para realizar sua função, e o que produz. Os componentes do método EKD são definidos em termos de conceitos que constituem a abordagem e a notação que habilita o uso desses conceitos. O conjunto total de conceitos do EKD é denominado metamodelo EKD que define a forma lógica do conhecimento organizacional. O metamodelo inclui informações sobre a semântica do conhecimento organizacional, identificando as entidades organizacionais e seus atributos e relacionamentos explícitos entre eles.

É importante distinguir metamodelos EKD e notações EKD. O metamodelo representa meios para definir a informação que se deseja manter sobre aspectos diferentes complementares do sistema. Para cada aspecto do sistema pode existir apenas um metamodelo EKD. Em contraste, a notação EKD é o meio para apresentar o EKD metamodelo; cada metamodelo pode ser visto em termos de muitas notações diferentes. Por exemplo, no Modelo de Objetivos o conceito de objetivo tem uma definição única. Entretanto, é permitido usar muitas notações para descrever incluindo texto, caixas e setas entre outros.

É importante escolher notações que segundo BUBENKO *et al.*, (1998):

- Sejam fáceis de usar: os principais objetivos dessas notações são serem compreendidos por todos usuários EKD, sem que se requeira muito treinamento.
- Clareza: cada metamodelo contém extensa informação sobre as entidades organizacionais. Apresentar toda essa informação em um simples diagrama, seria inviável. O objetivo da notação proposta é prover uma estrutura apropriada para informações EKD. Cada notação concentra em conceitos específicos que formam o esqueleto do metamodelo, dando imagem da situação “a primeira vista”, enquanto impede detalhes que poderiam confundir o usuário. Naturalmente, é possível melhorar esse esqueleto com o restante da informação ou, idealmente, usar uma ferramenta que ajudará a armazenar e explorar toda a informação disponível.

3.5.1. Benefícios do EKD

Segundo BUBENKO *et al.* (2001), os benefícios do EKD são:

- Entendimento do negócio;
- Aprendizagem e comunicação organizacional sobre questões essenciais;
- Entendimento das capacidades e processos organizacionais;
- Melhor comunicação entre o sistema de informação *stakeholders* e desenvolvedores;
- Desenvolvimento de uma descrição estruturada do negócio para analistas da organização discutirem;
- Auxilia os desenvolvedores de sistemas de informação e *stakeholders* na determinação dos requisitos e objetivos do sistema;
- Descrição dos objetivos da organização, entidades, processos, requisitos, que é mais consistente e mais completo do que usando a tradicional abordagem em textos;
- Criação de um documento chamado repositório do conhecimento, a qual pode ser usado para:
 - Raciocínio sobre o negócio;
 - Discussão de mudanças no negócio e componentes do sistema de informação;
 - Construção da cadeia de componentes e decisões, que leva a várias interpretações de decisões e componentes do sistema de informação.

3.6. Componentes do EKD

Os componentes do EKD são modelos conceituais que examinam a organização e seus requisitos de um número de perspectivas inter-relacionadas. Esses modelos são abstrações do mundo físico. Para uma dada organização, esses modelos irão constituir coletivamente o Modelo Organizacional. Alguns desses modelos podem conter informações relevantes que apontam a necessidade de avaliação de alternativas de situações operacionais. Tais informações incluem critério para avaliação, escolhas disponíveis, parâmetros de medidas e argumentos registrados a favor e contra escolhas (ROLLAND *et al.*, 2000). X

Durante o estágio de desenvolvimento, o modelo organizacional pode ser um meio de entendimento e comunicação entre os participantes ou entre participantes e outros *stakeholders* do processo EKD, tornando-se um ponto de referência comum, cruzando muitas áreas diferentes a fim de que sua posse não seja confinada a aplicações específicas ou grupos particulares. Além disso, deve ser independente de qualquer tecnologia, podendo ser implementado por diversas tecnologias.

O modelo precisará ser modificado apenas quando a situação e o contexto nos quais a organização existe mudar. O modelo pode ser usado como um meio de avaliação das opções, no qual os custos de cada opção potencial podem ser avaliados tão bem quanto mais clara estiver a documentação de todos os aspectos intangíveis (LOUCOPOULOS *et al.*, 2001).

Segundo ROLLAND *et al.* (2000), o EKD apóia a construção de modelos diferentes, representando o estado inicial da organização e o futuro estado, além de expressão de estratégias alternativas para mudanças, a avaliação dessas estratégias, e outras atividades como *brainstorming*. A “regra de parada” para o desenvolvimento do Modelo Organizacional é um conjunto de modelos descritos em detalhes suficientes a fim de que possa ser usado como uma base para implementar as sugestões nos modelos, ou para resolução de problemas por outros meios.

A abordagem EKD envolverá tipicamente estrategistas, gerentes táticos e funcionários do nível operacional, que juntamente com o facilitador e técnicos, familiarizados com EKD, iniciarão o processo de:

- Diagnóstico: modelar a situação corrente e os requisitos de mudanças;
- Entendimento: interpretar, entender, raciocinar, deliberar e discutir o estado corrente e futuro da empresa;
- Projeto: discutir e modelar as situações alternativas futuras e os cenários.

O modelo resultante estará disponível para os tomadores de decisões atuarem sobre as estratégias futuras, táticas e objetivos da empresa. O Modelo Organizacional contém um número de submodelos inter-relacionados. Cada um representa algum aspecto da organização.

De acordo com BUBENKO *et al* (2001), os tipos de submodelos e as questões que eles abordam são:

1- Modelo de Objetivos (MO): enfoca na descrição de idéias da organização. Descreve o que a organização e os empregados querem alcançar ou evitar, e quando. Usualmente esclarece questões como:

- Para onde deveria ser movida a organização;
- Quais os objetivos mais importantes e prioridades desses objetivos;
- Como cada objetivo é relacionado aos outros e quais problemas estão escondidos na realização das metas.

2- Modelo de Regras do Negócio: é usado para definir e manter explicitamente regras do negócio formuladas e consistentes com o Modelo de Objetivos. Regras do Negócio podem ser vistas como operacionalização ou limites dos objetivos. O modelo de Regras do Negócio geralmente esclarece questões como:

- Quais regras afetam os objetivos da organização;
- Quais são as políticas declaradas;
- Como cada regra do negócio é relacionada com os objetivos e
- Como os objetivos serão apoiados pelas regras.

3- Modelo de Conceitos (MC): é usado estritamente para definir “coisas e fenômenos” relacionados a outros modelos. Representa entidades organizacionais, atributos e relacionamentos. Entidades são usadas para definir mais estritamente expressões do Modelo de Objetivos tanto quanto o conteúdo do conjunto de informações do Modelo de Processos do Negócio. O Modelo de Conceitos esclarece questões como:

- Quais entidades ou conceitos são reconhecidos na organização (incluindo seus relacionamentos com objetivos, atividades e processos, e atores);
- Como as entidades são definidas e
- Quais regras do negócio e restrições monitoram esses objetivos e conceitos.

4- Modelo de Processos do Negócio (MPN): é usado para definir processos organizacionais, e a forma pela qual eles interagem e manuseiam a informação e materiais. Um processo de negócio deve consumir as entradas em termos de informação e/ou material e produzir uma saída de informação e/ou material. Em geral o MPN é similar aos tradicionais modelos de diagramas de fluxo de dados (DFD). O Modelo de Processos do Negócio esclarece questões como:

- Quais atividades e processos do negócio são reconhecidos na organização (ou deveriam ser) para gerenciar a organização em concordância com as metas.
- Como os processos de negócio e tarefas deveriam ser realizados (workflows, transição de estado, ou modelo de processos), e quais as informações necessárias.

5- Modelo de Atores e Recursos (MAR): é usado para descrever como diferentes atores e recursos se relacionam, e como eles são relacionados a componentes do Modelo de Objetivos, e a componentes do Modelo de Processo do Negócio. Por exemplo: um ator pode ser responsável por um particular processo no MPN ou o ator pode buscar um particular objetivo no MO. O Modelo de Atores e Recursos normalmente esclarece questões como:

- Quem está ou deveria estar realizando quais processos e tarefas e
- Como está a estrutura de informação e responsabilidade entre os atores definidos.

6- Modelo de Requisitos e Componentes Técnicos (MRCT): torna-se relevante quando a proposta do EKD é ajudar a definir os requisitos para o desenvolvimento de um sistema de informação. A atenção é direcionada para o sistema técnico que é necessário para apoiar os objetivos, processos e atores da organização. Inicialmente é necessário desenvolver um conjunto de requisitos de alto nível ou objetivos para o sistema de informação como um todo. Baseado nesses requisitos, o sistema de informação é estruturado em um número de subsistemas, ou componentes técnicos. O MRCT é uma tentativa inicial de definir toda a estrutura e propriedades do sistema de informação para apoiar as atividades do negócio, como definidos no MPN. O MRCT usualmente esclarece questões como:

- Quais requisitos são gerados pelos processos do negócio e
- Qual o potencial da tecnologia da informação pra melhoria do processo.

Cada um desses submodelos inclui um número de componentes que descreve diferentes aspectos da organização. Por exemplo, o Modelo de Objetivos contém objetivos do negócio,

problemas do negócio que são divididos em tratamento, fraquezas, causas, oportunidades do negócio e restrições.

Os componentes dos submodelos são relacionados entre si (relacionamento intramodelo), tanto quanto com componentes de outros submodelos (relacionamento intermodelos).

A figura 06 mostra o relacionamento entre modelos. A habilidade de traçar decisões, componentes e outros aspectos em todas as partes da organização é dependente do uso e entendimento desses relacionamentos. Quando se desenvolve um modelo organizacional completo, esses relacionamentos entre componentes de diferentes submodelos executam uma tarefa essencial. Por exemplo, declarações do modelo de objetivos permitem que diferentes conceitos sejam definidos no Modelo de Conceitos. Da mesma forma, objetivos do Modelo de Objetivos motivam processos particulares no Modelo de Processos do Negócio. Os processos são necessários para atingir os objetivos declarados. Uma ligação (link) é definida entre um objetivo e um processo. Ligações entre modelos fazem o modelo transparente, mostrando por que certos processos e requisitos do sistema de informação têm sido introduzidos.

De acordo com BUBENKO *et al.* (2001), existem limitações na forma em que os submodelos e seus relacionamentos podem ser povoados. Esses são controlados por um número de regras de consistência estáticas e dinâmicas, que controlam seus estados de transição permissíveis. Os relacionamentos entre modelos são importantes porque permitem a análise e comparação dos elementos organizacionais.

Os componentes do modelo organizacional têm representações gráficas, como caixas retangulares, elipse e outras formas. O relacionamento entre esses componentes é representados com ligações (links). Existem dois textos associados a cada componente: *short name* e *long name*.

O *short name* é usado para numerar e identificar os componentes do modelo deve ser único para servir de referência.

O *long name* é usado para expressar e armazenar a descrição do componente.

A seguir os modelos serão descritos mais detalhadamente. É difícil descrever EKD sem apresentar exemplos, assim será apresentado um caso de uma biblioteca de BUBENKO *et al* (2001).

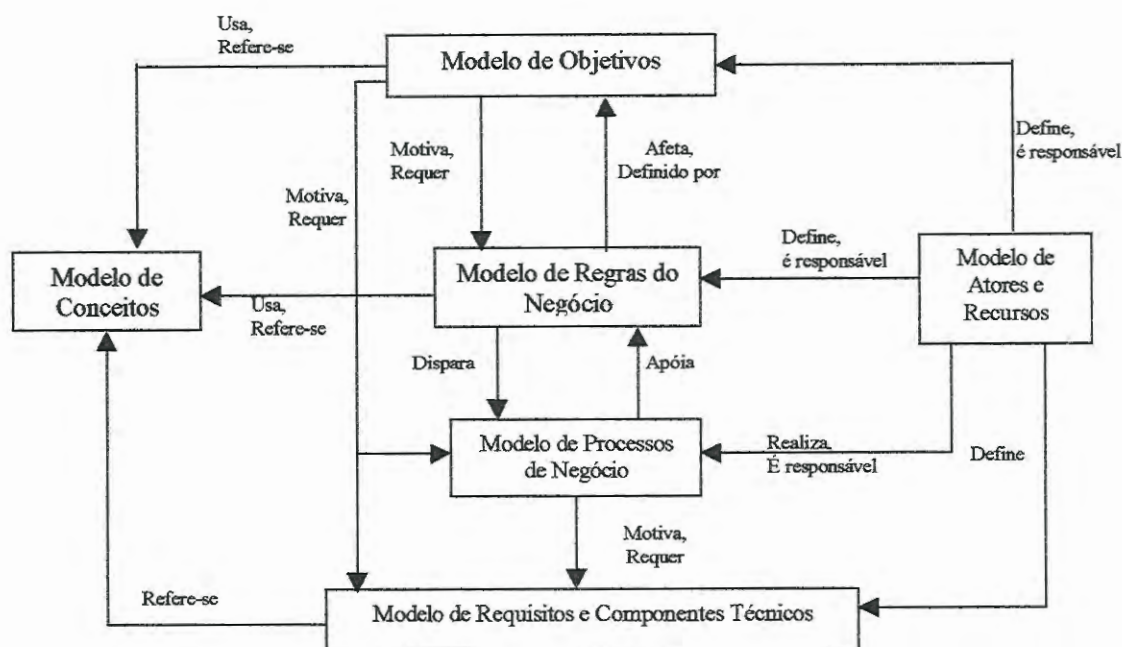


Figura 06 – Submodelos que compõem o Modelo Organizacional

Fonte: BUBENKO *et al* (2001), p.22.

3.6.1. Modelo de Objetivos (MO)

Este Modelo é usado para descrever os objetivos da organização e todas as questões associadas para atingi-los. O Modelo de Objetivos descreve essencialmente a razão, ou a motivação para atividades e entidades de outros submodelos.

As entidades desse modelo são relacionadas à organização e a sua razão de existir. Os objetivos e requisitos do sistema de informação não devem ser declarados nesse modelo. O Modelo de Objetivos forma a estrutura com a qual a relevância dos processos e requisitos técnicos do sistema são medidos e na qual eles estão ligados. Através de ligações com outros modelos, o MO explica porque processos e requisitos existem ou não existem. Os componentes do MO são relacionados através de ligações semânticas monodirecionais, cujos principais tipos são apoio, impedimento e conflitos (BUBENKO *et al.* 2001).

A presença do Modelo de Objetivos fornece uma possibilidade de declarar explicitamente os objetivos da organização, que por sua vez concede aos funcionários a oportunidade de harmonizar seus objetivos com os da organização.

Os componentes do Modelo de Objetivos são: objetivos, problema, causa, restrição e oportunidade. Embora, um número de sessões de modelagem prática mostra que algumas vezes é necessário acrescentar componentes adicionais para o modelo, tais como, comentários, suposições, cenários, tarefas, entre outros. Se tal extensão é aprovada pelo grupo de modelagem, ela pode melhorar consideravelmente a expressividade e clareza do modelo BUBENKO *et al* (2001), explica cada componente:

- O Objetivo do Negócio é o estado desejado do negócio ou estado do negócio que se deseja alcançar. Objetivos expressam o que a organização e seus empregados querem alcançar, ou evitar e quando. Eles podem ser expressos como um conjunto de estados comensuráveis, ou como focos gerais, visões ou direções. Objetivos podem ter muitos significados tais como metas, intenções, necessidades, requisitos, estados desejados entre outros.

As sentenças devem iniciar-se como: “O objetivo é...” Os objetivos podem ter valores opcionais de prioridade e criticabilidade (com possíveis valores: alto, médio, baixo), que permitem às pessoas que estão modelando fixar diferentes níveis de prioridades e notar o nível de criticabilidade. A modelagem de objetivos requer que os participantes reflitam muito, e declarem suas metas a curto e a longo prazo na organização. Para tanto, é necessário que os participantes discutam e concordem sobre: a importância individual dos objetivos, a criatividade dos objetivos, a prioridade dos objetivos e as formas alternativas de atingir os objetivos.

- O Problema é usado para expressar que o ambiente está ou pode ficar em um estado não desejado que dificulte alcançar os objetivos. Para documentar problemas encontrados é criada uma base que detecta objetivos escondidos. Se um problema declarado não está impedindo nenhum objetivo, então o conjunto de objetivos está incompleto ou o problema realmente não é um problema da organização. Problemas podem ser especificados em dois subtipos:

- Ponto fraco: um tipo de problema que descreve fatores que podem reduzir a possibilidade de atingir um objetivo. A organização tem os recursos e conhecimento para reduzir os efeitos do problema.

- Ameaça: um tipo de problema para o qual a empresa tem recursos para reduzir seus efeitos, mas não tem o conhecimento requerido.

- A causa é usada para expressar explicações ou razões para os problemas. Causas são usualmente situações ou estados, fora do controle do projeto, processo, ou da organização. Pode ser alguma coisa que é bem entendida e não precisa mais ser analisada. Tipicamente a causa não pode ser afetada pela organização.

- Restrição é usada para expressar restrições do negócio, leis, regras, ou políticas do mundo externo afetando componentes e ligações com o Modelo Organizacional. Regras internas do negócio e políticas da organização são definidas no modelo de Regras do Negócio
- Oportunidade é usada para expressar recursos que podem tornar certos objetivos mais fáceis de alcançar. Estados alcançáveis não considerados como objetivos, ou mesmo para declarar novos objetivos da organização. Por exemplo, novas tecnologias de comunicação podem facilitar as possibilidades da organização alcançar um objetivo para ampliar o mercado internacional dos seus produtos. Oportunidades são situações em que se deseja obter vantagens. Sendo assim, a oportunidade deveria ser transformada em um objetivo.

3.6.1.1. Ligações dentro do Modelo de Objetivos

Os tipos de ligações, segundo BUBENKO *et al* (2001), entre os componentes do Modelo de Objetivos são (figura 9):

- Relacionamento de Apoio que é essencialmente visto como “vertical”. É usado para refinar ou decompor objetivos e outros componentes.
- Relacionamento de Impedimento que é usado para mostrar influências negativas entre componentes do Modelo de Objetivos, e pode ser considerado como o oposto do relacionamento de apoio.
- Relacionamento de Conflito que é usado para definir situações que ao alcançar um objetivo haverá um conflito com outro. Por exemplo, o objetivo de uma biblioteca em atrair mais clientes ficando mais tempo funcionando, estará em conflito com o objetivo de economizar dinheiro nos custos dos salários dos funcionários.

Inicialmente o Modelo de Objetivos pode ter um alto nível de abstração. Para obter mais clareza e detalhes é necessário decompor ou refinar os objetivos em sub-objetivos. O refinamento é realizado usando relacionamento do tipo AND/OR.

O relacionamento de refinamento AND representa um conjunto de sub-objetivos que são necessários para satisfazer e apoiar o objetivo original. Se o Objetivo 1 for refinado em Objetivo 1.1, 1.2, e 1.3, demonstra-se então, que para atingir o Objetivo 1 é necessário atingir o Objetivo 1.1, 1.2 e 1.3. A figura 07 ilustra o refinamento.

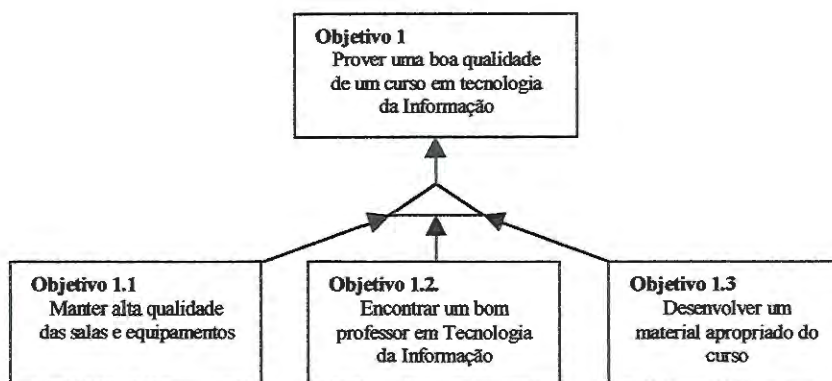


Figura 07 – Fração de um Modelo de Objetivos contendo relacionamento AND.

Fonte: BUBENKO *et al.* (2001), p. 30.

O relacionamento de refinamento OR representa um conjunto de sub-objetivos alternativos. Para apoiar o objetivo original, é suficiente satisfazer apenas um objetivo do conjunto. A figura 08 exemplifica o refinamento OR.

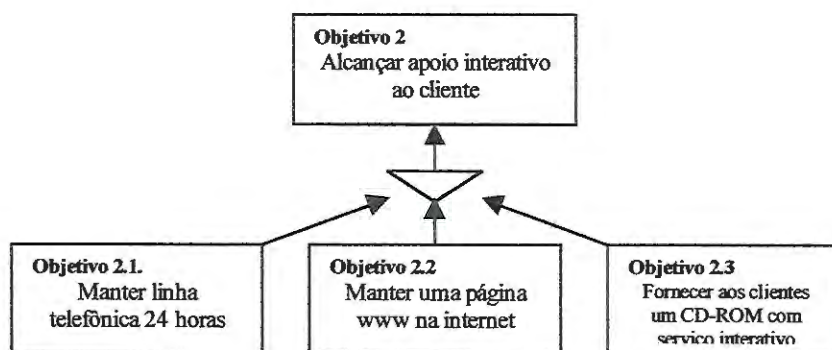


Figura 08 – Fração de um Modelo de Objetivos contendo relacionamento OR.

Fonte: BUBENKO *et al.* (2001), p. 30.

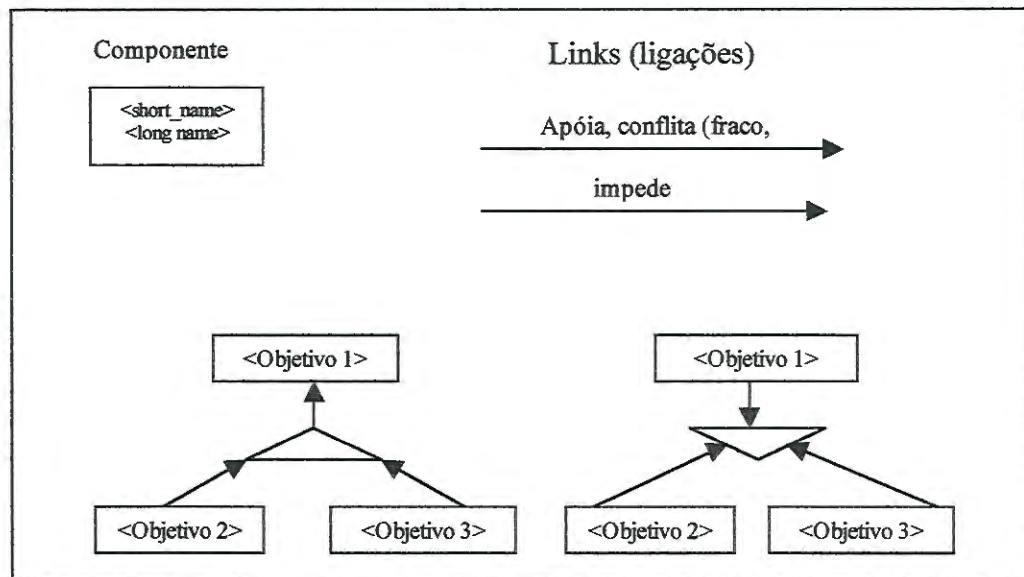


Figura 09 – Notações do Modelo de Objetivos
 Fonte: BUBENKO *et al.* (2001), p. 31.

3.6.1.2. Desenvolvendo o Modelo de Objetivos

BUBENKO *et al.*, (2001) explicam que o MO deve ser desenvolvido inicialmente com a atividade de *brainstorming*. As visões e contribuições dos participantes devem ser consideradas, e normalmente essas visões fazem o produto inicial de modelagem desestruturado e difícil de entender. O passo seguinte inclui estruturação, classificação e operacionalização dos componentes do modelo de objetivos. Isso é normalmente feito pelos projetistas em uma forma interativa, onde os participantes são consultados continuamente para validar o progresso. Essa validação é muito importante para que o Modelo de Objetivos concentre-se nos objetivos do negócio e não no apoio ao sistema de informação e seus objetivos mais técnicos. Para apoiar o sistema de informação existe o Modelo de Requisitos e Componentes Técnicos. As regras estáticas e dinâmicas que governam os estados de mudança da organização são informalmente definidas no MO. Durante a modelagem algumas questões devem ser consideradas:

1. Quais são as estratégias dessa parte da organização?
2. Existem políticas declaradas que podem influenciar esse modelo?
3. Quais convenções, regras, regulamentos e leis são relevantes?

4. O que a organização gostaria de alcançar?
5. Existe algum problema particular impedindo o objetivo?
6. Esse problema está relacionado com algum objetivo particular?
7. Qual é a causa desse problema?
8. Como esse problema poderia ser eliminado?
9. Existem algumas oportunidades particulares que poderiam ser usadas?
10. Quais ações poderiam ser tomadas para melhorar a situação?
11. Como esse objetivo pode ser alcançado?
12. Esse objetivo pode ser definido em termos operacionais, dando um número de sub-objetivos de apoio?

Os objetivos devem ter sentenças completas para evitar ambigüidades ou incertezas. É importante evitar palavras com muitos significados, e palavras que podem ser mal-entendidas em diferentes situações. Deve-se evitar frases: “Ter um bom ABC”, buscando utilizar termos mais tangíveis e comensuráveis como “Ter um bom sistema ABC que tenha baixo custo de produção e manutenção e uma interface multi-usuário” (BUBENKO *et al.*, 2001).

3.6.1.3. Refinamento e Operacionalização dos Objetivos

A proposta de operacionalização de objetivos visa detalhar como o alto nível de objetivos pode ser satisfeito. A operacionalização do Modelo de Objetivos encoraja o desenvolvimento da rede de objetivos. Os aspectos chaves do processo de operacionalização de objetivos são:

- Ênfase na criatividade: a operacionalização de objetivos reflete um pulo criativo dos fatos presentes para futuras possibilidades que trazem alguma coisa nova que previamente não existia.
- Ênfase na natureza dinâmica do processo: o resultado do processo não é estático, mas depende das decisões de projeto e as visões das situações futuras durante o processo de operacionalização. Isso significa que o resultado do processo não é sempre o mesmo.

As principais atividades da operacionalização são refinamento dos objetivos e gerenciamento de conflitos. Durante o refinamento dos objetivos, novos objetivos são gerados através do detalhamento do alto nível. Nesse sentido, um objetivo de alto nível é refinado em

um ou mais sub-objetivos. O resultado desses refinamentos sucessivos é uma estrutura hierárquica multi-nível, iniciando-se de um objetivo de alto-nível vago, até objetivos operacionais específicos. No refinamento de objetivos é possível usar relacionamento AND/OR nas estruturas para refinar objetivos em muitas combinações alternativas de sub-objetivos, ou um sub-objetivo que pode ser realizado por muitas alternativas de modelo de projeto.

O gerenciamento de conflitos consiste de um número de atividades:

- **Deteção de conflitos:** visa a identificação de conflitos entre objetivos. Pode ser difícil relacionar novos objetivos com os objetivos existentes e determinar o efeito daqueles para estes. Para conseguir encontrar conflitos deve-se procurar exaustivamente no modelo de objetivos e comparar o novo objetivo para cada objetivo existente. Uma forma razoável de procurar por potenciais conflitos é usar os conflitos de objetivos de alto nível que já foram identificados durante o estágio de aquisição.

- **Classificação de conflitos:** enfoca a identificação do tipo de conflito que tem sido detectado. Conflito final ocorre quando dois objetivos contraditórios são desejados. Conflito de meios (*means*) ocorre quando os atores relacionados a objetivos idênticos querem usar o mesmo recurso. A classificação de conflitos pode ser usada por métodos de gerenciamento de conflitos para reagir de modo devido.

Para tratar conflitos, alguns dos seguintes tipos de tratamento podem ser usados:

- **Ignorar:** quando o conflito não impede nenhum desenvolvimento, embora seja necessário não perder de vista o conflito em caso do impacto aumentar.

- **Amenizar:** o balanceamento dos objetivos conflitantes pode não ser claro até que várias possibilidades de projeto sejam exploradas em termos de modelos alternativos de projetos. Essa forma de decisão é trocada para o estágio de geração do modelo de projeto, quando dados concretos sobre o caso estão disponíveis.

- **Resolver o conflito:** freqüentemente um objetivo conflitante implica na indisponibilidade de uma alternativa específica para alcançar ambos objetivos. Nesse caso, o ranking de objetivos pode ser útil para decidir uma eventual omissão de um objetivo. Entretanto, alguns conflitos podem ser resultados:

- Da redefinição de objetivos;
- Da especificação do contexto em que cada objetivo é alcançado ou
- Do refinamento de objetivos alternativos.

Segundo BUBENKO *et al* (2001), as duas questões chaves em gerenciamento de conflito são: conhecimento da trajetória do conflito e gravação das informações sobre esses conflitos, tais como as circunstâncias que levam a esses conflitos. Muito frequentemente, os objetivos de alto nível, os problemas, regras do negócio, entre outros, adquiridos no estágio de aquisição, contêm um número de requisitos informais e imprecisos: é recomendado que a saída do Modelo de Objetivo inicial seja estruturada no estágio inicial. Essa tarefa envolve:

- **Classificação de Objetivos:** para melhorar a compreensão e entendimento de inúmeros objetivos, aconselha-se classifica-los em uma tabela matriz, onde eles podem ser categorizados de acordo com origem, *stakeholders*, função, domínio, etc. Isso permitirá a comparação, análise e talvez descobrir a necessidade de mais discussões baseadas na análise dos padrões dos objetivos (figura 10).
- **Priorização de Objetivos:** possibilita a resolução dos conflitos entre objetivos. Um alto nível de objetivos age como uma restrição em um objetivo de baixo nível.
- **Correlação de Objetivos:** é compreendida como interação positiva e negativa entre objetivos. Em geral, colaboração de objetivos é desejável, desde que a colaboração entre dois objetivos signifique que a satisfação de um objetivo apoiará a satisfação de outro objetivo; mostrando também um consenso entre objetivos e *stakeholders*. Por outro lado, a existência de objetivos antagônicos poderia impedir a satisfação dos objetivos. Além disso, falhas para reconhecer objetivos antagônicos poderiam causar confusão através do processo de projeto. Uma abordagem sistemática tem a função de revelar correlações de objetivos usando matrizes para assegurar que todas as interações possíveis sejam descobertas, e que os diagramas tornem claro o padrão dos relacionamentos (figura 11).

Objetivos	Categorias				
	Categoria 1	Categoria 2	Categoria 3	...	Etc
Objetivo 1	X				
Objetivo 2		X	X		
Objetivo 3			X		
Objetivo 4			X		
			X		

Figura 10 – Matriz de classificação por categorias.

Fonte: BUBENKO *et al.* (2001), p 14.

Objetivos	Obj. 1	Obj. 2	Obj. 3	Obj. 4	Obj. 5	Obj. 6
Obj. 1		-	+	+	+	
Obj. 2			-			
Obj. 3				+		
Obj. 4						
Obj. 5						+
Obj. 6						

Figura 11: Matriz de correlação de objetivos.

Fonte: BUBENKO *et al.*, (2001), p.15.

Para ilustrar os principais aspectos do Modelo Organizacional será utilizado o exemplo de uma biblioteca pequena, como descrito em BUBENKO *et al.* (2001). Nas reuniões para modelagem da biblioteca muitos fatores foram identificados:

- A biblioteca terá seu orçamento cortado 50% nos próximos 3 anos.
- A biblioteca precisa encontrar fontes adicionais de recursos financeiros.
- Existe a necessidade de melhorar o serviço.

- Existe uma longa lista de espera para emprestar livros.
- A biblioteca deveria fornecer um alto padrão de serviço.
- A biblioteca necessita de recursos financeiros externos de R\$500.000,00 nos próximos três anos.
- A lista de espera deverá ser minimizada.
- A biblioteca precisa de mais clientes pagantes.
- Existe um regulamento, segundo o qual a biblioteca não deveria cobrar de estudantes e acadêmicos.
- Não existe um bom sistema de registro de todas as transações financeiras.
- A biblioteca está situada na ELECTRUM que é um prédio para companhias de tecnologia da informação.

É freqüentemente possível determinar intuitivamente o tipo de um fato particular. Por exemplo, a sentença “a biblioteca deveria fornecer um serviço de alto nível” parece um objetivo que a gerência da biblioteca deveria alcançar no futuro. A sentença “a biblioteca precisa de mais clientes pagantes” é claramente entendida se é considerado que a biblioteca tem falta de recursos financeiros. Parte do Modelo de Objetivos está representado na figura 12. Este é apenas um modelo inicial baseado nas primeiras discussões. Pode ser elaborado e detalhado de muitas formas para atingir objetivos claros e operacionais e outros componentes do modelo.

Geralmente, existem dois tipos principais de atividades no trabalho do Modelo de Objetivos: aquisição e operacionalização.

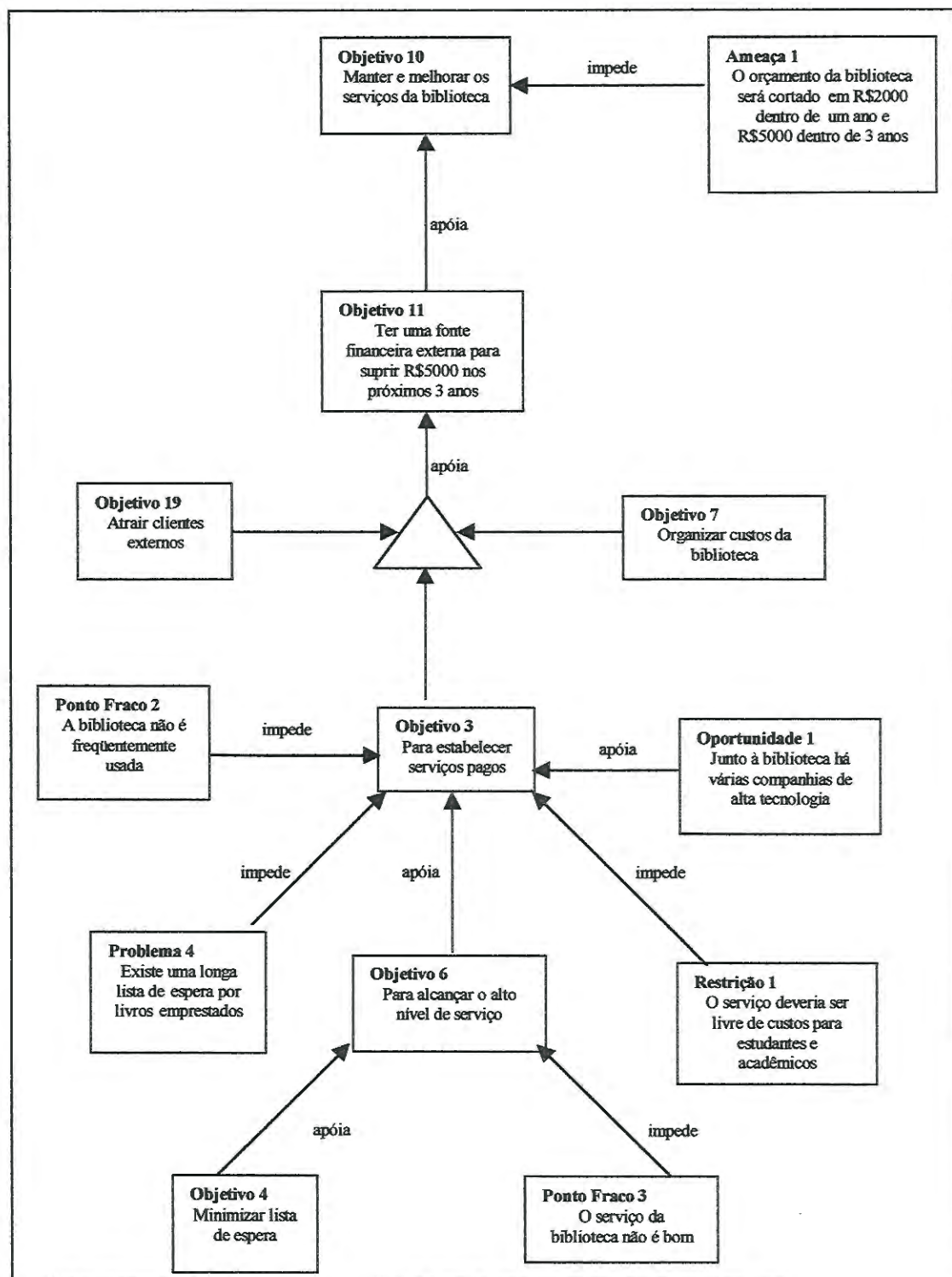


Figura 12 – Parte de um Modelo de Objetivos do caso de uma biblioteca

Fonte: BUBENKO *et al.* (2001), p.32.

3.6.2. Modelo de Regras do Negócio

O Modelo de Regras do Negócio, de acordo com BUBENKO *et al.*, (2001), é usado para definir e manter explicitamente regras do negócio formuladas, consistentes com o Modelo de Objetivos. Regras do Negócio podem ser vistas como operacionalização ou limites dos objetivos. Regras do Negócio são regras que controlam a organização no sentido de definir e restringir quais ações podem ser executadas em várias situações. Elas podem ser:

- Declarações precisas que descrevam o método escolhido pelo negócio para alcançar seus objetivos e para implementar suas políticas.
- Regras impostas externamente sobre o negócio, tais como regulamentos e leis.

Regras do Negócio formam uma hierarquia onde o baixo nível de regras define a forma pelo qual o alto nível de regras ou objetivos são implementados. A modelagem de Regras do Negócio está intimamente relacionada com o Modelo de Objetivos. Regras são definidas por objetivos enquanto também afetam a realização de outros objetivos. Elas disparam processos de negócio e fazem referência a conceitos definidos no Modelo de Conceitos. Atores do Modelo de Atores são responsáveis por alcançar e definir regras de negócio. Regras do negócio requerem funcionalidade do sistema de informação. Componentes do Modelo de Requisitos e Componentes Técnicos podem ser motivados por regras do negócio.

Regras do Negócio podem ser categorizados, segundo BUBENKO *et al.*, (2001), em Regras Derivadas, Regras de Evento-Ação e Regras de Restrição.

Regras Derivadas são expressões que definem componentes derivados da estrutura da informação em termos de entidades que já estão presentes na base de informação do modelo da organização. Elas são introduzidas como um meio de capturar o domínio estrutural de conhecimento que não precisa estar armazenado e seus valores podem ser derivados dinamicamente usando valores existentes ou outras informações derivadas. Uma regra derivada é, por exemplo “Um mau cliente de biblioteca é um cliente que não retorna o livro até a data limite por duas vezes consecutivas”.

Regras de Evento-Ação estão relacionadas com a invocação de atividades. Em particular expressam as condições sobre as quais as atividades devem ser realizadas, como um conjunto de condições disparadoras e/ou um conjunto de pré-condições que devem ser satisfeitos antes da sua execução. Por exemplo: se o retorno de um empréstimo está quatro dias atrasados, envie um lembrete.

Regras de Restrição estão relacionadas à integridade da informação, à estrutura dos componentes, ou às atividades e comportamentos permitidos na organização. Uma restrição é, por exemplo: “o salário de um empregado não deve decrescer”. As regras de restrição podem ser estáticas e de transição. As restrições estáticas aplicam-se a todo estado da base da informação e são independentes de tempo, como por exemplo: “a locação de cada cópia do livro é única e apenas uma”. As restrições de transição definem um estado de transição na base de informação, desta maneira especificando restrições no comportamento do sistema. Uma restrição de transição é, por exemplo: “uma cópia de livro está faltando, se o empréstimo dele está atrasado por mais de quatro semanas”.

No Modelo de Regras do Negócio existem dois tipos de relacionamentos: de apoio e de impedimento.

O relacionamento de apoio é essencialmente visto como vertical, podendo, por exemplo, ser usado para refinar ou decompor regras.

O relacionamento de impedimento é usado para mostrar influências negativas entre componentes do Modelo de Regras do Negócio, e pode ser considerado como o oposto de apoio. Assim como no Modelo de Objetivos, também existem as estruturas de decomposição AND/OR no Modelo de Regras do Negócio.

O relacionamento de refinamento AND representa um conjunto de sub-regras, que são necessárias para satisfazer a regra original refinada. O relacionamento de refinamento OR representa um conjunto de sub-regras alternativas no qual, para apoiar uma regra original, é necessário satisfazer apenas uma regra do conjunto (BUBENKO *et al.*, 2001).

BUBENKO *et al.*, (2001), afirmam que todas as regras do negócio deveriam ser expressas de acordo com a figura 13. Entretanto, nem sempre é possível escrever regras do negócio usando esse padrão proposto. Geralmente existem três formas sob as quais as regras do negócio podem ser declaradas: informalmente, em linguagem natural; formalmente, utilizando o português estruturado; ou formalmente, usando linguagens especialmente para desenvolver regras, tal como TEMPORA ERL, que é uma notação técnica para expressar regras do negócio. As duas últimas formas de expressar regras de forma não-ambígua contribuem para facilitar a implementação delas no projeto de sistema de informação.

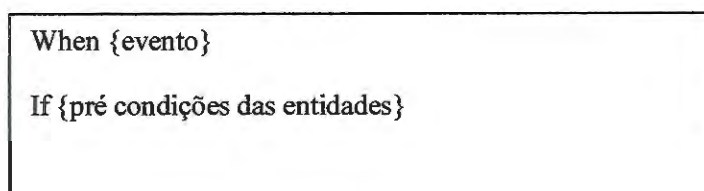


Figura 13: Estrutura de Regras do Negócio

Fonte: Pádua (2000), p. 92.

Cada expressão de regras do negócio deveria ter apenas uma regra atômica, que é uma declaração específica e formal de uma simples restrição, fato, derivação, ou termo que não pode ser decomposto posteriormente. Por exemplo, a regra: “Clientes tornam-se maus-clientes se demorarem mais do que quatro semanas para entregarem o livro” será declarada da seguinte forma:

Quando verificar_atraso_livros

Se hoje – empréstimo.retorno_dia > 29

Então Declare_cliente_como_um_mau (Empréstimo.Cliente_ID)

Através de uma regra escrita dessa forma (figura 14), os participantes e projetistas são capazes de obter informações mais precisas sobre:

- Quais entidades (Empréstimo, Cliente, Mau Cliente) estão envolvidas ou relacionadas na realização dessa regra;
- Quais processos apóiam e são disparados (*check_delay_books*) por essa regra.

De acordo com BUBENKO *et al.* (2001), é possível formalmente expressar apenas regras atômicas. Regras que são mais complexas precisam ser expressas em linguagem natural ou seminatural e, então decompostas e refinadas usando relacionamentos AND/OR. Existem regras atômicas que são quase impossíveis de serem definidas de forma formal. Por exemplo, regras que são relacionadas a requisitos não-formais.

Ao modelar regras do negócio, é importante considerar as seguintes questões:

1. Existem regras declaradas e política com a companhia que podem influenciar esse modelo?
2. Por quais regras os objetivos da organização podem ser alcançados?
3. A regra está relacionada a um objetivo particular?

4. Como essa regra pode ser decomposta?
5. Como a organização pode ajustar-se à especificação dessa regra?
6. Como validar se a regra é cumprida?
7. Quais processos disparam essa regra?
8. Essa regra pode ser definida em uma forma operacional?
9. Pode uma regra ser decomposta em regras simples?

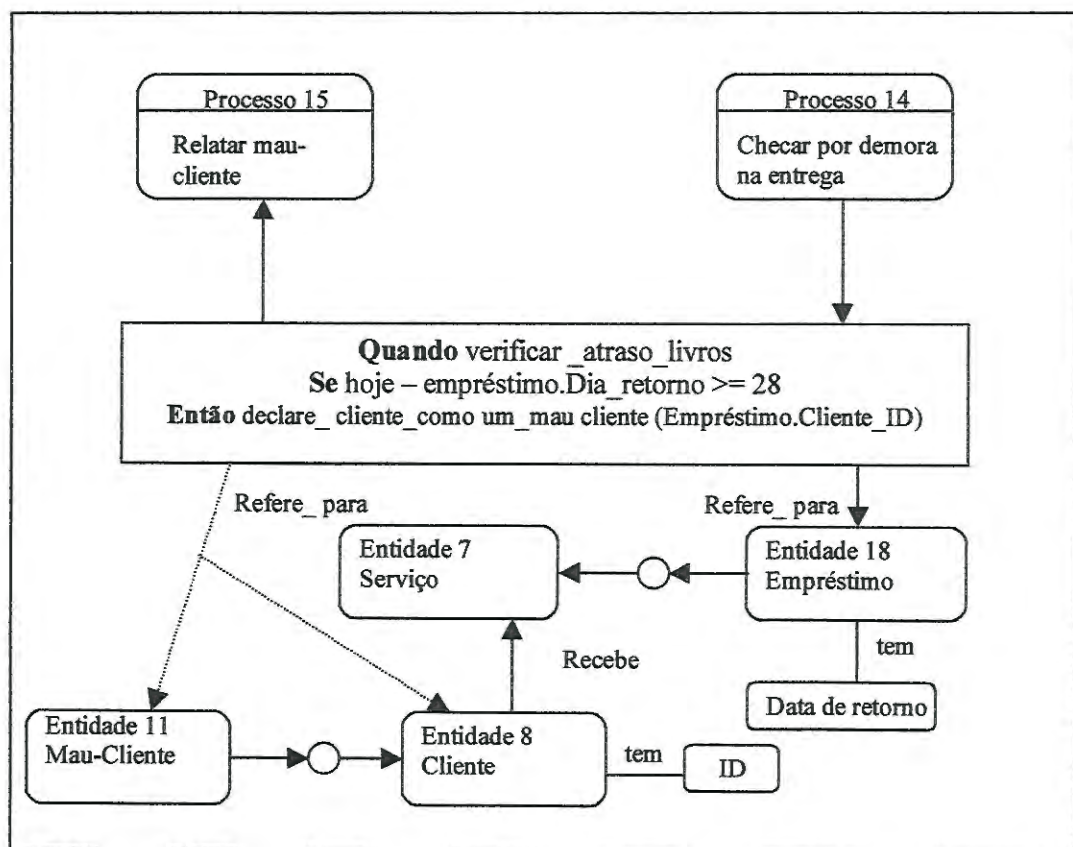


Figura 14 – Regras referem-se a entidades do modelo de conceitos e são apoiadas por processos no Modelo de Processos do Negócio.

Fonte: BUBENKO *et al.* (2001), p.41.

Segundo BUBENKO *et al.* (2001), considerando novamente o caso da biblioteca, é possível formular muitas regras e representá-las graficamente (figura 15):

- Clientes devem ser considerados como maus clientes se eles atrasarem a entrega do livro em ocasiões consecutivas ou em quatro semanas.

- Clientes deveriam pagar multas se o livro está atrasado, independente de qualquer privilégio que eles tenham.
- Maus-clientes que não são clientes pagantes não deveriam ser habilitados a pedir livros de outras bibliotecas.
- Clientes pagantes não deveriam ter prioridade na lista de espera.
- O catálogo da biblioteca deveria ser atualizado logo que as mudanças ocorressem.

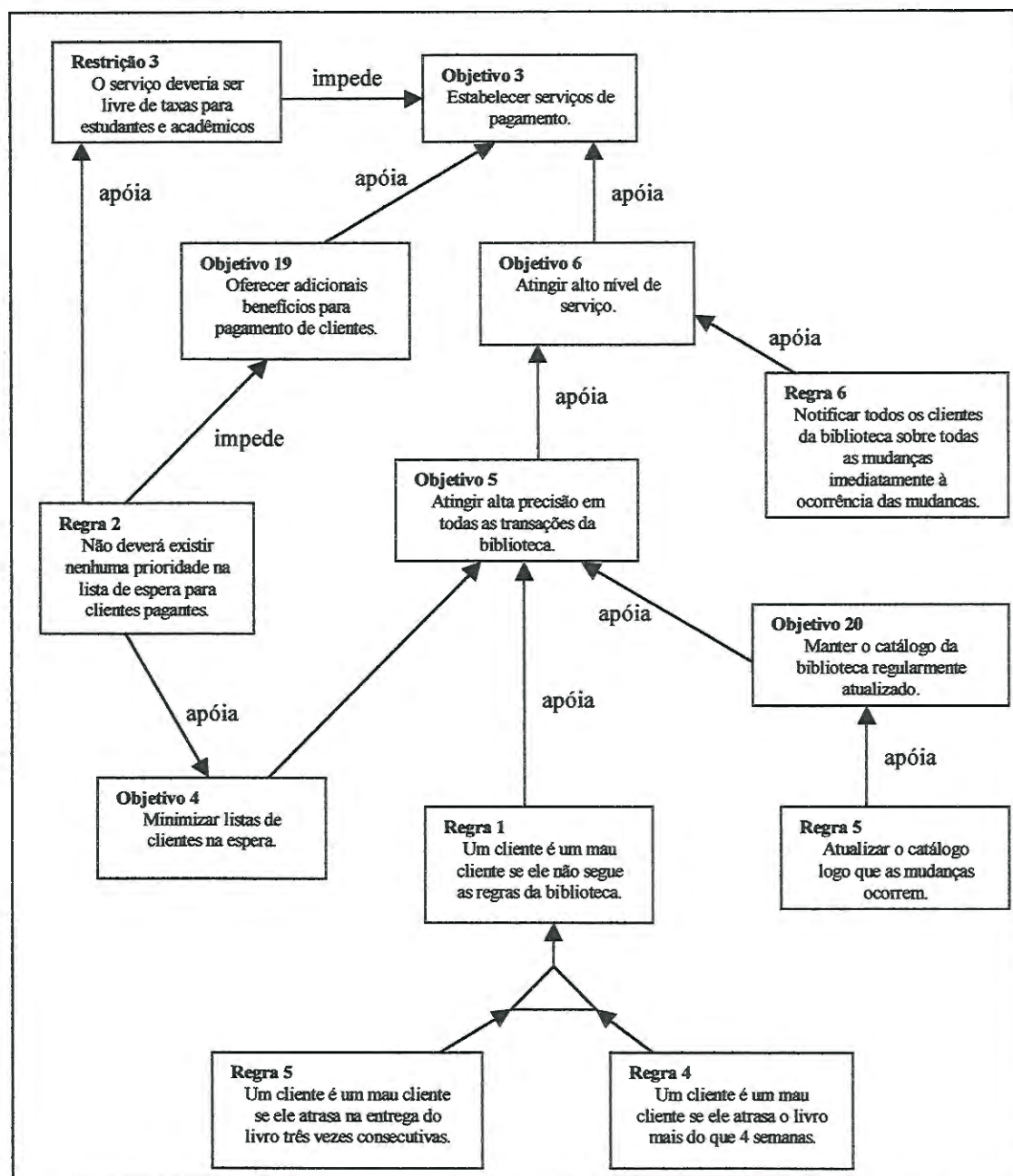


Figura 15: Interação entre o Modelo de Regras do Negócio e componentes do Modelo de Objetivos.

Fonte: BUBENKO *et al.* (2001), p.40.

3.6.3. Modelo de Conceitos

De acordo com BUBENKO *et al.* (2001), o Modelo de Conceitos é usado estritamente para definir “coisas” e “fenômenos” que estão em outros modelos, além de definir entidades e dados da aplicação em um nível conceitual, servindo como um dicionário. O Modelo Conceitual deve, ao menos, incluir componentes pelos quais se pode descrever o conteúdo de diferentes conjuntos de informação e fluxos do Modelo de Processos do Negócio. Por exemplo, a expressão do objetivo: “manter e melhorar o serviço da biblioteca”, requer a definição do conceito “serviço da biblioteca” no Modelo de Conceitos. É importante que todas as entidades usadas nos outros modelos sejam definidas aqui para evitar a possibilidade de mal-entendidos entre os participantes e *stakeholders*. Por essa razão, inconsistências são evitadas.

Os Modelos de Conceitos incluem componentes tais como: entidades, relacionamentos binários e atributos de informação. Os relacionamentos ISA e Part OF são incluídos no Modelo de Conceitos para permitir a modelagem de generalização e de componentes complexos. O Modelo de Conceitos permite definir diferentes “grupos de componentes de Modelo de Conceitos”. Esse grupo é simplesmente uma visão de uma parte de um modelo de conceitos, e inclui um sub-conjunto de entidades do Modelo de Conceitos, relacionamentos e atributos. Um grupo pode ser um membro de outro grupo.

Entidade é alguma coisa do domínio de interesse e aplicação que deve ser entendida, caracterizada e definida usando relacionamento para outras entidades. Atributo é uma entidade usada apenas para caracterizar outra entidade. As entidades podem ser relacionadas por meio de relacionamento semântico, tais como: relacionamento binário, relacionamento generalização/especialização (ISA), e relacionamento de agregação (Part OF).

Para KIRIKOVA (2000), todos os elementos da estrutura ISA não refletem apenas por elementos da área e recursos do negócio, mas também por elementos relacionados diretamente e indiretamente. A intenção aqui não é mostrar estruturas de conhecimento correspondente, mas convencer de que a estrutura fornece uma possibilidade de representar o conhecimento requerido.

O relacionamento binário é uma semântica entre duas entidades ou com uma entidade. Relacionamentos binários são inerentemente bidirecionais. A cada direção pode ser dado um nome, preferivelmente em forma de uma frase verbal. A direção indicada pela seta pode ser chamada de *forward* (para frente) ou primária, e a direção oposta como direção inversa. Um exemplo de relacionamento binário é dado na figura 16. As entidades participantes no

relacionamento podem executar certas tarefas no relacionamento. No relacionamento “Editor imprime Livro/Livro é impresso pelo Editor”, o editor é quem imprime, enquanto o livro é o que é impresso pelo Editor (BUBENKO *et al.*, 2001).

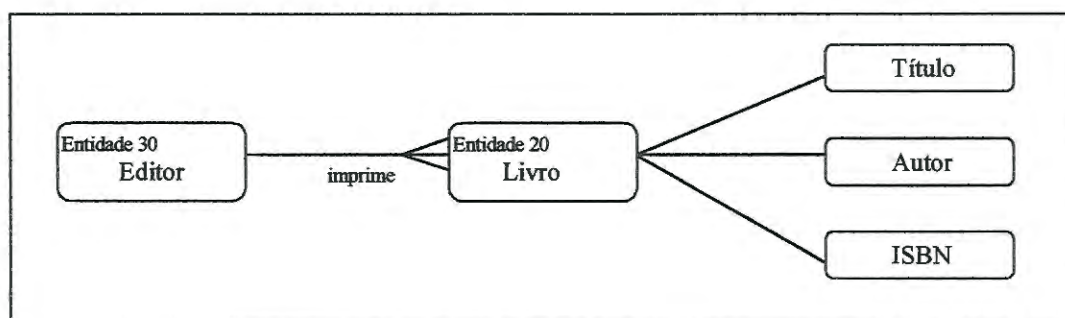


Figura 16 – Exemplo de Relacionamento Binário.

Fonte: BUBENKO *et al.* (2001) pg. 44.

O relacionamento ISA é um tipo específico de relacionamento semântico entre entidades. Se “A” ISA “B”, então “B” é o conceito mais genérico e “A” é o conceito específico. Esse tipo de relacionamento é também referenciado como generalização. O oposto ou inverso de generalização é chamado de especialização. (figura 17). A propriedade mais significante de um relacionamento ISA é a herança. Tudo que é especificado para ser verdadeiro sobre um conceito genérico é também verdade para um conceito específico, significando que todos atributos, seus valores e restrições são herdados de um nível mais genérico de conceito para um nível mais específico. Os sub-tipos que resultam de uma especialização particular de um tipo de objeto podem não coincidir. Considerando a especialização de Cliente como Individual e Organizacional, um simples cliente não pode ser individual e organizacional ao mesmo tempo. Tal tipo de especificação é chamado total. Existe uma diferença no modelo entre cliente “organizacional” e cliente “individual”. Se se espera relatar alguma coisa específica sobre um mau cliente que não está relacionado a um tipo geral de cliente, é necessário introduzir um conceito separado para ele. A especificação é parcial porque não estão sendo especificados outros subtipos de clientes, que poderiam ser, por exemplo, freqüente ou normal. A especialização é total quando todas as instâncias de um tipo genérico são membros de um tipo específico. Isso ocorre quando a especialização é uma partição de um tipo genérico. Quando a

especialização é parcial existem instâncias de um conceito genérico que não é membro de nenhum sub-tipo.

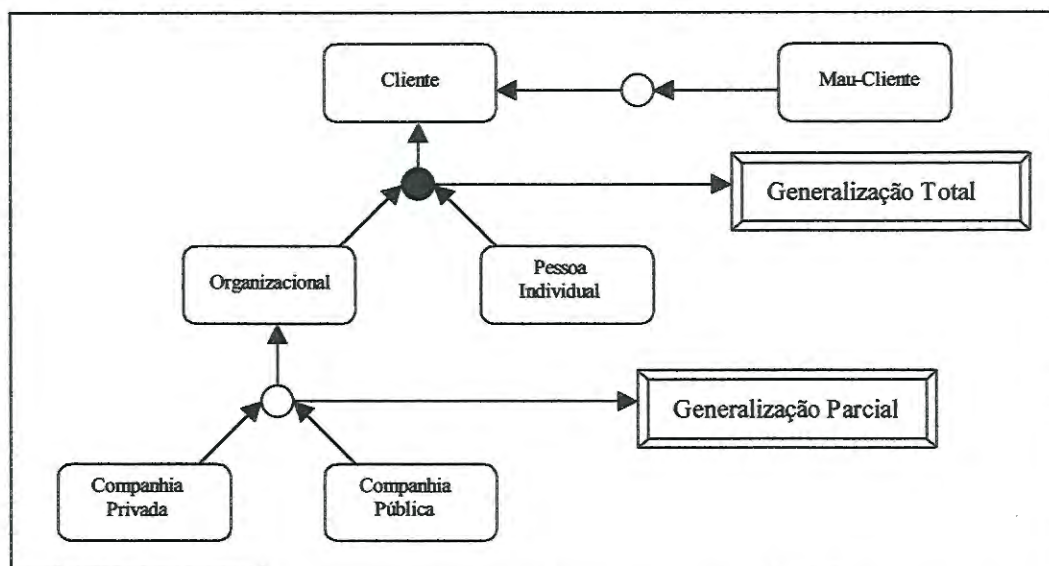


Figura 17 – O conceito de “Cliente” pode ser especializado de formas diferentes.

Fonte: BUBENKO *et al.* (2001) pg. 45.

O relacionamento Part OF ou uma agregação (figura 18) é uma forma especial de relacionamento semântico, onde as entidades inter-relacionadas são “fortemente e firmemente acopladas” uma as outras. O objeto agregado é uma reunião de partes, e as partes são componentes de agregação. Os objetos componentes são freqüentemente subordinados ao objeto agregado. O exemplo mais típico de agregação é uma exposição de uma parte, onde a parte em um nível mais superior tem um número de componentes, e onde alguns desses componentes no próximo nível superior tem um número de componentes, e onde alguns desses componentes no próximo nível são vistos como agregados, que por sua vez é composto de partes. A construção do relacionamento Part OF é incluída em um Modelo de Conceitos por razões de conveniências, de forma que será possível usá-la sempre que for natural e compensador, ver e operar o relacionamento sobre algo como parte de uma hierarquia ou estrutura de componentes. A figura 24 mostra que um documento é uma estrutura de componentes: um cabeçalho e um corpo de documento. O corpo do documento tem um ou mais capítulos e apêndices como componentes.

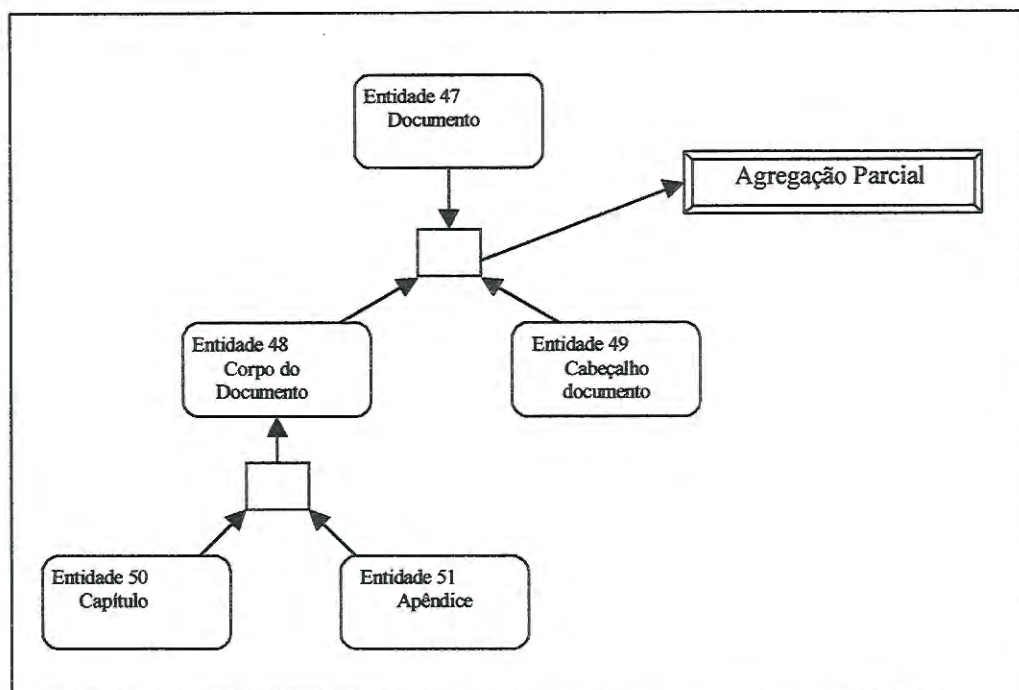


Figura 18 – Exemplo de Relacionamento Part OF.

Fonte: BUBENKO *et al.*, (2001) pg. 46.

No Modelo de Conceitos incluem componentes que representam a informação necessária ou produzida pelos processos do Modelo de Processos do Negócio. Portanto, se existe a necessidade de algum processo do Modelo de Processo do Negócio tal como, “Procurar a disponibilidade de um livro no Catálogo da Biblioteca”, então o livro e suas cópias devem ser incluídos no Modelo de Conceitos e seus atributos relevantes devem ser declarados. A inclusão da informação no Modelo de Conceitos e seus atributos relevantes devem ser declarados. A inclusão da informação no Modelo de Conceitos não implica na realização de um sistema de informação computadorizado.

O catálogo da biblioteca pode ser visto como um processo de alto nível do Modelo de Processos de Negócio, mesmo se for usado outro nome, como “Catalogamento de Livros”. Esse processo do Modelo de Processos do Negócio conterá todos os sub-processos que podem ser vistos como parte do “processo de alto nível” Catalogação de Livros.

Um conceito pode ter diferentes interpretações e diferentes usos; e pode aparecer em diferentes submodelos. Isso é importante para distinguir claramente os diferentes usos e colocar

componentes em um modelo apropriado no início, antes que o modelo cresça muito e fique confuso.

Segundo BUBENKO *et al.*, (2001), no Modelo de Conceitos pode-se usar “entidades reais”. A experiência tem mostrado que o modelo de operações pode ser muito ilustrativo e apoiar o entendimento humano. Palavras com significados específicos podem algumas vezes ser necessárias para documentação, e ter seus relacionamentos discutidos. Pode acontecer dessas entidades serem introduzidas em textos de declaração de objetivos no Modelo de Objetivos, e precisarem de mais definição e discussão. Ao modelar entidades muitas questões devem ser consideradas para facilitar o trabalho, tais como:

1. Quais são as principais entidades da aplicação?
2. Como essas entidades são relacionadas?
3. Por que essa entidade é necessária?
4. O que é necessário saber sobre o conceito da aplicação? Quando e onde surgirá a necessidade dessa entidade?
5. Quantas instâncias dessa entidade existem?
6. Quando essa instância deixa de existir?
7. Quais situações acima refletem no Modelo de Processos do Negócio?
8. Os atributos são simples ou multivalorados?
9. O tipo de entidade é geralmente relacionado a algum outro tipo?
10. Existe a necessidade de algum dado histórico sobre entidades, relacionamentos ou atributos?
11. O relacionamento é estável ou varia de acordo com o tempo?
12. Como as entidades variam de acordo com o tempo?
13. Existem características temporais dos atributos?

3.6.4. Modelo de Processo do Negócio

De acordo com BUBENKO *et al.*, (2001), o Modelo de Processo do Negócio destina-se a analisar o processo e fluxo de informação e material da organização. Processos podem ser decompostos em sub-processos. Componentes do Modelo de Processos de Negócio são, primariamente motivados pelos componentes do Modelo de Objetivos e também possibilitam que objetivos do Modelo de Objetivos sejam alcançados. O Modelo de Processos de Negócio

descreve as atividades organizacionais (funções e processos da organização). O núcleo da organização é o conjunto de processos, contribuindo para os valores da organização. Para alcançar uma boa abstração, o Modelo de Processos de Negócio permite total liberdade no processo de decomposição em sub-processos (figura 19 e 20). Dependendo do propósito da modelagem, os processos descritos podem ser existentes ou futuros (processos planejados). Um exemplo de um processo de negócio é o “Gerenciamento de retornos de empréstimos”, que pode ser decomposto em “registro de retorno”, “checar-se o livro está reservado” e “retornar à prateleira”.

Os componentes do Modelo de Processos de Negócio são processos, processos externos e conjunto de materiais e informações. Processo é uma coleção de atividades que:

- Consome entradas e produz saídas em termos de informação ou material;
- É controlada por um conjunto de regras, indicando como processar as entradas e produzir saídas;
- Tem um relacionamento com o Modelo de Atores e Recursos, em termos do executor ou responsável pelo processo, e
- Como uma instância do Modelo de Processos de Negócio é esperado para consumir, quando iniciado, um montante finito de recursos e tempo.

Processos externos representam uma coleção de atividades que:

- São localizados fora do escopo da área da atividade organizacional;
- Comunica-se com processos ou atividades da área do domínio do problema
- São essenciais para documentar.

Processos externos algumas vezes podem ser considerados como fluxo de entrada ou saída de alguma informação e material. Um exemplo típico de processo externo pode ser o de um cliente que pede ou recebe algum serviço da biblioteca. Conjunto de material ou informação é um conjunto de informação ou material enviado de um Processo ou Processo Externo para outro. Os conteúdos do fluxo de material e informação são descritos no Modelo de Conceitos, onde eles são decompostos se necessário. O fluxo de material e informação deve ter ao menos um envio de processos externos e ao menos um recebimento de processo ou processo externo. Os processos de alto nível deveriam ser separados dos processos de baixo nível com o mecanismo de decomposição. Igualmente, se não existir nenhum nível máximo de intensidade

de decomposição, a restrição é desejada, pois evita estruturas complexas desnecessárias (BUBENKO *et al.*, 2001).

Um exemplo de decomposição é realizado de um simples processo de verificação que tem como entrada “Endereço” e duas saídas “Endereço Inválido” e “Endereço Válido” (figura 19). Na figura 20 o processo é decomposto em quatro sub-processos. Cada um desses sub-processos tem um fluxo de dados de entrada. No exemplo considera-se que “Endereço” é um fluxo de informação complexa sendo então dividido em vários componentes. No caso da biblioteca existem vários processos como: emprestar e devolver livros, recebimento entre outros. Todos eles podem ser decompostos. A figura 21 mostra um modelo para empréstimo de livros. Nesse exemplo, o processo de empréstimo de livros é disparado. São mostradas partes do Modelo de Atores e Recursos que realizam o processo Negociação com o Cliente. Ligações entre fluxo de informação e entidade indicam o relacionamento entre Modelo de Processos de Negócio e Modelo de Conceitos. Não existe a necessidade de mostrar todos os possíveis relacionamentos entre Modelo de Processos de Negócio e outros submodelos do Modelo Organizacional, apenas as ligações mais importantes para reduzir a complexidade do modelo.

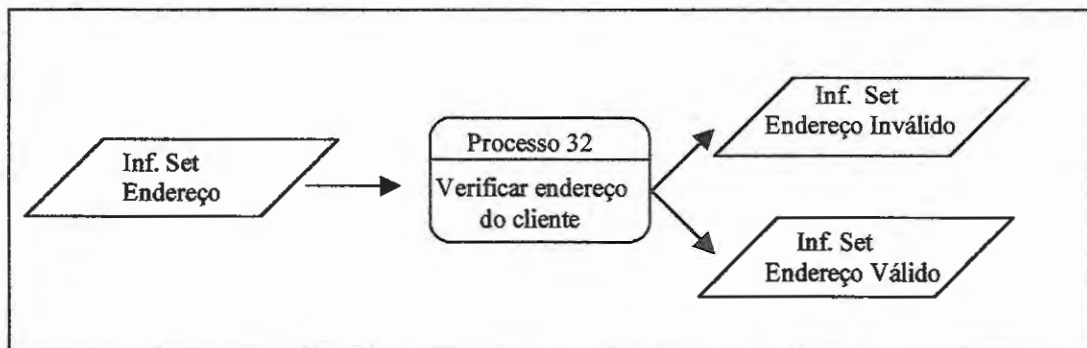


Figura 19: O processo verificação do endereço do cliente não é decomposto.

Fonte: BUBENKO *et al.* (2001), p 54.

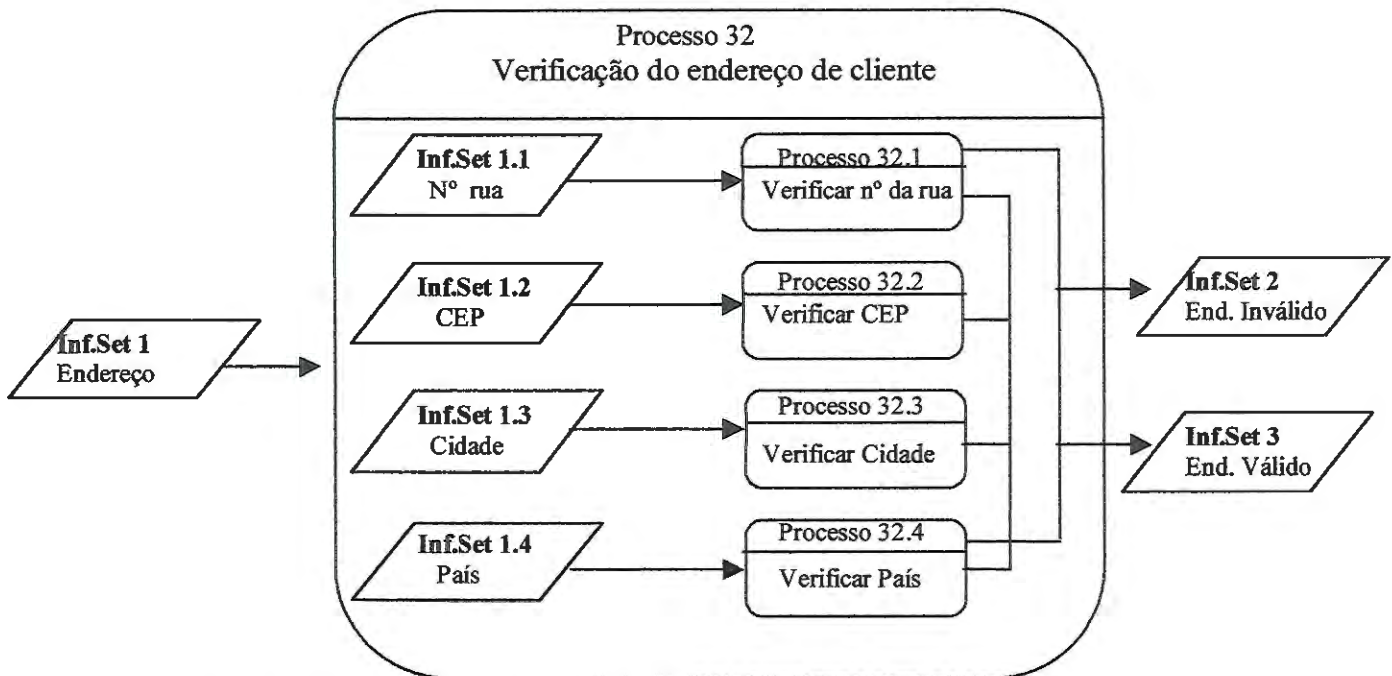


Figura 20 – O processo verificação de endereço é decomposto em quatro sub-processos.

Fonte: BUBENKO *et al.* (2001), p.54.

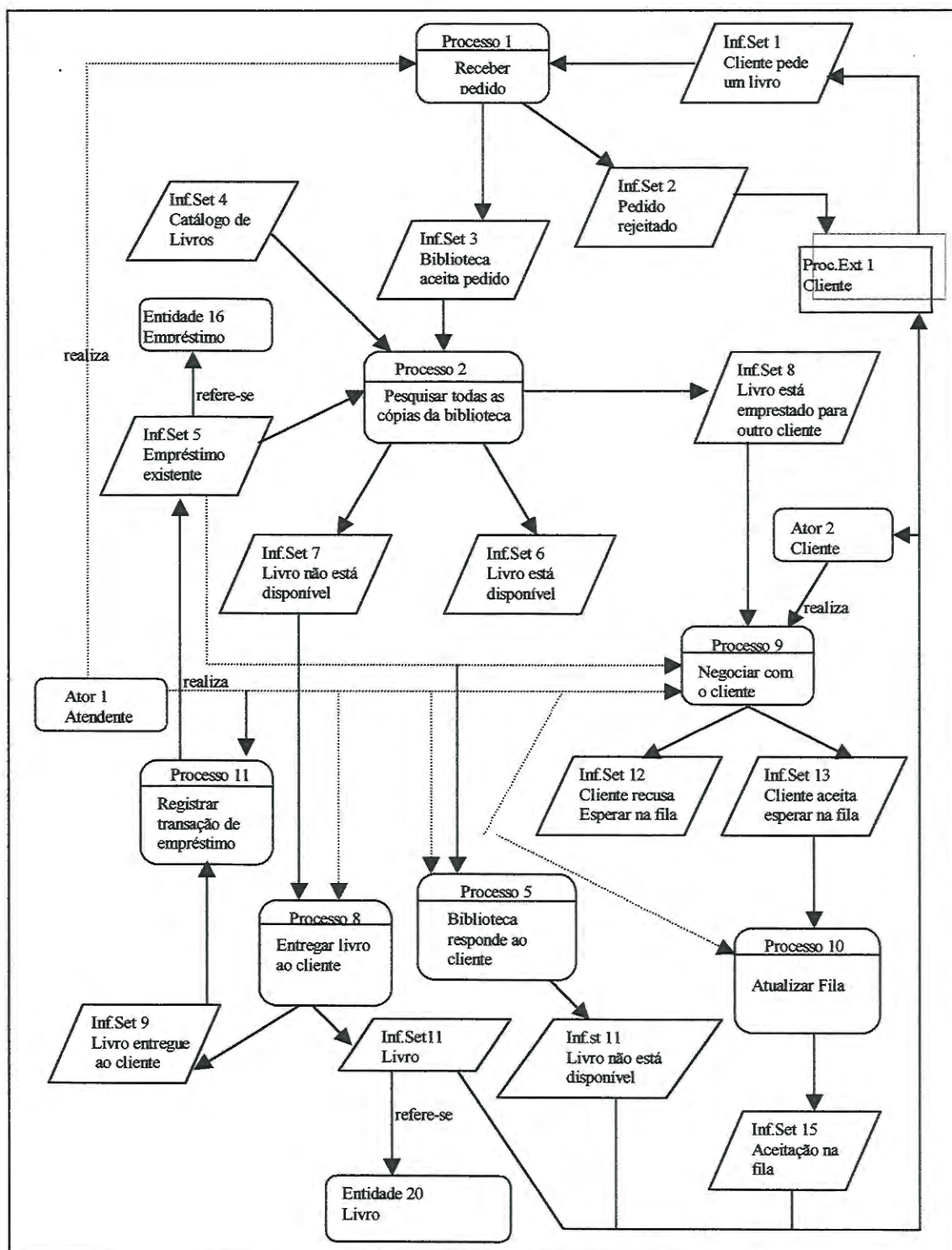


Figura 21 – Um exemplo de Modelo de Processos de Negócio.

Fonte: BUBENKO *et al.* (2001), p.53.

Os componentes do Modelo de Processos do Negócio devem ser motivados pelos objetivos da organização definidos no Modelo de Objetivos. Processos do Modelo de Processos de Negócio são realizados com a informação descrita pelos componentes do Modelo de Conceitos. Tais como, entidades, atributos e relacionamentos ou grupos de componentes do Modelo de Conceitos. Os componentes do Modelo de Processos do Negócio também estão intimamente relacionados com todos componentes do Modelo de Atores e Recursos. O relacionamento entre o Modelo de Processos de Negócio e Modelo de Atores e Recursos pode ser de diferentes tipos, tais como:

- Ator A realiza processo P;
- Ator A é responsável por processo P;
- Ator A apóia processo P ou
- Ator A é um consultante para processo P.

De acordo com BUBENKO *et al.* (2001), cada componente do Modelo de Processos do Negócio deve, ao menos no nível de decomposição, ter um relacionamento definido no Modelo de Atores e Recursos. É importante observar que o Modelo de Processos do Negócio descreve processos da área do negócio, e não unidades organizacionais e sistema. Para um componente ser qualificado para inclusão no Modelo de Processos do Negócio, deve-se descrever um conjunto de possíveis processos com um tempo de início e fim. Em um alto nível de abstração, esse conjunto de processos pode ser razoavelmente bem definido. As principais distinções entre um processo (tipo) e um componente do Modelo de Atores e Recursos são:

- Temporal:
 - Um componente de Modelo de Atores e Recursos é criado. Ele existe até se desfazer ou ser excluído pelo ambiente.
 - O Modelo de Processos do Negócio descreve tipos de processos, cujas instâncias existem por um tempo limitado.
- Instanciação
 - O Modelo de Atores e Recursos contém componentes em um nível de instância, como unidades organizacionais, recursos individuais ou atores humanos e tarefas.
 - O Modelo de Processos do Negócio descreve processos em um nível de classe.

Seguindo esta abordagem, algumas questões devem ser consideradas na modelagem de processos de negócio:

1. Quais são os principais processos da organização?
2. Como esses processos são relacionados?
3. Por que esse processo é necessário?
4. Quais informações e fluxos de material são necessários?
5. Quais fluxos de informação e material são necessários?
6. O que o fluxo de informação e material produzem?
7. Eles são representados no Modelo de Conceitos?
8. As situações que “criam” e “destroem” esses conjuntos de informações ou material são refletidas no Modelo de Processos do Negócio?
9. Quais regras disparam esse processo?
10. Quais atores são responsáveis por realizar e apoiar esse processo?

3.6.5. Modelo de Atores e Recursos

O Modelo de Atores e Recursos (MAR) define que tipo de atores e recursos, ou quais atores individuais estão envolvidos nas atividades organizacionais. O Modelo de Atores e Recursos descreve:

- Como diferentes atores e recursos são relacionados entre si e como são relacionados com componentes do Modelo de Objetivos;
- E de que forma os objetivos estão relacionados aos processos do Modelo de Processos do Negócio.

O Modelo de Atores e Recursos descreve o sistema do negócio existente ou futuro, contendo recursos humanos e não-humanos. Isso permite a inclusão, como parte da engenharia de requisitos, a descrição de um sistema sócio-técnico para ser desenvolvido. Essa descrição não pode ser feita pelo Modelo de Processos de Negócio e Modelo de Conceitos. Um ator do Modelo de Atores e Recursos pode ser dependente de um número de outros atores para a realização de certas tarefas ou processos.

Os componentes do Modelo de Atores e Recursos definem atores e recursos envolvidos nas atividades empresariais, articulados no Modelo de Processo de Negócio, ou atores

relacionados com outros modelos ou para o desenvolvimento do sistema. Ator e recurso podem ser:

- Individual: denota uma pessoa da organização. Por exemplo: Fabiana ou Pedro. Indivíduos são identificados pelo seu nome. Entretanto como os nomes podem não ser únicos, devem ser usados com moderação. Pessoas essenciais com habilidades ou tarefas específicas são incluídas no Modelo de Atores e Recursos, uma vez que elas esclarecem e adicionam significado ao modelo e seus relacionamentos. Indivíduos podem executar tarefas e pertencer a unidades organizacionais. Por exemplo, o indivíduo Fabiana executa a tarefa “gerenciar biblioteca” e pertence a uma Unidade Organizacional “Biblioteca Central”. Indivíduos podem, entretanto, ser relacionados com outros indivíduos, tarefas, unidades organizacionais e recursos não-humanos, por relacionamentos binários semânticos. Os relacionamentos ISA e Part OF não são relevantes para indivíduos.

- Unidade Organizacional: pode representar toda estrutura na organização tais como grupo, departamento, divisão, projeto, time, subsidiário, entre outros. Sendo atores, unidades organizacionais podem ter sub-unidades. Eles podem também executar tarefas e ter outros atores. Unidades organizacionais podem estar relacionadas a outras unidades organizacionais, a indivíduos, tarefas, e recursos não-humanos por relacionamento binário semântico.

- Recursos não-humanos: podem ser tipos de máquinas, sistemas e equipamentos. Sendo atores, recursos não-humanos podem ter componentes, ser generalizados ou especializados e executar tarefas. Naturalmente, a mesma tarefa em situações diferentes pode ser executada por diferentes recursos não-humanos. Recursos não-humanos podem ser recursos para processos. Eles podem também ser relacionados a outros recursos não-humanos, a indivíduos, unidades organizacionais e tarefas em relacionamentos semânticos e binários.

- Tarefas podem ser executadas por unidades organizacionais em contextos diferentes. Uma unidade organizacional pode, por exemplo, executar tarefas de administrador e autorização no mesmo contexto. Isso pode ser importante para identificar requisitos dependendo da tarefa que eles têm. Por exemplo: Ator, Supervisor, Gerente, entre outros. As tarefas podem pertencer a uma ou mais unidades organizacionais, e ser relacionadas a outras tarefas, indivíduos, unidades organizacionais, e recursos não-humanos por relacionamento binário (com nome do usuário). Tarefas podem ser generalizadas ou especializadas ou tarefas componentes, podem realizar processos e serem responsáveis pela realização dos processos e objetivos alcançados. Elas podem também definir objetivos.

Relacionamento binário é usado para descrever tipos diferentes de relacionamentos entre seus componentes. Os dois principais propósitos para relacionamento binário entre os componentes do Modelo de Atores e Recursos e componentes de outros submodelos são definidos por:

- **Responsabilidade:** é um relacionamento entre atores e processos de negócio, regras do negócio e objetivos. Responsabilidade pode ser delegada ou transferida entre atores. Responsabilidade pode ser Organizacional ou Operacional.

A Responsabilidade Organizacional é relacionada à liberdade de um ator tomar decisões por outras entidades da organização, como objetivos, regras, recursos, processos de negócios e outros atores. Responsabilidades Organizacionais podem ser representadas com os relacionamentos abaixo.

- Ator_define_objetivo
- Ator_é_responsável_por_objetivo
- Ator_é_responsável_por_regra
- Ator_define_regra
- Ator_é_responsável_por_recurso
- Ator_é_responsável_por_processo_de_negócio
- Ator_proprietário_recurso
- Ator_monitora_outro_ator

A Responsabilidade Operacional é relacionada principalmente com a execução de tarefas e indica que um ator é alocado para realizar um processo de negócio ou que o processo de negócio é fixado por um ator. A Responsabilidade Operacional pode ser representada como “ator realiza processo”, significando que o realizador da tarefa tem uma responsabilidade operacional.

- **Dependência** é uma relação entre atores da organização. Um ator depende de outro para alguma coisa que pode ser um recurso ou um processo de negócio. Dois tipos de dependências podem ser identificados: Operacional e Autoridade. A dependência pode ser simultaneamente Operacional e de Autoridade.

A Dependência Operacional diz respeito a dependências criadas no fluxo de trabalho. Por exemplo, atores dependem de outros para usarem recursos que são necessários ao processo

de negócio que eles realizam, ou um ator pode esperar por uma saída de outro processo de atores.

A Dependência de Autoridade está relacionada com a dependência criada por causa das regras organizacionais, regulamentos, ou relacionamentos de autoridades e poder. Por exemplo: um usuário precisa de uma senha para trabalhar em um sistema computacional, um vendedor precisa de permissão para usar linhas internacionais.

A notação do Modelo de Atores e Recursos é apresentada na figura 22. Dois tipos específicos de relacionamentos fazem parte do Modelo de Atores e Recursos: ISA e PartOF.

ISA é usado para descrever relacionamentos de generalização entre tarefas do Modelo de Atores e Recursos. A expressão “A” ISA “B” declara que componentes que executam a tarefa B também executa a tarefa A, como “Mau-Cliente ISA Cliente”. Propriedades e relacionamentos são herdados por B. Isso significa que se A está operando processo P, então B também está operando processo P.

PartOF é usado como “B” PartOF “A”, e declara que B é um componente de A.

Esse tipo de relacionamento pode expressar relacionamentos de componentes de sistemas técnicos ou ser útil na modelagem das hierarquias organizacionais, por exemplo, a unidade organizacional X é PartOF de Y.

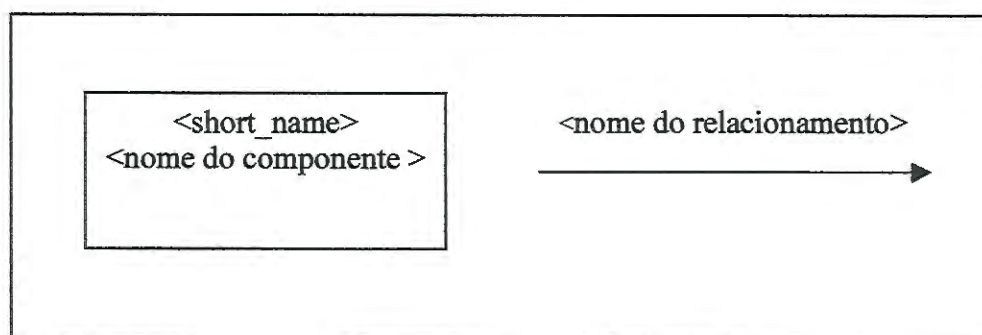


Figura 22 – Notação dos componentes do Modelo de Atores e Recursos.

Fonte: BUBENKO *et al.* (2001), p 59.

Ligações entre os componentes do Modelo de Processos do Negócio e o Modelo de Atores e Recursos descrevem o tipo de relacionamento que existe entre um particular processo

de negócio, um ator ou recurso. Para facilitar o processo de modelagem e melhorar a qualidade do modelo as seguintes questões devem ser consideradas:

1. Quais são os principais atores dessa aplicação?
2. Como os atores estão relacionados?
3. Por que o ator é necessário?
4. Qual é o seu propósito?
5. Esse ator é representado no Modelo de Conceitos?
6. Por quais processos o ator é responsável?
7. Quais processos esse ator realiza?
8. Quais objetivos são definidos por esse ator?
9. Quais regras do negócio são definidas por esse ator?
10. Por quais regras do negócio esse ator é responsável?
11. De quais recursos esse ator é proprietário?
12. Por quais recursos esse ator é responsável?

Referenciando o caso da biblioteca, foi encontrado, entre outras, as seguintes tarefas (papéis) e relacionamentos binários:

- a) a tarefa de prover serviços para clientes;
- b) a empresa KTH financia a biblioteca;
- c) estudante faz o papel de cliente não pagante;
- d) o atendente trabalha na biblioteca.

A figura 23 mostra apenas o Modelo de Atores e Recursos. Entretanto os componentes do Modelo de Atores e Recursos também podem ser relacionados a outros modelos. Por exemplo: No Modelo de Processos de Negócio (figura 21) existem dois executores Atendente da Biblioteca e Cliente que realizam alguns processos.

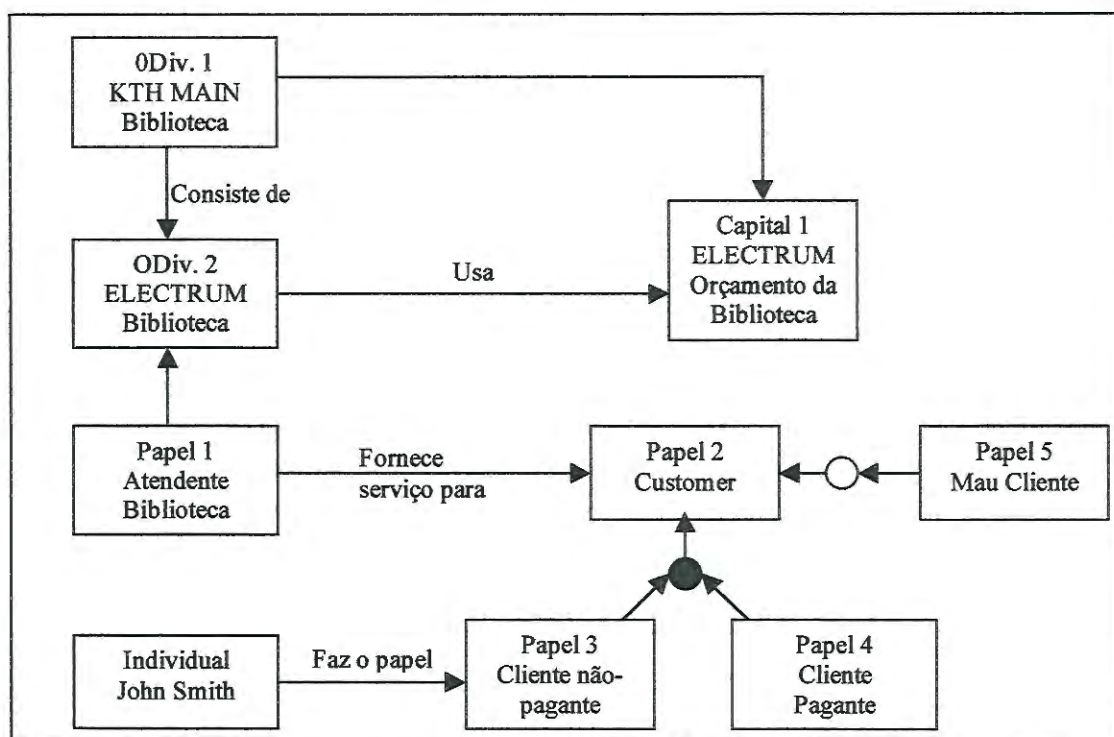


Figura 23 – Um exemplo de Modelo de Atores e Recursos de uma biblioteca

Fonte: BUBENKO *et al.* (2001), p.60.

3.6.6. Modelo de Requisitos e Componentes Técnicos

O que foi elaborado pelo Modelo de Objetivos, Modelo de Regras do Negócio, Modelo de Conceitos, Modelo de Processos do Negócio e Modelo de Atores e Recursos é uma descrição inicial dos objetivos da organização, das regras do negócio, dos processos, do sistema de atores das entidades de informação, de modo que os processos de negócio devam contribuir com os objetivos declarados. Se o objetivo é desenvolver um sistema de informação para apoiar processos, então existe a necessidade de tratar dos requisitos técnicos do sistema de informação, inicialmente de uma forma menos formal.

A abordagem EKD inclui um submodelo simples para descrever e para relacionar entre si, requisitos iniciais e não claros do sistema de informação. O Modelo de Requisitos e Componentes Técnicos se parece com a estrutura do Modelo de Objetivos e indiretamente com o Modelo de Sistemas de informação. Inicialmente é necessário desenvolver um conjunto de requisitos ou objetivos de alto nível para o sistema de informação como um todo. Baseado nesse

conjunto, o sistema de informação é estruturado em um número de sub-sistemas ou componentes técnicos. Para cada sub-sistema, é necessário definir um conjunto de objetivos e requisitos que são mais específicos. Esses objetivos e requisitos devem ser derivados e consistentes com os outros submodelos descritos anteriormente. O Modelo de Requisitos e Componentes Técnicos é uma tentativa inicial de se definir as estruturas e propriedades do sistema de informação para apoiar as atividades do negócio, como definido no Modelo de Processos do Negócio.

Segundo BUBENKO *et al.* (2001), os componentes presentes no Modelo de Requisitos e Componentes Técnicos são: Objetivos do Sistema de Informação, Problemas do Sistema de Informação, e Requisitos do Sistema de Informação.

Os Objetivos do Sistema de Informação são usados para expressar o alto nível de objetivos com relação ao sistema de informação e ou sub-sistema ou componentes. Eles podem ser expressos com propriedades mensuráveis ou não mensuráveis, focos, visões, ou direções. Os Objetivos do Sistema de Informação são tipicamente motivados por atividades do Modelo de Processos do Negócio, e podem ser motivados pelos objetivos do Modelo de Objetivos.

Os Problemas do Sistema de Informação são usados para expressar estados não desejáveis do negócio ou do ambiente, ou fatos problemáticos sobre a situação corrente com respeito ao sistema de informação a ser desenvolvido. Os Problemas do Sistema de Informação normalmente impedem os objetivos do sistema de informação.

Os Requisitos do Sistema de Informação expressam requisitos a serem designados para propriedades do sistema de informação. A propriedade pode ser funcional ou não-funcional. Uma expressão de requisitos sempre se refere a componentes do Modelo de Processos do Negócio e pode referir-se a componentes do Modelo de Atores e Recursos e Modelo de Conceitos. Os requisitos do sistema de informação podem apoiar ou impedir os requisitos funcionais e não funcionais do sistema de informação.

Os requisitos funcionais do sistema de informação são usados para expressar requisitos definitivos com relação à propriedade funcional do sistema de informação ou algum sub-sistema. Requisitos funcionais devem ser claramente definidos com referência ao Modelo de Conceitos. Preferivelmente, um requisito deve ser expresso de uma forma formal ou ao menos semiformal. Todos os conceitos de dados referentes aos requisitos funcionais podem ser diretamente apoiados pelos objetivos do sistema de informação, mas são mais freqüentemente vistos como refinamentos dos requisitos do sistema de informação. Requisitos funcionais são apoiados por componentes de outros submodelos, em particular o Modelo de Processos do Negócio, e também

pelo Modelo de Objetivos. Um requisito funcional deve ser relacionado a um processo ou a um sub-processo, definido no Modelo de Processos do Negócio.

Os requisitos não funcionais do sistema de informação são usados para expressar um tipo de requisito e restrição, considerando o sistema de informação a ser construído ou o processo de construí-lo. Requisitos não funcionais nem sempre são definitivos e podem algumas vezes ser negociados. Pode acontecer, por exemplo, de dois requisitos não funcionais não serem satisfeitos no mesmo nível completo, devido a alguma restrição financeira. Nesse caso, o nível de realização desses requisitos deve ser negociado. Alguns requisitos não funcionais podem impedir outros requisitos não funcionais, objetivos e problemas do sistema de informação. Eles podem ser apoiados ou apoiar objetivos e requisitos do sistema de informação. Eles podem ser relacionados ou apoiados pelos componentes do Modelo de Objetivos, e processos do Modelo de Processos do Negócio. Um requisito não funcional pode ser relacionado para um componente no Modelo de Atores e Recursos com o relacionamento “ator_envolvido”.

Os componentes da modelagem dos Objetivos e Problemas do Sistema de Informação são do mesmo tipo dos componentes do Modelo de Objetivos. Eles têm regras similares para nomeação e definição dos componentes do Modelo de Objetivos. Os objetivos do sistema de informação também deveriam iniciar com “O objetivo é...”. Entretanto, ao modelar os requisitos do sistema de informação, é importante focar nos requisitos e componentes do sistema de informação, e não na organização ou questões gerais. Os componentes desse submodelos também devem ser mensuráveis e fáceis de realizar verificação, para que eles formem a base do projeto do sistema de informação.

Expressões como “melhor do que”, “mais do que”, e “o melhor”, normalmente não contribuem para o entendimento de um requisito particular. Os componentes desse submodelo devem ser intimamente relacionados com os componentes do Modelo do Processo de Negócio, Modelo de Objetivos ou Modelo de Regras do Negócio.

O sistema de informação pode apoiar requisitos funcionais e não funcionais. O requisito funcional refere-se a uma propriedade funcional do sistema de informação como armazenar dados sobre os livros pedidos na biblioteca e para responder a consulta “listar todas as publicações sobre linhas de transmissão de voltagem publicado depois de 1994”. Um requisito não funcional pode ser de vários tipos diferentes tais como, restrições de operação, restrições políticas, restrições econômicas, restrições de segurança de informações. Um exemplo seria: a pesquisa do catálogo da biblioteca deve ser implementada em um sistema computacional XYW existente.

Os requisitos do sistema de informação devem apoiar os objetivos e processos da organização. O Modelo de Requisitos e Componentes Técnicos é usado como base para a Modelagem do Sistema de Informação. O MRCT permite a discussão de: métricas; operações para criar, modificar e deletar objetos e relacionamentos; consultas e objetos de pesquisa e relacionamentos e a funcionalidade para checagem e análise de dados.

As seguintes questões devem ser consideradas na elaboração do Modelo de Requisitos e Componentes Técnicos:

1. Quais restrições e padrões existem considerando a comunicação com sistemas ou hardware existentes?
2. Quais são os requisitos importantes considerando requisitos não funcionais do tipo segurança, disponibilidade, usabilidade, entre outros?
3. Quais restrições estão sendo consideradas no software existente ou nos sistemas que estão sendo desenvolvidos?
4. Quais restrições econômicas, pessoais, políticas, ou outras existem?
5. Existem restrições legais para desenvolvimento do sistema?
6. Esse requisito pode ser refinado mais claramente (talvez decomposto) em uma forma que possa ser verificado e mensurado?

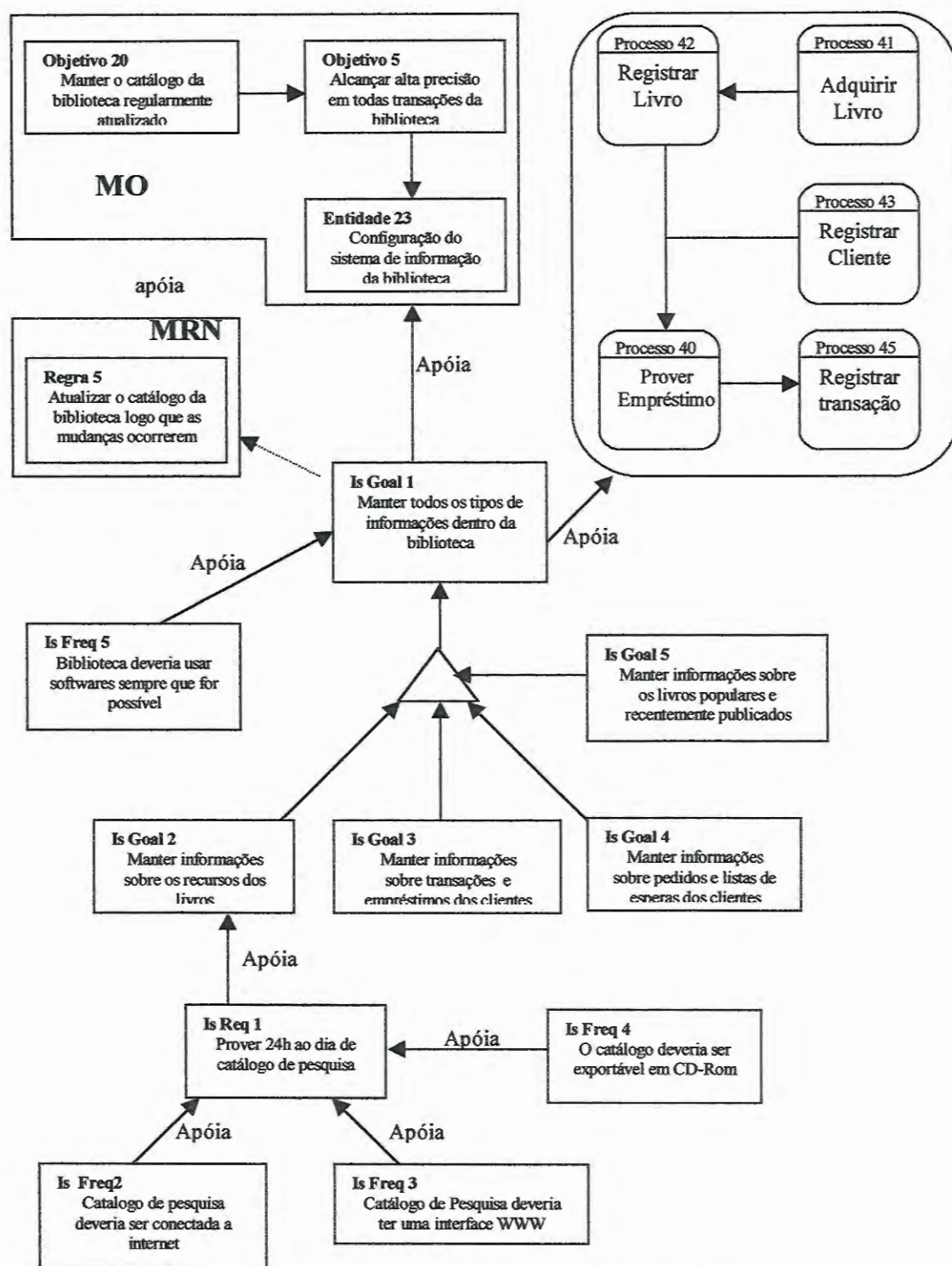


Figura 24 – Exemplo de um Modelo de Requisitos e Componentes Técnicos

Fonte: BUBENKO *et al.* (2001), p 64.

3.6.7. Relacionamento entre os submodelos

BUBENKO *et al.* (2001), explicam que no desenvolvimento de um modelo organizacional completo, as ligações entre os componentes de diferentes submodelos executam uma tarefa essencial. Por exemplo, declarações no Modelo de Objetivos permitem que conceitos diferentes sejam definidos mais claramente no Modelo de Conceitos, no qual uma ligação é especificada entre o componente correspondente no Modelo de Objetivos e o conceito no Modelo de Conceitos. Da mesma forma, objetivos no Modelo de Objetivos motivam processos particulares no Modelo de Processos do Negócio. Os processos são necessários para alcançar os objetivos declarados. Assim uma ligação é definida entre o objetivo e o processo que deverá ser realizado para cumpri-lo. Ligações entre modelos tornam o conhecimento mais disponível, sendo possível ver por que certos processos e requisitos do sistema de informação têm que ser introduzidos. De acordo com a figura 25, as ligações entre os submodelos do Modelo Organizacional são explicadas a seguir:

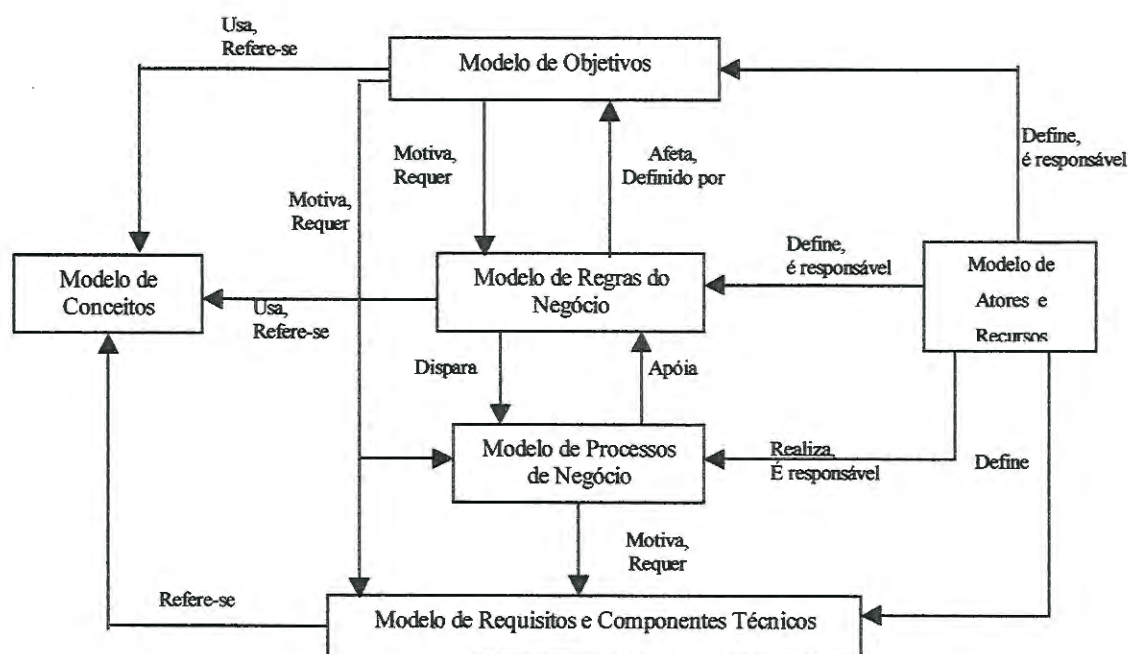


Figura 25 – Relações entre submodelos

Fonte: BUBENKO *et al.* (2001), p 68.

- Ligações entre o Modelo de Objetivos e o Modelo de Conceitos são normalmente usadas para descrever componentes do Modelo de Objetivos que referenciam entidades do Modelo de Conceitos. Por exemplo, o objetivo: “melhorar a satisfação do cliente” deveria referenciar o conceito “Cliente” no Modelo de Conceitos.
- Ligações entre o Modelo de Objetivos e o Modelo de Processos de Negócio tipicamente relacionam objetivos do Modelo de Objetivos a processos do Modelo de Processos de Negócio com o relacionamento “motiva”. Exemplo: “Melhorar a satisfação do cliente” poderia inicialmente motivar um particular processo de alto nível da organização com “Monitoramento das Relações com Cliente”.
- Ligações entre o Modelo de Objetivos e o Modelo de Atores e Recursos podem significar várias coisas: podem motivar ou requerer a introdução de novos atores particulares, como: Agentes de Relações de Clientes (motivado pelos objetivos de melhorar relacionamentos com clientes), ou elas podem descrever quais componentes do Modelo de Atores e Recursos são responsáveis para alcançar um particular objetivo ou defini-lo.
- Ligações entre o Modelo de Objetivos e o Modelo de Regras do Negócio tipicamente descrevem como componentes diferentes do Modelo de Objetivos são implementados em termos de regras do negócio no modelo de Regras do Negócio. Por exemplo, o objetivo “Registrar Maus Clientes” declarado no Modelo de Objetivos requer uma regra do negócio no Modelo de Regras do Negócio, que declara como exatamente isso deveria ser distinguido, como “Cliente é considerado mau cliente se demorar mais do que quatro semanas para entregar um livro”
- Ligações entre o Modelo de Regras do Negócio e o Modelo de Processos de Negócio normalmente descrevem como processos do Modelo de Processos de Negócio são disparadas pelas regras do negócio do Modelo de Regras do Negócio. Por exemplo, se existe uma regra que declara “Clientes são registrados como maus clientes se demorarem mais do que quatro semanas para entregarem o livro”, então essa regra dispara o processo que realiza o registro de maus clientes.
- Ligações entre o Modelo de Processos de Negócio e o Modelo de Conceito estão tipicamente entre o conjunto de informação do Modelo de Processos do negócio e componentes do Modelo de Conceitos. Por exemplo, o conjunto de informação “Vôo esperados” deveria referir-se a entidades incluindo atributos e relacionamentos como Vôo, Linha Aérea, Piloto e dados temporais sobre chegadas.

- Ligações entre o Modelo de Processos de Negócio e o Modelo de Atores e Recursos tipicamente descrevem como componentes distintos do Modelo de Atores e Recursos são relacionados ou envolvidos nos processos do Modelo de Processos de Negócio. Exemplos de nomes de ligações são: realiza, é responsável por e apóia. Por exemplo, ator auxiliar de biblioteca realiza o processo entrega livro para clientes.

- Ligações entre o Modelo de Atores e Recursos e o Modelo de Regras do Negócio normalmente descrevem como componentes diferentes do Modelo de Atores e Recursos são relacionados a regras do negócio do Modelo de Processos de Negócio. Exemplo de nomes de ligações são: define, é responsável por.

- Ligações entre o Modelo de Requisitos e Componentes Técnicos e outros componentes do modelo podem ser mais complexas do que os relacionamentos binários normais. Mais tipicamente, o Modelo de Processos do Negócio motiva os objetivos do sistema de informação. Simples declarações binárias, tais como “O sistema de catálogo da biblioteca deve ser capaz de manusear X pedidos simultaneamente”, mas pode também ter declarações mais complexas, tais como, “O tempo de resposta para o usuário do tipo X, quando estiver realizando o processo P, na data definida C, deve ser menor do que Z segundos”.

Os componentes do Modelo devem ser ligados de muitas formas. As ligações que devem ser estabelecidas dependem do propósito de um projeto EKD particular. Cada Modelo Organizacional tem um propósito e um foco que as ligações de cada modelo devem refletir. Toda ligação representa uma declaração feita sobre a organização e, possivelmente, seus requisitos do sistema de informação. A semântica de todas ligações deveria ser analisada cuidadosamente. Existe, entretanto, um conjunto mínimo de ligações que deveria ser definido pela representação para ser considerado completo. A figura 26 mostra algumas ligações entre os submodelos do caso da Biblioteca.

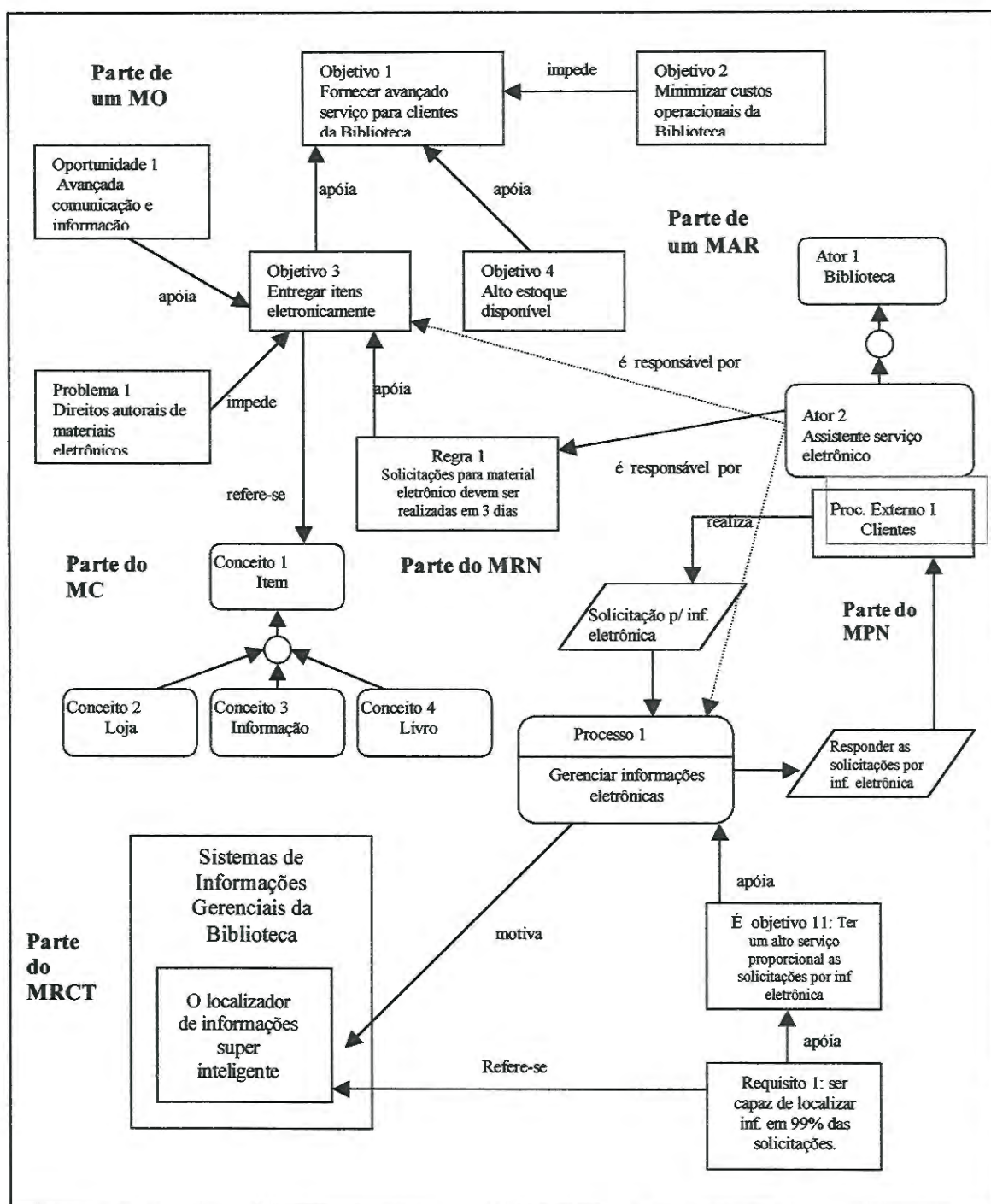


Figura 26 – Ligações entre submodelos do caso da Biblioteca

Fonte: BUBENKO *et al.* (2001), p.70.

3.7. Componentes para auxiliar a modelagem

BUBENKO *et al.* (2001) esclarece que algumas vezes é interessante estender o Modelo Organizacional com alguns componentes adicionais ou “ordinários”. As razões para usar esses componentes variam. Uma razão é, por exemplo, melhorar a expressividade do modelo, ou permitir mais liberdade para o grupo de desenvolvimento. Isso é mais comum nas primeiras sessões de modelagem, onde a tarefa mais importante é gerar e tornar os participantes familiarizados com o processo de modelagem. Os tipos de componentes que podem ser adicionados aos submodelos do MO não são estritamente prescritos ou definidos. Os requisitos básicos para adicionar componentes são: que todos os participantes do grupo aceitem e entendam o significado de cada componente, e que o facilitador de modelagem aceite a extensão como benéfica. Alguns dos componentes adicionados podem ser mais tarde reformulados e usados como tipos de componentes regulares.

Os componentes adicionados mais comumente são: comentários, ações para desenvolvimento do modelo, suposições e cenários.

Os comentários normalmente esclarecem alguns componentes de modelagem, ou todo o modelo. Pode ainda conter alguma informação que o grupo de modelagem acha importante, embora seja difícil expressar em termos de modelagem de componentes ordinários. Comentários podem também conter dicas para melhor elaboração do modelo. Se os comentários enfocam em algum componente de modelagem particular, estes são, então, interligados por setas.

As ações para o desenvolvimento do modelo são usadas para expressar ações concretas ao desenvolvimento do modelo, ou refinamento de um componente particular de modelagem ou um grupo de componentes.

As suposições expressam tipos hipotéticos de fatos, que podem ser preparados para um outro tipo de componente de modelagem por tipos de ligações tais como: motiva, apóia, impede, conflito, entre outros. Depois, durante a fase de elaboração e refinamento do modelo, as suposições podem ser transformadas em oportunidades, problemas, fraquezas, ameaças, objetivos, processos, requisitos do sistema de informação.

Os cenários são usados para representar uma série de eventos alternativos possíveis, processo, ou questões, os quais podem ser descritos ou elaborados no modelo.

Um outro tipo de prática de modelagem apresentada por BUBENKO *et al.* (2001), é colocar os componentes de modelagem em grupos. Essa prática é comum na sessão

intermediária de modelagem, quando há uma estrutura de componentes de modelagem solta, sem relacionamento com as outras estruturas, uma situação comum que surge quando uma sessão de modelagem é muito pequena, ou o modelo não é completo por alguma razão.

Existem muitas formas de agrupar categorias como por conteúdo (por exemplo, problemas de falta de competitividade, problemas de produção), por prioridade (como processos críticos do negócio) e por natureza (objetivos para mudanças, objetivos para estado futuro do negócio). Por exemplo, depois de uma reestruturação inicial do Modelo de Objetivos, é possível incluir grupos de objetivos estratégicos, objetivos de marketing e objetivos de produção. Depois, cada um desses grupos pode ser repostos como um simples objetivo, decomposto em um grupo separado contendo sub-objetivos que são membros do grupo, ou conectados juntos de qualquer forma.

3.8. Utilização da abordagem EKD

Segundo BUBENKO *et al.* (2001), a abordagem EKD está longe de ser constituída de apenas um produto, o Modelo Organizacional e seus submodelos. O sucesso de aplicação do EKD depende inteiramente da forma pela qual é introduzido na organização e na forma em que o processo de desenvolvimento é conduzido. No trabalho de BUBENKO *et al.* (2001) são apresentadas algumas diretrizes para introduzir e usar a abordagem EKD. Mesmo a abordagem e seus predecessores tendo sido aplicados por muitos anos, muito conhecimento novo é ainda observado e desenvolvido, considerando-se a aplicação da abordagem. Essas diretrizes devem ser consideradas como conhecimento em constante evolução e extensão. Um grande número de experiências em Modelagem Organizacional (ou Modelagem de Negócio) tem sido feito na Suécia e resultaram em:

- Descrições Claras e com rigor adicional;
- Uma evolução na aprendizagem organizacional;
- Ajuda a realização de mudanças e reengenharia serem mais facilmente aceitas na organização.

Neste trabalho foi observado, que a abordagem é difícil de explicar, e que existe um risco de levar crenças em mudanças mágicas na organização, ou seja, acreditar que a abordagem automaticamente ajudará a resolver todos os problemas da organização. Existe a necessidade de

muita disciplina para mover-se da fase inicial até versões de modelos melhorados usando ferramentas computacionais. A abordagem EKD pode ser aplicada por muitas razões diferentes e, sendo assim, proporcionará saídas e resultados diferentes. A saída desejada é chamada de alvo na abordagem EKD. Em BUBENKO *et al.* (2001), diferentes alvos são descritos:

- Uma companhia deseja desenvolver uma estratégia para desenvolvimento em longo prazo de seus recursos humanos. Essa aplicação irá concentrar-se no Modelo de Objetivos. Quais são os objetivos gerais em longo prazo? Quais problemas são enfrentados e quais ameaças e restrições existem? Mas esse tipo de análise pode muito bem introduzir a necessidade de melhorar a análise conceitual e modelagem de conceitos essenciais para o problema em questão. Por exemplo, o que se entende por recursos humanos, como avalia-los, o que se entende por competência e como avaliá-la, que tipo de competência será necessária no futuro, considerando os objetivos declarados. Essa análise poderá trazer o desenvolvimento do Modelo de Processo de Negócio. Novos tipos de posições, tarefas, e habilidade podem requerer ainda o desenvolvimento do Modelo de Atores e Recursos.

- Uma questão geral pode ser formulada como “o sistema corrente satisfaz a necessidade de apoio de informações para a série de estratégias em longo prazo. Se não, o que deve ser mudado para alcançar uma solução satisfatória?” Inicialmente, os objetivos da situação futura devem ser analisados. Depois, baseado nesse conjunto de objetivos, o conjunto de processos e regras do negócio deve ser examinado e reprojeto. Um novo Modelo de Atores e Recursos será provavelmente desenvolvido. O conjunto atual de sistemas de informação deve ser descrito e suas propriedades gerais dadas no Modelo de Requisitos e Componentes Técnicos (MRCT). Um modelo do futuro conjunto de componentes técnicos necessários e seus requisitos devem ser desenvolvidos baseados na documentação de objetivos, processos, conceitos e atores para um futuro sistema de negócio. Uma comparação entre as propriedades do sistema atual técnico e requisitos do futuro conjunto de sistema de informação fornece a base para análise das necessidades para mudanças. Claramente, em muitos casos, alternativas diferentes do futuro sistema de informação pode ser analisado.

A seguir algumas dicas, serão apresentadas segundo BUBENKO *et al.* (2001), dando uma pequena noção de vários problemas que poderão surgir:

1. Para introduzir a abordagem na companhia, é importante que esta seja aceita por todos. Para isso a gerência ou o responsável pelo orçamento deve estar convencido dos efeitos benéficos da abordagem. Outros funcionários da companhia, importantes para o projeto, devem ser convencidos de que o processo EKD não ameaçará seu emprego e posição. Aplicações custarão tempo e recursos humanos. Parece ser razoável entender que pessoas são em geral, cuidadosas em aceitar uma nova forma de trabalho, considerando as centenas oferecidas por firmas de consultorias. Nesse estágio inicial, pode ser útil mostrar histórias de sucesso referente a companhias que usaram EKD.
2. O projeto EKD deve ter uma missão e expectativas de resultados bem definidos. A proposta deve ser clara a todos os envolvidos.
3. O projeto EKD não pode ser executado com sucesso sem recursos suficientes e autoridade. Pessoas envolvidas devem ter tempo para participar e recursos computacionais (entre outros) devem se alocados.
4. Um projeto EKD tipicamente envolve um número variado de atores, tais como, gerente de projeto, o comitê de direção, grupo de referência, participantes da modelagem (tipicamente e 5 a 8 pessoas), facilitador da modelagem, técnicos na modelagem e outros. Entretanto, em um projeto pequeno não são necessários todos esses atores.
5. Ninguém pode conhecer todos os aspectos possíveis de um problema de projeto. Deve-se tentar compor um grupo abrangente para cobrir aspectos diferentes do problema em questão.
6. Não é aconselhável iniciar a primeira sessão de modelagem sem primeiro reunir os membros do grupo de modelagem individualmente. A reunião pode preferivelmente ser conduzida como uma entrevista. O entrevistador deveria ser o gerente do projeto ou facilitador. A entrevista normalmente dá um entendimento melhorado do que deveria (ou poderia) ser feito no primeiro seminário de modelagem.
7. Baseado nas entrevistas realizadas, o gerente de projeto e o facilitador de modelagem juntamente com outros membros do time técnico, devem preparar um plano de ação do primeiro seminário de modelagem. Esse plano deveria incluir as maiores questões para serem passadas

ao grupo e ser realizada uma análise dos possíveis resultados e como estrutura-los. Possíveis obstáculos e conflitos devem ser analisados.

8. O primeiro seminário é realizado usando uma grande tela para anotações na parede, onde os participantes colocam suas idéias de diferentes componentes dos modelos usados, tais como: objetivos, problemas e oportunidades. A primeira sessão é normalmente iniciada com uma pequena introdução dos submodelos EKD e como o processo de modelagem será procedido. As diretrizes mostram como formular objetivos, problemas, entre outros. O primeiro seminário não deveria ter um tempo menor de 3 ou 4 horas. O resultado desse seminário é normalmente um gráfico não estruturado cobrindo alguns dos vários submodelos.

9. A tarefa dos técnicos em modelagem é, primeiramente, transportar o modelo da camada plástica para um modelo computadorizado, depois, o modelo é analisado e reestruturado. Declarações redundantes são eliminadas, componentes fortemente relacionados são aproximados. Descrições e relacionamentos são analisados criticamente. Componentes são generalizados e especializados (detalhados). Questões e comentários críticos são documentados.

10. Depois de refinar o modelo, os técnicos têm que consultar os participantes da modelagem usando questões e comentários críticos do passo anterior. Possíveis mudanças e a extensão do modelo são discutidas com participantes individualmente.

11. As discussões do passo anterior fornecem informações para que técnicos em modelagem possam melhorar o modelo. Esse é um estágio também necessário para desenvolver uma descrição do modelo e familiarizar os participantes de modelagem para outras pessoas envolvidas no processo.

12. Depois que a nova versão do modelo foi concluída, incluindo textos e gráficos que agregados estão no computador, o seminário *walkthrough* deverá ser realizado em um laboratório equipado com meios suficientes para expor o resultado do grupo de pessoas. O local do seminário deve ter um projetor com uma boa resolução. A tarefa do facilitador de *walkthrough* é agora expor os resultados obtidos para o grupo de modelagem, e convidá-lo para criticar e dar comentários construtivos e sugestões para mudanças. Todos os comentários e sugestões são documentados.

13. O trabalho baseado em dois seminários de modelagem normalmente não resulta em um modelo suficiente e completo. Análises adicionais e trabalho de reestruturação e *walkthrough* adicionais são necessários para alcançar um resultado útil. Trabalho contínuo, entretanto, tipicamente precede como interações incluindo trabalho entre o primeiro seminário e o *walkthrough*. A regra de parada é um conjunto de modelos descrito em detalhes suficientes que podem ser usados como base para um bom trabalho contínuo de implementação das sugestões dos modelos.

Em BUBENKO *et al.* (2001) é apresentado um conjunto de pré-condições para organizar a aplicação de um projeto EKD:

1. Deve ser passada uma missão clara para todo time de modelagem.
2. Devem ser alocados tempo e recursos suficientes para a atividade.
3. A composição do grupo de modelagem deve ser baseada na idéia de que o time, coletivamente, tenha conhecimento em todos os campos necessários tais como: estratégias de negócio, objetivos, computação, software, sistema de informação, gerenciamento, questões operacionais, entre outras.
4. O grupo de modelagem deve ter a autoridade para reprojeter a organização.
5. Responsabilidades devem ser designadas considerando a documentação, uso e manutenção do Modelo Organizacional a ser desenvolvido.
6. A atividade de modelagem tem que ser bem planejada com consideração:
 - Às questões a serem discutidas;
 - Aos participantes envolvidos;
 - À alocação de tarefa;
 - Aos participantes sendo alocados em tempo;
 - Às expectativas para ser completadas;
 - Aos participantes recebendo treinamento no uso da Modelagem Organizacional antes do início da sessão de modelagem e;
 - À participação de um facilitador experiente.

O gerente e os participantes do processo de modelagem devem entender completamente e concordar com todos os aspectos do projeto. O propósito, objetivos e escopo do projeto devem ser documentados. A alocação de recursos (pessoal, responsabilidade, tempo, dinheiro, recursos

computacionais) deve ser determinada. Garantia de qualidade em termos de resultados e validação deve ser mantida e registrada.

As seguintes habilidades (algumas delas podem ser executadas por uma pessoa) são necessárias no projeto de modelagem:

O gerente de projeto, sendo da organização será responsável por:

- Planejamento do projeto;
- Liderar o projeto e
- Informar a gerência e o comitê de direção.

Comitê de Direção será tipicamente responsável por:

- Apoio e “venda” do projeto dentro da organização;
- Plano de trabalho;
- Alocação de recursos;
- Controle de qualidade e
- Avaliação.

Grupo de referência é normalmente formado por pessoas com conhecimento no domínio e outros *stakeholders* do projeto, que são responsáveis por:

- Suprir o domínio de conhecimento e informação;
- Ajudar o facilitador com a estruturação de descrição dos modelos;

O grupo deve ser composto por pessoas:

- De várias partes da organização com conhecimento e visões de abrangências muito amplas;
- Que consignam sugerir mudanças organizacionais;
- Entusiasmadas, de mente abertas e cooperativistas.

O facilitador (pode ser recrutado fora da organização), uma ou duas pessoas que:

- Lidera(m) e adverte(m) durante as sessões de modelagem e
- Apóia(m) cada um na aquisição de conhecimento e idéias do grupo de aplicação

“Equilibrista de Ferramenta” é uma pessoa que tenha facilidade para lidar com ferramentas computacionais o suficiente para ajudar o facilitador na sessão *de walkthrough*.

O projeto também precisa de um Especialista em Ferramenta, responsável por modelar todo o desenvolvimento e realizar possíveis adaptações.

Um “Contador de Histórias” ou apresentador é necessário para fornecer aos participantes de modelagem introdução eficiente para as principais questões apresentadas no modelo.

Projetistas são importantes pela habilidade de ilustrar resultados essenciais das sessões de modelagem.

Especialistas em Comunicação é responsável por desenvolver e adaptar técnicas e ferramentas para o trabalho síncrono ou assíncrono nos ambientes distribuído para indivíduos e grupos.

O primeiro seminário não deve falhar. Caso falhe, será difícil convencer o grupo a continuar a se dedicar. Por essa razão deve ser bem preparado. Se o projeto é aceito pela gerência e as pré-condições necessárias são encontradas, algumas questões devem ser tratadas no estágio de preparação. As linhas gerais são apresentadas de acordo com BUBENKO *et al.* (2001):

1- Selecionar as pessoas que sejam familiarizadas com o problema. Algumas vezes pode ser benéfico ter o lado do “consumidor” e do “produtor” representando uma visão mais abrangente. Em alguns estágios do projeto, pode ser necessário associar especialista de certas áreas do projeto. Esses especialistas podem sugerir soluções organizacionais ou de tecnologia da informação para satisfazer objetivos específicos declarados como a reengenharia de alguns processos de negócio ou desenvolvimento de alguns tipos de soluções de tecnologia da informação.

Na abordagem “*top-down*”, o ponto inicial é a modelagem de objetivos, na qual é necessário muito raciocínio para discutir o que se deseja alcançar. Por isso, é necessário que os participantes estejam diretamente envolvidos ou tenham conhecimento da tomada de decisão e formulação dos objetivos em um nível pertinente da organização, sendo operacional ou estratégico. Por outro lado, a abordagem “*bottom-up*”, tipicamente usada na reestruturação de processos, deve ser iniciada com o processo de modelagem e pode requerer a participação de pessoas envolvidas nos processos para descrever e documentar processos organizacionais. Em todas as situações, no entanto, pode ser necessário mudar membros de um grupo, uma vez que as

discussões e modelos movem de uma área para outra, sendo necessário pessoas com conhecimentos diferentes.

Não é aconselhável ter um grupo em que os integrantes trabalhem juntos com a mesma atividade. Essa situação não permite o pensamento criativo, e até certo ponto ocorre uma repetição de idéias e visões conhecidas. O mesmo problema pode surgir se um superior direto estiver participando do grupo. É aconselhável que os membros do grupo não se conheçam para estimular a troca de novas idéias, e deve-se evitar grandes diferenças de idades que podem estimular preconceitos.

2- Entrevistar os especialistas em aplicações. O objetivo do seminário de modelagem é chegar a um consenso coletivo orientado às decisões. Isso deveria ser herdado com preparação adequada. O facilitador deveria entrevistar os participantes para:

- Preparar-se e tornar-se familiarizado com cada participante;
 - Ganhar conhecimento do domínio do problema e de questões essenciais relacionadas;
 - Obter informações que irão melhorar a sua habilidade em conduzir os seminários de modelagem, conseguindo apresentar diretrizes para os participantes concentrarem-se nas questões.
- Preparar-se para possíveis conflitos e para a necessidade de balancear um grupo ou pessoas dominantes que inibem a participação de outros.

3- A entrevista deveria focalizar as visões do entrevistado, os objetivos, suposições, os problemas enfrentados e as restrições observadas na área em questão. Supondo que a companhia tem o nome de COMP, uma divisão da companhia chamada DIV com uma função chamada F. Assume-se aqui que a proposta do projeto é analisar F e sugerir diferentes possibilidades de melhoramentos. As seguintes questões podem ser consideradas:

- Como você descreveria a função F, suas tarefas e atividades correntes dentro de DIV e COMP?
- De acordo com sua opinião, descreva algumas questões importantes em F a fim de que sejam consideradas nos próximos 3 a 5 anos?
- Descreva alguns problemas correntemente experimentados por DIV com a função F?
- Sugira objetivos de pequena e grande abrangência para a função F?
- O que faz de F uma função necessária em DIV?

- Em sua opinião quais são as forças e fraquezas atuais da função F?
- Quais oportunidades existem na área de F?
- Quais restrições externas você gostaria de mencionar considerando F?
- Quais tendências externas podem influenciar a operação de F? Como?
- Quais decisões importantes, com conseqüências abrangentes, nós teremos que tomar dentro de um ano considerando F?
- Você vê algum problema em realizar essas decisões?

4- As entrevistas fornecem ao gerente do projeto e facilitador uma visão melhorada das pessoas que participam nas sessões de modelagem e de suas visões, problemas, esperanças, preconceitos e sentimentos. Possibilitando ao facilitador a oportunidade de planejar o início de uma sessão de modelagem, como conduzi-la, e como lidar com situações inesperadas.

5- Em geral o primeiro seminário deveria ser organizado de tal forma que promovesse trabalho concentrado. Sendo interessante que o local seja diferente daquele em que, usualmente o grupo se reúne, ou que não seja associado á organização, passando uma atmosfera mais tranqüila e sem interrupções.

6- A sala deve conter:

- Ao menos uma parede livre (3m x 2m) para fixar uma tela de anotações.
- Cadeiras e mesas para todos os participantes, bem distribuídas na sala, de forma que ninguém fique escondido ou impossibilitado de acompanhar o seminário. Não deve haver distrações, como telefone.
- Canetas (com tintas não permanentes) de cores diferentes para cada participante para facilitar a identificação.
- Apagador e tesoura.
- Papéis com adesivos (para anexar na tela para anotações).
- Folhas com os nomes dos componentes para fazer anotações.

Uma pequena introdução deve ser dada no início do seminário para cada um dos seguintes itens:

1. Todos os presentes;
2. O principio do EKD;

3. A agenda do dia;
4. O(s) tópico(s) para discussão;
5. As regras para o máximo de produtividade:
 - Todos devem participar;
 - Tudo que for importante deve ser escrito em sentenças completas;
 - Todos devem ouvir o outro e pensar individualmente;
 - Todos devem se esforçar para balancear e ter consenso no resultado.

O objetivo do EKD não é apenas o Modelo Organizacional e sim obter um melhor entendimento para resolver problemas e desenvolver o conhecimento da organização. Para tanto, é necessário um estudo analítico e crítico do Modelo Organizacional e seus relacionamentos, baseado em um bom entendimento dos princípios da modelagem organizacional. Essa análise pode ser alcançada com a ajuda de questões. A lista de questões citadas a seguir não é completa, mas apenas exemplos que deverão ainda ser expandidos.

- Cada objetivo é apoiado por um processo no Modelo de Processo? Se não, por que não?
- Deveria ser introduzido o processo?
- Quem é responsável por esse processo no Modelo de Atores e Recursos?
- Deveria haver um investimento em um novo recurso para ajudar a execução desse processo?
- Esse recurso precisa de um sistema de informação novo ou melhor?
- Pode-se identificar no Modelo de Requisitos e Componentes Técnicos, os requisitos para o sistema de informação?
- Existem regras do negócio que podem colocar restrições nos requisitos?
- Existe uma definição comum da organização para essas restrições e requisitos no Modelo de Conceitos?

De acordo com BUBENKO *et al.* (2001), quando se procura por relacionamentos e inconsistências e se descobrem lacunas, aumenta-se o conhecimento e entendimento da organização. O EKD torna o processo cognitivo de análise mais fácil através dos gráficos e das técnicas de representação estruturada na forma de Modelo Organizacional. Existem muitos perigos quando se envolve a comunicação de idéias entre humanos. No caso específico do EKD incluem-se:

- Evitar iniciar a modelagem com longas explicações de conceitos abstratos.
- Começar com atividades, objetivos ou processos práticos e bem conhecidos.
- Evitar modelos desestruturados.
- Conduzir a reestruturação e esclarecimento de atividades logo depois da sessão de modelagem, caso contrário, muitas informações inerentes no modelo desestruturado serão esquecidas.
 - Evitar descrições com poucas palavras que não são intuitivamente entendidas.
 - As descrições devem ser escritas cuidadosamente para que mais tarde elas possam ser facilmente entendidas.
 - Não colocar objetivos que não contribuem para a missão da organização.
 - Todos objetivos auxiliam uns aos outros. Não devem existir objetivos soltos.
 - Evitar compor declarações que tenham muitos relacionamentos internos e externos, porque isso dificulta o entendimento e análise.
 - Evitar detalhar atributos antes que todo o contexto tenha sido estabelecido.
 - Os atributos não são todos relevantes.
 - Não verbalizar o que é aparente no modelo.
 - Evitar entidades duvidosas.

A primeira sessão é tipicamente uma atividade de *brainstorming*, sendo então necessária uma estrutura bem montada. Podem existir componentes redundantes, com dois objetivos que significam a mesma coisa, ou faltar componentes e relacionamentos importantes. A terminologia escrita pelos participantes pode ser ambígua. Depois do *brainstorming* vem a maior parte do trabalho, que é estruturar e analisar o resultado da primeira sessão, buscando torná-la apresentável para a próxima sessão. Para melhorar o primeiro modelo é necessário:

1. Introduzir relacionamentos. Os modelos começam a ter mais significado através dos relacionamentos entre os componentes. Os relacionamentos implícitos e indesejáveis devem ser descobertos e ajustados.

2. Uma classificação sistemática deve ser criada. A criação da estrutura depende do critério usado, da disponibilidade e uso de uma ferramenta computacional, do modelo a ser simplificado e se ter uma terminologia clara.

Antes da próxima sessão, os participantes devem ter em mãos todo o material da sessão anterior para que seja possível estudar o material e refletir sobre o conteúdo como uma introdução para a próxima sessão.

No seminário de *walkthrough* os analistas apresentam para o grupo de aplicação o trabalho feito desde a primeira sessão. Durante o seminário os seguintes itens deverão ser realizados:

- Revisão do trabalho na primeira sessão.
- Realização de correções e/ou adição para modelos e descrições.
- Estreitar o campo de discussão e especificar o domínio.
- Criar novos modelos;
- Expandir modelos anteriores.
- Alocar tarefa para os participantes.
- Sugerir mais trabalhos e direções futuras.

O estágio inicial da revisão deveria ser liderado pelos analistas que trabalharam com o modelo original e produziram o estado corrente. O segundo estágio deveria ser liderado pelos facilitadores que conduziram a primeira sessão de modelagem para assegurar a continuidade. Todas as modificações devem ser justificadas.

3.9. Considerações Finais

É incontestável a importância da definição de requisitos, considerando-se que o êxito no desenvolvimento de software depende diretamente da fidelidade com que são atendidas as necessidades dos usuários.

A definição apropriada dos requisitos depende de um preciso conhecimento do ambiente e das razões pela quais o software é proposto.

A técnica de Modelagem Organizacional EKD é composta por submodelos que representam diferentes aspectos da organização e que vistos de forma integrada, representam as características e requisitos da organização.

Neste sentido, o EKD torna o processo cognitivo de análise mais fácil através dos gráficos e das técnicas de representação estruturada na forma do Modelo Organizacional, porém o objetivo da técnica EKD não é apenas o Modelo Organizacional e sim obter um melhor entendimento para resolver problemas e desenvolver o conhecimento da organização.

Através da aplicação da técnica EKD é possível chegar a um repositório do conhecimento, que pode ser usado para discutir mudanças no negócio e componentes do Sistema de Informação e traçar uma cadeia de componentes e decisões que leva a várias interpretações das decisões e dos componentes de informações.

No próximo capítulo será realizada uma revisão bibliográfica sobre a abordagem *Use Case*.

4. Abordagem *Use Case*

4.1. Considerações Iniciais

Por muito tempo, tanto na orientação a objeto como nas abordagens estruturadas, técnicas baseadas em cenários têm sido utilizadas na Engenharia de Software para entender, modelar e validar os requisitos dos usuários. Uma das abordagens de cenários que tem se destacado é o *Use Case*. Esta técnica é parte integrante da Linguagem de Modelagem Unificada (UML) (BOOCH *et al.*, 1999), que tem sido adotada como padrão de linguagem de modelagem para o desenvolvimento de software orientado a objeto.

Essa técnica tenta retratar e suprir as principais falhas encontradas nos métodos tradicionais por intermédio de uma abordagem que leva em consideração a visão do usuário na interação com o sistema, fazendo com que a coleta de requisitos atenda aos seus anseios e tornando desta forma essencial sua participação no processo de desenvolvimento de software.

Apesar do consenso e do reconhecimento de cenários como uma ferramenta importante no processo de engenharia de requisitos, pode-se apontar várias carências no que diz respeito à técnica. Este capítulo tem como objetivo apresentar o diagrama *Use Case* como uma técnica para capturar e descrever os requisitos funcionais do sistema e expor segundo a visão de diversos autores os principais problemas relacionados a este.

4.2. UML

A UML (Unified Modeling Language) é uma linguagem para especificação, construção, visualização e documentação de sistemas de software.

A UML é o resultado da união das metodologias de orientação a objetos de maior uso como RUMBAUGH, BOOCH e JACOBSON originando uma modelagem única e padronizada pelo OMG (*Object Management Group*).

A UML pode ser usada para mostrar as fronteiras de um sistema e suas funções principais, utilizando diagramas *Use Case*; ilustrar a realização de *use cases* com diagramas de interação; representar a estrutura estática de um sistema, empregando os diagramas de classes;

modelar o comportamento do objeto com diagramas de estado e revelar a arquitetura de implementação com diagramas de componentes e diagramas de implantação.

O processo de desenvolvimento de software com a UML, está estruturado, segundo o tempo em 4 (quatro) fases:

- **Concepção:** nesta fase é estabelecido o caso de negócio para o sistema e delimitado o escopo do projeto.
- **Elaboração:** Consiste em uma análise mais refinada do sistema a ser construído, juntamente com um plano detalhado do trabalho a ser feito.
- **Construção:** durante esta fase, um produto completo é desenvolvido de maneira interativa e incremental para que esteja pronto para transição a comunidade usuária.
- **Transição:** o software é disponibilizado a comunidade usuária.

Em cada uma destas fases diferentes artefatos são produzidos usando diferentes atividades e técnicas.

Em cada uma das fases do processo, têm-se as fases normais do ciclo de vida do software:

- **Análise de Requisitos:** a meta desta fase é descrever o quê o sistema deve fazer, permitindo a desenvolvedores e usuários concordarem sobre uma descrição comum.
- **Design:** a meta desta fase é mostrar como o sistema será materializado na fase de implementação.
- **Implementação:** esta fase ocorre no instante em que as classes necessitam ser programadas. O sistema é percebido através da implementação produzindo-se os arquivos de código fonte que resultarão em um sistema executável.
- **Teste:** o objetivo desta fase é identificar erros no sistema, através de um número de caso de testes que avaliem diferentes aspectos de cada modelo sendo testado.

4.3. Modelagem *Use Case*

A modelagem *Use Case* foi criada por Jacobson, baseado em suas experiências no desenvolvimento de um sistema para a Ericsson com a utilização dos métodos OOSE e *Objectory*.

É uma técnica de modelagem usada para descrever o que um novo sistema deverá fazer ou para descrição de um sistema já existente.

Dentre as cinco fases de desenvolvimento de sistemas existentes (análise de requisitos, análise, projeto, programação e testes), a modelagem *Use Case* está relacionada com a primeira delas, pois conta com os diagramas *use cases* para capturar os requisitos solicitados pelo cliente. Através dessa modelagem pode-se ter uma imagem de como será o funcionamento do sistema, sem se preocupar com a implementação do mesmo.

Na *visão use-case* o sistema é visto como uma caixa-preta que fornece situações de aplicações (*use cases*). Neste modelo não é importante saber como o sistema implementa os *use cases* ou como ocorre o funcionamento interno.

A construção desses modelos é feita a partir de várias discussões entre as pessoas envolvidas com o sistema a ser modelado: desenvolvedores, clientes e usuários finais. A participação deles durante a modelagem é extremamente importante, pois o modelo será constantemente adaptado de forma a refletir em detalhes as necessidades de clientes/usuários.

Os componentes primários de um modelo *Use Case*: são os atores, os *use cases* e o sistema modelado. Os limites deste sistema são definidos pela funcionalidade que o sistema exige. A funcionalidade de todo o sistema é representada por vários *use cases* e cada um deles também especifica uma funcionalidade completa.

O propósito primário dos *use cases* são:

- Decidir e descrever a funcionalidade do sistema a partir do entendimento entre os desenvolvedores do software, cliente e/ou usuário final;
- Fornecer uma descrição consistente e clara sobre o que o sistema fará, pois esta documentação será usada durante todo o processo de desenvolvimento e possivelmente em outros projetos;
- Servir de base para testes de verificação.

A modelagem *Use Case* não é apenas usada para reconhecer os requisitos do sistema, ela também pode ser usada quando novas versões de sistemas são desenvolvidas.

A criação do modelo *Use Case* envolve as seguintes tarefas.

- Definição do sistema;
- Encontrar os atores e os *use cases*;
- Descrever os *use cases*;
- Definir os relacionamentos entre *use cases*;
- Validar o modelo.

O modelo é formado por diagramas *Use Case*, com atores, *use cases* e seus relacionamentos. Tais diagramas oferecem uma visão geral do modelo, mas as descrições reais dos *use cases* são feitas através de texto, pois modelos visuais não são capazes de fornecer toda a informação necessária. O modelo *Use Case* representa a visão *use case* do sistema. Esta visão é muito importante, pois tem influência sobre outras visões existentes em um sistema.

4.4. Diagrama *Use Case*

Em UML, um modelo *Use Case* é descrito como um diagrama *Use Case*, e um modelo pode ser dividido em outros diagramas *Use Case*.

Diagramas *Use Cases* permitem descrever a visão externa do sistema e suas interações com o mundo exterior, representando um alto nível de funcionalidade. Esses diagramas contêm elementos do modelo para o sistema (atores e *use cases*) e mostram diferentes relacionamentos, tais como generalização, associação e a dependência entre os elementos. Conforme a figura 27, os quatro elementos básicos de um diagrama são:

- Ator;
- *Use case*;
- Relacionamentos;
- Sistema.

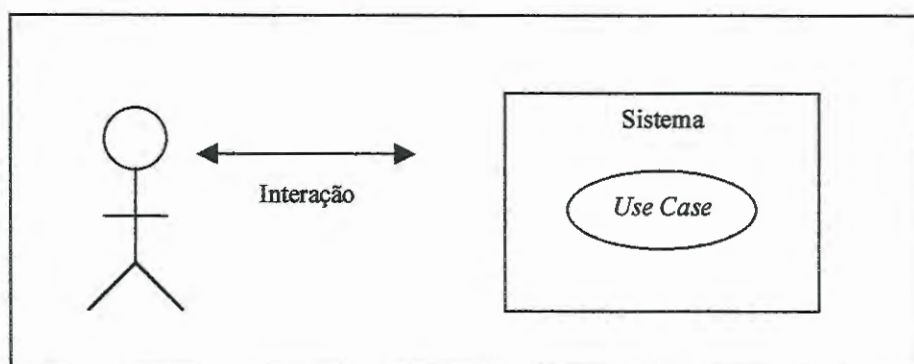


Figura 27 – Diagrama *Use Case*.

Fonte: FURLAN (1998), p170.

O propósito primário do diagrama *Use Case* é:

- Descrever os requerimentos funcionais do sistema de maneira consensual entre usuários e desenvolvedores de sistemas;
- Fornecer uma descrição consistente e clara sobre as responsabilidades que devem ser cumpridas pelo sistema, além de formar a base para a fase de desenho;
- Oferecer as possíveis situações do mundo real para o teste do sistema.

4.5. Atores

Um ator é alguém ou algo que interage com o sistema. Isso significa que o ator recebe, envia mensagens e troca informações com o sistema. Eles representam o papel de uma entidade externa ao sistema como um usuário, um hardware ou outro sistema que interage com o sistema modelado.

Um ator é um tipo (uma classe) e não uma instância, ele representa um papel e não um usuário do sistema. Por ex: se uma pessoa quer locar uma fita de vídeo, é o seu papel como locatário que estamos interessados em modelar e não a própria pessoa. Dependendo do seu papel no sistema, uma pessoa pode ser vários atores. Os papéis que alguém pode ter no sistema também podem ser restringidos. Por ex: continuando o exemplo da vídeo locadora, uma pessoa pode ser proibida de ser o caixa e o atendente do balcão.

Os nomes atribuídos aos atores devem refletir seus papéis e não uma instância específica de um ator ou a funcionalidade do ator. A comunicação entre um ator e o sistema é feita através de mensagens, de forma similar ao que se pode encontrar na programação orientada a objetos.

Os atores são classificados como:

- Primário: é um ator que utiliza funções primárias do sistema. Por ex: Em um sistema de uma vídeo locadora um ator primário é aquele que verifica o cadastro do locatário e entrega uma fita a ele.
- Secundário: é um ator que utiliza funções secundárias do sistema. Continuando com o exemplo, este tipo de ator estará relacionado com as funções de administração do sistema. Ex: gerenciamento de banco de dados de clientes e fitas.

Ambos os tipos são modelados para assegurar que seja descrita a completa funcionalidade do sistema, mesmo que as funções de maior interesse do cliente/usuário sejam as

primárias. Os atores também podem ser vistos como ativos ou passivos. Um ator ativo é aquele que inicia um *use case*, enquanto que um ator passivo nunca inicia um *use case*, mas participa de um ou mais deles.

4.5.1. Reconhecendo os Atores

Pela identificação dos atores, estabelecemos as entidades interessadas no uso e interação com o sistema. Após esta etapa é possível reconhecer a posição do ator, tentar descobrir seus requisitos no sistema e quais os *use cases* que ele necessita. Pode-se identificar os atores com as respostas as seguintes perguntas:

- Quem irá usar a principal funcionalidade do sistema, ou seja, quem serão os atores primários?
- Quem irá manter, administrar e fazer com que o sistema permaneça operando (atores secundários)?
- Quais dispositivos de hardware são necessários ao sistema?
- Com quais outros sistemas será necessário interagir?
- Quem ou o quê tem interesse nos resultados produzidos pelo sistema?
- Quais são interessados nos resultados que um sistema produz?

Pela observação de um sistema atual (implementado ou não), podemos identificar diferentes atores através da observação dos vários papéis que eles executam para realizar suas tarefas dentro do sistema.

4.5.2. Atores em UML

Os atores em UML são classes com o estereótipo <<ator>> e a classe deve possuir um nome que reflita o papel do ator. A classe ator pode ter atributos e comportamento, bem como uma documentação que descreva o ator. O ícone de estereótipo padrão para utilização em caso de uso é a figura humana (em inglês, “*stick man*”) com o nome do ator abaixo do ícone.

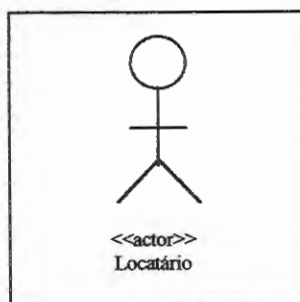


Figura 28: Representação do ator em *Use Case*.

4.5.3. Relacionamento entre os Atores

Como os atores são classes, eles podem ter os mesmos relacionamentos existentes entre as classes. Em diagramas *Use Case*, apenas os relacionamentos de generalização são usados para descrever um comportamento comum entre um número de atores.

Quando vários atores, como parte de seus papéis, também atuam com um papel mais genérico, temos uma generalização. Isto ocorre quando um comportamento comum é descrito como uma superclasse. Atores especializados herdam o comportamento da superclasse e, então estendem tal comportamento de alguma maneira. A generalização entre atores é descrita como uma linha com um triângulo ao seu final, em direção à superclasse. Esta notação é usada para demonstrar a generalização em quaisquer classes em UML.

4.6. Use Cases

Um *use case* representa uma funcionalidade completa na percepção do ator. A descrição formal de *use case*, segundo a UML, é “um conjunto de seqüências de ações que um sistema desempenha para produzir um resultado observável de valor a um ator específico”. As ações podem envolver a comunicação com os atores e também a realização de cálculos dentro do sistema. As características dos *use cases* são:

- Um *use case* é sempre iniciado a partir de um estímulo de um ator

Para iniciar um *use case* o ator envia uma mensagem ao *use case*, e este evento é chamado de estímulo.

- **Um *use case* fornece um valor ao ator.**

Um *use case* deve prover um valor tangível a um ator em resposta a sua solicitação. Estes valores podem ser explicitados na descrição textual de um *use case*. Por exemplo, quando o usuário utiliza o *use case* “Cadastro de Cliente” ele pode receber uma mensagem informando que o cliente foi cadastrado com sucesso ou uma mensagem de erro. Tais mensagens são os valores que o *use case* precisa levar ao ator.

- **Um *use case* é completo**

Um *use case* deve ser uma descrição completa, portanto não estará completo até que o valor final seja produzido mesmo se várias comunicações ocorrerem durante a interação.

Atores e *use cases* estão interconectados através de associações. Elas mostram com quais atores o *use case* se comunica, incluindo o ator que inicia a execução do *use case*. A associação é, normalmente um relacionamento um-para-um sem direção, isso significa que uma instância de ator comunica-se com uma instância de *use case* em ambas as direções.

Um *use case* possui um nome preferencialmente formado por uma frase e de acordo com a função que executa. Por exemplo, em um sistema de vídeo locadora: Emprestar Fita ao Cliente ou Solicitar Reserva de Fita.

Em UML um *use case* é representado por uma elipse (figura 29) contendo um nome que pode ser colocado dentro, acima ou abaixo do símbolo.

Um *use case* não é uma instância, é uma classe que descreve a funcionalidade do sistema como um todo, incluindo alternativas possíveis, erros e exceções que podem ocorrer durante a execução do *use case*. Uma instância de um *use case* se chama cenário, que representa um uso real do sistema. Por exemplo: O *use case* Solicitar Reserva de Fita poderia apresentar o seguinte cenário: “o cliente XX reserva uma fita de vídeo que não está disponível para empréstimo neste momento”.

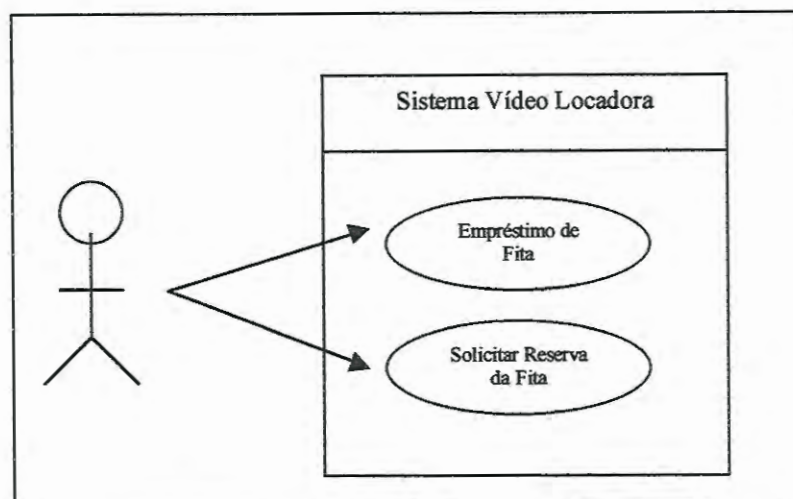


Figura 29: Representação de *use cases* (casos de uso)

Use Cases ocorrem quando um usuário realiza uma seqüência relacionada ao comportamento de transações em um diálogo com o sistema, portanto a arte está em identificar o que os usuários buscam cumprir em termos de atividades de negócio e não as funções que o sistema deve ter.

Um *use case* pode gerar vários cenários, ou seja, vários caminhos podem ser seguidos dependendo do contexto na execução do sistema. Estes caminhos são os possíveis cenários do *use case*.

Considera-se que o caminho básico para realizar um *use case* sem problemas e sem erros em nenhum passo da seqüência é denominado cenário primário. Neste tipo de cenário obtém-se com sucesso a execução dos passos para realizar a funcionalidade básica do *use case*. Por outro lado, caminhos alternativos bem como situações de erros podem ser representadas através de cenários secundários.

Cenários secundários descrevem seqüências alternativas e erros que podem ocorrer em um cenário primário associado a um *use case*. Cenários secundários podem ser descritos separadamente ou como extensão da descrição de um cenário primário. Cabe ressaltar que a visão de cenários pode variar de acordo com a abordagem utilizada.

A descrição em um cenário em diagrama *Use Case* deve conter:

- O objetivo do *use case*, ou seja, o que está se tentando fazer ou atingir.
- Como o *use case* é iniciado.

- O fluxo de mensagens entre atores e o *use case*, incluindo quais entidades são usadas.
- Fluxo alternativo em caso de erro ou exceção.
- Como o *use case* termina e o tipo de valor que ele retorna ao usuário.

4.6.1. Identificação de *Use Cases*

Após a definição dos atores, os *use cases* podem ser identificados fazendo-se as seguintes perguntas:

- Quais funções o ator necessita do sistema?
- O ator precisa ler, criar, destruir, modificar ou armazenar algum tipo de informação no sistema?
- O que o ator necessita fazer?
- O ator precisa ser notificado sobre eventos no sistema ou ainda notificar o sistema em si?
- É possível simplificar ou deixar mais eficiente o trabalho do ator através da inclusão de novas funções no sistema.

Existem questões na identificação dos *use cases* que não envolvem os atores, como:

- Quais são as entradas e as saídas, juntamente com sua origem e destino que o sistema requer?
- Quais são os principais problemas com a implementação atual do sistema?

4.6.2. Relacionamentos entre *Use Cases*

Existem três tipos de relacionamentos dentro de um diagrama *Use Case*:

- Comunicação: um ator comunica-se com um *use case*, assim, cada participação sua é mostrada conectando-se o símbolo de ator ao símbolo do *use case* por um caminho sólido.
- Relacionamento de Extensão: extensões são freqüentemente usadas para mostrar comportamento de exceção e casos especiais que aumentariam a quantidade de *use cases* no modelo. Trata-se de um relacionamento de um *use case* para outro, especificando como o comportamento definido para o primeiro caso pode ser inserido no comportamento definido para o segundo. É desenhada através de uma seta de generalização etiquetada com o estereótipo

<<estende>>, do *use case* que fornece a extensão para o *use case* básico. Um relacionamento de extensão de um *use case* A para o *use case* B indica que uma instância de B pode incluir o comportamento especificado por A

- Relacionamento de Uso: quando um número de *use cases* tem comportamento comum, esse comportamento pode ser modelado em um simples *use case* na qual é utilizado por outros casos. Um relacionamento de uso entre *use cases* ocorre quando há uma parcela de comportamento similar entre eles sugerindo uma reutilização em vez de nova cópia da descrição do comportamento. É desenhado como uma seta de generalização do *use case* que faz o uso ao *use case* que é utilizado, etiquetada com o estereótipo <<usa>>. Um relacionamento de uso de caso A para uso do caso B indica que uma instância do *use case* A também incluirá o comportamento como especificado por B.

A figura 30 apresenta as notações básicas utilizadas para descrever *Use Cases*.

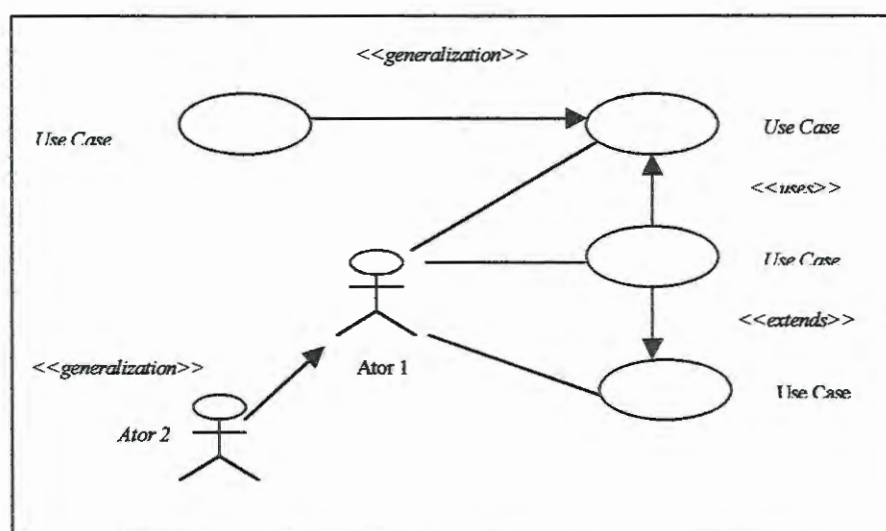


Figura 30 - Notações para *Use Cases* em UML

Fonte: SANTANDER & CASTRO (2000), p.167.

4.7. Construção do diagrama *Use Case*

O processo de construção de *Use Case* se inicia com a descoberta dos atores do sistema e prossegue com a descoberta do(s) *use cases* associados com estes atores. Para cada ator, são

encontrados todos os *use cases* relacionados ao mesmo. Isso ocorre porque cada ator requer do sistema algumas funcionalidades e os passos necessários para obtê-las são descritos em *use cases*.

O segundo passo consiste em definir o caminho básico (cenário primário) e, posteriormente os caminhos alternativos (cenários secundários) para cada um dos *Use Cases*.

O terceiro passo envolve revisar descrições de aspectos comportamentais de *Use Cases*, encontrando relacionamentos do tipo generalização e os estereótipos: *uses* e *extends*. Este processo é geralmente realizado adotando-se o princípio de desenvolvimento de software iterativo e incremental.

4.8. Problemas Relacionados ao *Use Case*

A importância do diagrama *Use Case* na análise orientada a objeto é destacado por diversos autores como PRADO (1999), JACOBSON *et al* (1999), LARMAN (1999), FOWLER & SCOTT (1997), como sendo de suma relevância para compreender o âmbito do problema a ser modelado.

Apesar do consenso e do reconhecimento de cenários como ferramenta importante no processo de Engenharia de Requisitos podemos apontar algumas carências da técnica principalmente no que diz respeito à inclusão de aspectos inerentes ao ambiente organizacional na qual o software está inserido.

De acordo com SUTCLIFFE *et al* (1998) e BUHR (1998), o diagrama *Use Case* é uma especificação de uma seqüência de eventos. Para os autores, cenários exigem muito trabalho para capturar e documentar todo o espaço do domínio, além disso, poucas recomendações concretas existem sobre como deveria ser a engenharia de requisitos baseada em cenários. Não existem passos ou diretrizes a seguir sobre como os problemas podem ser descobertos em cenários, deixando em aberto para o julgamento humano a forma de determinar os requisitos do sistema. Os requisitos são implícitos em modelos *Use Cases* e a falta de requisitos explícitos pode levar a falhas na validação. O desenvolvimento do diagrama *Use Case* não é tão simples como é apresentado na literatura, deve ser realizado em estágios que devem ser refinados e validados.

O diagrama de *Use Case* representa uma descrição de um completo curso de eventos iniciados por um usuário do futuro sistema e a interação entre usuário e o sistema. Não há



captura de requisitos nesse diagrama, além de dificultar a adaptabilidade. A fase inicial do projeto de orientação a objeto é o pior erro no diagrama.

O diagrama *Use Case* é tratado por muitos como forma de focar o modo como os usuários vêem a operação do sistema, mas na maioria das vezes o sistema ainda não existe e se existe ainda não é satisfatório. Além disso, a visão do usuário está relacionada com processos existentes, computadorizados ou não e a tarefa do desenvolvedor é buscar novas soluções e não continuar automatizando modos antiquados de operação.

Segundo MEYER (1997), o diagrama *Use Case* pode ser utilizado para validar aspectos finais de implementação, para ter certeza que o sistema inclui as rotinas dos típicos cenários usados, mas na tecnologia orientada a objeto ele não pode ser utilizado para captura de requisitos.

Para LEE *et al* (1998) e LILLY (2000), a engenharia de requisitos é uma tarefa crítica de qualidade de software. O diagrama *Use Case* é a técnica de eliciação mais utilizada nas aplicações industriais, devido sua notação simples e especificações em linguagem natural, mas tem como desvantagem o fato de descrever o comportamento parcial e, além disso, não existe uma abordagem sistemática para analisar o ambiente.

Para LILLY (2000), o diagrama *Use Case* tem uma notação muito trivial, não é suficiente para mostrar os limites do sistema. A má definição dos limites é provavelmente o problema mais universal em projetos iniciados como o diagrama *Use Case*. Quando o limite é confuso é difícil dizer o que está dentro ou o que está fora do sistema.

4.9. Considerações Finais

O diagrama *Use Case* é considerado como uma das abordagens de cenários que mais tem se destacado.

Este diagrama é parte integrante e central da UML, pois seu conteúdo é base do desenvolvimento dos outros diagramas presentes no desenvolvimento de software.

O diagrama *Use Case* é uma técnica utilizada para capturar os requisitos do sistema, segundo a visão do usuário.

Apesar do consenso e do reconhecimento como sendo uma técnica importante na captura de requisitos, pode-se apontar várias carências e dificuldades na sua utilização.

Desta forma, após ter-se levantado todos os aspectos positivos e negativos da técnica, no próximo capítulo pretende-se, através da técnica de Modelagem Organizacional – EKD (estudada no capítulo anterior) estabelecer uma correlação entre os elementos presentes no EKD e os elementos componentes em *Use Cases*, com o propósito de se estabelecer diretrizes que possam ser úteis no desenvolvimento de diagramas *Use Cases*.

5. Contribuição da Técnica EKD no Desenvolvimento de Diagramas *Use Case*

5.1 - Considerações Iniciais

O desenvolvimento de novas técnicas de suporte as atividades de Engenharia de Requisitos tem sido uma preocupação atual da comunidade acadêmica e industrial. Diversas abordagens para apoiar as atividades de Engenharia de Requisitos têm surgido desde as primeiras técnicas utilizadas com esse fim. Dentre estas abordagens, o diagrama *Use Case* tem-se destacado. Esta técnica é utilizada para capturar os requisitos do sistema segundo a visão do usuário.

Contudo, o diagrama *Use Case* não têm distinguido a fase inicial (captura dos requisitos organizacionais) e a fase final da Engenharia de Requisitos.

A fase inicial tem o objetivo de definir como o sistema pretendido irá satisfazer os objetivos da organização, porque ele é necessário, quais as alternativas existentes e quais as implicações das alternativas para as várias partes envolvidas, enquanto a fase final da Engenharia de Requisitos foca na perfeição e consistência do sistema de informação.

As atividades realizadas na fase inicial da Engenharia de Requisitos tem objetivos e pressuposições que são diferentes da fase final. Dessa forma, torna-se necessário o fornecimento de diferente suporte de modelagem e raciocínio para estas fases.

5.2. Introdução

A declaração dos requisitos iniciais em diagramas *Use Case* inicia-se com uma descrição informal do que é esperado que o sistema faça, segundo as intenções e desejos dos usuários.

Muitos problemas associados ao desenvolvimento de software pode-se iniciar nesta fase, pois quando os usuários desejam desenvolver um software, na maioria dos casos o que se tem são intenções de facilitar a execução de atividades no ambiente organizacional.

De acordo com YU (1997), muitas das pesquisas de Engenharia de Requisitos têm tomado como ponto de partida a declaração dos requisitos iniciais que expressam os desejos do cliente sobre o que o sistema deverá fazer. Requisitos iniciais são frequentemente ambíguos, incompletos, inconsistentes e, normalmente expressos informalmente.

Neste contexto, muitos *frameworks* tem sido propostos com o intuito de representar o conhecimento e apoiar o raciocínio na Engenharia de Requisitos, tais como DARDENE *et al.* (1993), DUBOIS *et al.* (1986), FICKAS & HELM (1992), GREENSPAN *et al.* (1982), MYLOPOULOS *et al.* (1992) e YU (1995).

Porém BUBENKO (1993), faz uma observação importante em relação ao grande número de *frameworks* propostos.

Para o autor, a maioria dos *frameworks* são direcionados para a metade ou final do processo de desenvolvimento de software, os quais focam somente na perfeição e consistência do sistema de informação.

Praticamente, nenhum deles direciona de forma estruturada o início do processo, os objetivos do negócio, os estágios de geração de requisitos e o problema de mover do informal para o domínio formal.

Os métodos existentes não são designados para captura explícita e para representação de forma estruturada do conhecimento organizacional e do negócio para ser subsequente utilizado no projeto do sistema de informação. Não são mantidas ligações entre o modelo organizacional e a especificação do sistema.

Para BUBENKO (1998), é necessária estabelecer esta ligação, pois o alto custo de manutenção é um fator importante para incentivar o entendimento do negócio e derivar os requisitos desse sistema. Assim, é fundamental que os Engenheiros de Requisitos direcionem os sistemas de informação aos objetivos organizacionais.

A importância do diagrama *Use Case* na análise orientada a objeto é destacado por diversos autores como PRADO (1999), JACOBSON *et al.* (1999), LARMAN (1999), FOWLER & SCOTT (1997), como sendo de suma relevância para compreender o âmbito do problema.

Porém, apesar do consenso e do reconhecimento do diagrama *Use Case* como uma ferramenta importante no processo de Engenharia de Requisitos, pode-se apontar algumas carências da técnica no que diz respeito às atividades da fase inicial de Engenharia de Requisitos.

Segundo MEYER (1997), o diagrama *Use Case* pode ser utilizado para validar aspectos finais de implementação, mas não pode ser utilizado para captura dos requisitos inerentes ao ambiente organizacional (fase inicial).

As atividades realizadas nesta fase são feitas informalmente e sem muito suporte de uma ferramenta adequada e especializada.

À medida que a complexidade do domínio do problema aumenta é evidente que o apoio de uma ferramenta será necessária para ajudar os desenvolvedores a compreender o que os usuários realmente necessitam, e estes a compreender o que os sistemas técnicos podem fazer por eles. É comum encontrar situações, onde muitos sistemas não satisfazem as necessidades do negócio embora estejam tecnicamente corretos.

Sabendo-se que as atividades da fase inicial de Engenharia de Requisitos têm objetivos e pressuposições que são diferentes da fase final, torna-se apropriado o fornecimento de diferente suporte de modelagem para as duas fases.

Após ter revisto alguns dos potenciais problemas referentes ao diagrama *Use Case*, pretende-se nos itens posteriores mostrar a ligação entre a especificação de requisitos e o Modelo Organizacional e estabelecer uma correlação entre os submodelos do EKD e os elementos componentes em *Use Case* propondo desta forma, algumas diretrizes que poderão ser úteis no desenvolvimento de diagramas *Use Case*. Assim fatores do ambiente organizacional inicialmente não considerados em *Use Case* poderão ser integrados ao mesmo.

Este trabalho concentra-se na técnica EKD, proposta por BUBENKO. Para os leitores interessados, no Workshop de Engenharia de Requisitos (WER-2000) foi apresentado um trabalho de autoria de SANTANDER & CASTRO com a técnica *i**, a qual serviu como base para compor algumas diretrizes do presente trabalho.

5.3. A Ligação entre a Especificação de Requisitos e o Modelo Organizacional

Entende-se por requisito como uma condição da qual o usuário precisa para que seja solucionado um problema ou alcançado um objetivo.

Quando se inicia o desenvolvimento de software diretamente a partir do mapeamento de sistemas, os analistas conseguem encontrar soluções técnicas, mas que na maioria das vezes são

baseados em artifícios puramente relativos a tecnologia que podem não atender ao negócio de forma plena ou ainda de forma incorreta.

Para desenvolver sistemas que satisfarão verdadeiramente as reais necessidades da organização, freqüentemente, necessita-se ter uma compreensão mais aprofundada de como o sistema está embutido no ambiente organizacional (YU, 1997).

As organizações, segundo as técnicas de modelagem organizacional, são representadas a partir da elaboração de modelos que facilitam chegar as especificações de requisitos funcionais e não funcionais bem próximos dos requisitos reais da organização.

Boa parte dos problemas dos sistemas de informação são originados pela não realização deste estudo na fase inicial de Engenharia de Requisitos.

Para tanto, BUBENKO (1993), cita os itens que devem estar em uma especificação de requisitos e a questões a qual o processo de modelagem organizacional devem trazer respostas:

- 1- Por que o sistema é necessário? Quais são as metas do ambiente da aplicação? Quais são as prioridades das metas? Como as metas relacionam-se?
- 2- Quais conceitos ou temas da organização, incluindo seus relacionamentos com objetivos, atividades, processos e atores. Como são definidos? Quais regras do negócio e restrições monitoram esses objetivos?
- 3- Quais atividades do negócio e processos existem ou deveriam existir para gerenciar a organização de acordo com os objetivos? Como são realizados os processos de negócio? Quais informações são necessárias?
- 4- Quem está realizando quais tarefas e processos?
- 5- Que requisitos são gerados por esses processos?

O Modelo Organizacional deve responder as seguintes questões:

- 1- Por que o sistema é construído, qual a sua justificativa?
- 2- Quais são os processos de negócio e quais são apoiados pelo sistema?
- 3- Quais são os atores da organização que realizam esse processo?
- 4- Quais conceitos eles estão processando ou falando sobre? Quais são as informações necessárias?
- 5- Quais objetivos iniciais e quais requisitos podem ser declarados considerando o sistema de informação para ser desenvolvido.

A ênfase está na compreensão dos porquês do requisito do sistema, ao invés de uma especificação detalhada e precisa “do quê o sistema deveria fazer”. Obter respostas para estes “porquês” é importante não somente para ajudar a desenvolver sistemas com sucesso em primeira instância, mas também facilitar o desenvolvimento de cooperação com outros sistemas com agentes humanos e reprojeter processos de negócios.

5.4. Desenvolvimento do Diagrama *Use Case* a Partir dos Modelos Existentes na Técnica EKD

Na Modelagem Organizacional é comum desenvolver-se várias visões com o propósito de representar os objetivos, as intenções, os atores, os por quês e os possíveis relacionamentos para um determinado processo.

O processo de construção do diagrama *Use Case* inicia-se com a descoberta dos atores do sistema e prossegue com a descoberta do(s) *use cases* associados com estes atores. Para cada ator, são encontrados todos os *use cases* relacionados ao mesmo.

Este processo pode ser auxiliado e facilitado efetuando-se a análise a partir dos modelos existentes na técnica EKD.

O que foi elaborado pelo Modelo de Objetivos, Modelo de Regras do Negócio, Modelo de Conceitos, Modelo de Processos do Negócio e Modelos de Atores e Recursos é uma descrição inicial dos objetivos da organização, das regras do negócio, dos processos, do sistema de atores e das entidades de informação, de modo que os processos de negócio devam contribuir com os objetivos declarados.

A técnica EKD inclui um modelo denominado Modelo de Requisitos e Componentes Técnicos para tratar os requisito técnicos do sistema de informação, este modelo define as possíveis estruturas e propriedades do sistema de informação com o objetivo de apoiar as atividades do negócio definidos no Modelo de Processos do Negócio.

Inicialmente é necessário definir um conjunto de objetivos e requisitos específicos sendo que é fundamental que esses objetivos e requisitos sejam derivados e consistentes com os outros submodelos de EKD.

Segundo BUBENKO *et al* (2001), os componentes presentes no MRCT são: objetivos do sistema de informação; problemas do sistema de informação e requisitos do sistema de informação.

Os objetivos do sistema de informação são usados para expressar o alto nível de objetivos em relação ao sistema de informação. Esses objetivos são motivados principalmente pelas atividades do Modelo de Processos do Negócio.

Os problemas do sistema de informação expressam estados não desejáveis do negócio em relação ao sistema de informação a ser desenvolvido.

Os requisitos do sistema de informação expressam requisitos a serem designados para propriedades funcionais ou não funcionais do sistema de informação. Isto é importante, pois sabe-se que os requisitos funcionais não existem isolados, eles estão relacionados aos requisitos não funcionais

Porém uma expressão de requisitos sempre refere-se a componentes do Modelo de Processos do Negócio e aos componentes do Modelo de Atores e Recursos.

No Modelo de Atores e Recursos identifica-se quais atores e recursos estão envolvidos nas atividades organizacionais, permitindo dessa forma, a inclusão como parte da Engenharia de Requisitos a descrição de um sistema sócio-técnico a ser desenvolvido e no Modelo de Processo do Negócio define-se quais atividades são reconhecidas na organização e a forma pela qual elas interagem e manuseiam materiais e informações.

Dessa forma para o Modelo de Requisitos e Componentes Técnicos ser utilizado como fonte de informação para o desenvolvimento de diagramas *Use Case*, torna-se necessário analisar todas as ligações entre os modelos presentes na técnica EKD. As ligações entre modelos tornam o conhecimento mais disponível, sendo possível ver por que certos processos e requisitos do sistema de informação têm que serem introduzidos.

Nas seções seguintes serão descritas, respectivamente as diretrizes para descoberta de atores bem como os *use cases* para cada ator.

5.4.1. Diretrizes para Descoberta de Atores

Como visto anteriormente, o processo de construção do diagrama *Use Case* inicia com a descoberta dos atores do sistema a ser desenvolvido. Para tanto, é fundamental observar o Modelo de Atores e Recursos.

Desta forma, para iniciarmos o processo de construção do diagrama *Use Case* é necessário analisar as características associadas com os atores nessas duas técnicas.

Ator em diagrama *Use Case*

- Objetiva-se com o diagrama *Use Case* descrever as ações de um ator interagindo com o sistema.
- Como ator em diagrama *Use Case* pode-se incluir pessoas (usuários), hardware ou outros sistemas que interagem com o sistema a ser modelado.
- Um ator desempenha um papel específico em relação ao sistema. Por ex: se uma pessoa quer emprestar um livro da biblioteca, é o seu papel como cliente que será modelado e não a própria pessoa.
- Atores são sempre partes externas ao sistema. Eles nunca são partes componentes (internas) do sistema.

Ator na técnica de Modelagem Organizacional – EKD

- Objetiva-se através do Modelo de Atores e Recursos (MAR) definir quais atores estão envolvidos nas atividades organizacionais.
- Como atores em EKD pode-se incluir pessoas, unidades organizacionais (grupo, departamento e divisão), recursos não-humanos (hardware, sistemas, equipamentos) e tarefas (ator, supervisor, gerente).
- Atores individuais (ex: Pedro Pádua e Fabiana Miranda) podem ser modelados no sistema.
- Os atores podem representar partes componentes do sistema ou o próprio sistema computacional.

Analisado as principais diferenças existentes na conceitualização de ator na técnica EKD e no diagrama *Use Case*, torna-se necessário estabelecer diretrizes com o intento de mapear os atores em EKD para atores em diagramas *Use Case*.

A seguir serão apresentadas algumas diretrizes que permitem apontar possíveis candidatos a atores em diagrama *Use Case* a partir de atores existentes na técnica EKD.

1 - Parte-se do pressuposto de que todo ator, existente no Modelo de Atores e Recursos da técnica EKD deve ser analisado como um possível candidato como ator em diagrama *Use Case*.

Estas análises incluem: fazer um estudo preliminar das ligações possíveis entre os Modelos de Atores e Recursos com os outros modelos presentes na técnica EKD.

O estudo destas ligações é fundamental, pois torna possível ver por que certos atores são necessários no Modelo Organizacional.

- **Ligação entre o Modelo de Objetivos e o Modelo de Atores e Recursos:**

Esta ligação pode ter vários significados, ou seja, podem motivar ou requerer a introdução de novos atores, ou elas podem descrever quais atores do MAR são responsáveis por alcançar um particular objetivo.

Por ex: o ator atendente de uma biblioteca é responsável em “manter o catálogo da biblioteca regularmente atualizado”.

- **Ligação entre Modelo de Processos de Negócio e o Modelo de Atores e Recursos:**

Esta ligação tem o objetivo de descrever como os componentes do MAR, tais como: pessoas, unidades organizacionais, recursos não humanos e tarefas são envolvidos nos processos do Modelo de Processos de Negócio.

Por ex: o ator atendente de uma biblioteca é responsável em executar o processo “registrar cliente”.

- **Ligação entre Modelo de Atores e Recursos e o Modelo de Regras do Negócio:**

Esta ligação tem o objetivo de descrever como os componentes do MAR são relacionados a regras do negócio do Modelo de Regras do Negócio.

Por ex: o ator atendente de uma biblioteca é responsável por “atualizar o catálogo da biblioteca logo que as mudanças ocorrerem”.

2 – Atores em diagramas *Use Case* são sempre externos, portanto é necessário verificar se o ator escolhido é externo ao sistema. Este aspecto é fundamental ser observado, pois na técnica EKD permite-se incluir atores representando partes do sistema ou até mesmo uma abstração de software como um todo o que no diagrama *Use Case* não caracteriza um ator.

Pode-se dizer que o ator é externo, quando o mesmo interage com o sistema a ser modelado, fornecendo e atendendo informações importantes para este.

Um adendo a fazer-se nesta etapa é que a análise do ator deve ser realizada observando-se o objetivo de se desenvolver um sistema computacional. Por exemplo: o ator cliente de uma biblioteca deseja realizar um empréstimo de um livro. O ator atendente pede ao ator cliente que

apresente seu documento de identidade (número) e seu cartão da biblioteca que contém uma etiqueta com o código do cliente. Este processo de empréstimo não considera a existência de um sistema computacional. Porém a análise deve ser realizada observando-se a intenção do desenvolvimento de um sistema computacional para solucionar o problema referente ao empréstimo. Dessa forma, o ator cliente e o ator atendente podem ser considerados externos ao sistema, pois os mesmos interagirão com o sistema a ser desenvolvido fornecendo informações importantes para este, tais como: o número do documento de identidade do cliente e o seu código de usuário. Assim, o ator atendente e o ator cliente são candidatos a atores em *Use Case*.

3 – Se o ator avaliado é considerado externo ao sistema, conforme a etapa anterior, torna-se necessário realizar algumas análises dos mesmos, visando garantir seu mapeamento para ator no diagrama *Use Case*.

Esta etapa é necessária devido à visão ampla que a técnica de Modelagem Organizacional EKD proporciona, o que pode implicar na existência de atores não relevantes ao contexto do sistema. As análises nesta etapa incluem:

Analisar os tipos de relacionamentos permitidos entre atores do MAR com os outros modelos e certificar-se de que estes relacionamentos são:

- **Responsabilidade Operacional / Responsabilidade Organizacional:** é o relacionamento entre atores e processos de negócio, regras do negócio e objetivos. A responsabilidade organizacional dá a liberdade do ator tomar decisões por outras entidades da organização tais como: objetivo, regras, processos entre outros. Por ex: por quais objetivos e processos o ator é responsável. A responsabilidade operacional objetiva-se a indicar qual ator é alocado para realização de um processo de negócio ou que o processo de negócio é fixado por um determinado ator. Por ex: quais processos o ator executa.

Por ex: O ator atendente da biblioteca irá manter o sistema de controle da biblioteca (responsabilidade organizacional), adicionando e deletando obras, itens, clientes e também mantendo processos tais como empréstimo e retorno de itens, reservas de livros e fornecendo informações sobre as possíveis datas de reserva do sistema (responsabilidade operacional).

- **Dependência Operacional / Dependência Autoridade:** é o relacionamento entre os atores da organização. Um ator depende do outro para alguma coisa que pode ser um recurso (físico ou informacional) ou uma saída de um processo de negócio. A dependência operacional diz

respeito a dependências criadas no fluxo de trabalho. A dependência de autoridade é relacionada com a dependência criada por regulamentos regras organizacionais, etc.

Por ex: O ator atendente depende das informações (recurso informacional) do ator cliente e este ator, depende da disponibilidade do livro (recurso físico) para a realização do empréstimo (dependência operacional). O ator atendente necessita de senha para trabalhar no sistema computacional da biblioteca (dependência de autoridade).

Analisando o ator atendente e o ator cliente verifica-se, que as dependências associadas a estes, caracterizam-nos como fundamentais ao contexto do sistema, pois estes fornecem informações que serão tratadas em um sistema computacional, visando o problema de empréstimo de livro. Portanto, estes atores possuem características que os tornam candidatos a atores no diagrama *Use Case*.

4 – Em relacionamentos do tipo ISA, na qual, são usados para descreverem relacionamentos de generalização/especialização entre tarefas (papéis) do MAR, tais como “ator A” ISA “ator B” ambos os atores devem ser avaliados conforme diretrizes anteriores.

Normalmente ambos são considerados atores em *Use Case*, pois esse tipo de relacionamento também é permitido e modelado em diagramas *Use Case*.

Por ex: No Modelo de Atores e Recursos o relacionamento entre “cliente” ISA “mau cliente”. Este relacionamento declara que componentes que executam a tarefa B também executam a tarefa A, as propriedades e relacionamentos são herdados por B. Isso significa que se A está operando processo P, então B também está operando processo P.

Conforme diretrizes aplicadas anteriormente na avaliação do ator cliente, o mesmo pode ser considerado como ator em diagrama *Use Case*.

Da mesma forma, aplicando-se as mesmas diretrizes ao ator mau cliente pode-se apontar o mesmo como ator em diagrama *Use Case*.

O relacionamento entre ambos deve ser do tipo << generalization >>. O ator mau cliente deve ser anotado como uma especialização do ator cliente no diagrama *Use Case*.

5 - Se ao ator analisado assumir “somente” o relacionamento de dependência operacional no Modelo Organizacional verificar-se as dependências em relação ao mesmo não estão relacionadas à dependência de atores do sistema em relação ao próprio software, pois o ator talvez seja o software ou parte componente do mesmo, o que caracteriza um ator na técnica EKD, porém não se enquadrando como ator em *Use Case*.

Por ex: o programa de controle da biblioteca, pois este ator representa o próprio sistema computacional a ser desenvolvido, o que fere o exposto na conceitualização de ator em diagrama *Use Case*.

As diretrizes descritas acima permitem a elaboração de uma lista de candidato a atores no desenvolvimento do diagrama *Use Case*. Após este processo, a lista de atores é estabelecida.

Segundo SANTANDER & CASTRO (2000), é importante ressaltar que através da aplicação das diretrizes para o estabelecimento dos candidatos a atores em *Use Case* o engenheiro de requisitos pode iniciar o desenvolvimento do diagrama *Use Case* já familiarizado com intenções e objetivos organizacionais e estratégicos, dessa forma o desenvolvimento de um Modelo Organizacional no processo de Engenharia de Requisitos possibilita uma visão bastante útil no desenvolvimento de *Use Case*.

5.4.2 Diretrizes para Descoberta de Use Cases

Dando continuidade ao processo de desenvolvimento do Diagrama *Use Case* a partir da Modelagem Organizacional torna-se fundamental uma análise detalhada do Modelo de Requisitos e Componente Técnicos e das ligações entre modelos da técnica EKD, para encontrar possíveis candidatos a *use cases* para cada ator.

A realização desta análise permite-nos a possibilidade de apontar os *use cases* para os atores bem como os cenários de cada *use case*

Através do Modelo de Requisitos e Componentes Técnicos podemos estabelecer os objetivos, os problemas do sistema de informação e os requisitos funcionais e não funcionais.

Os objetivos e os requisitos funcionais estão diretamente associados aos *use cases*, enquanto os requisito não funcionais estão associados indiretamente, porém é importante estabelecer uma correlação entre *use cases* e requisitos não funcionais, pois, sabe-se que os requisitos funcionais não existem isolados, eles estão relacionados aos requisitos funcionais. Desta forma, segundo SANTANDER & CASTRO (2000), o engenheiro de requisitos estará desde do início, informado das restrições que devem ser consideradas durante o processo de desenvolvimento.

Segundo os autores, esta tarefa de observar os modelos organizacionais e extrair informações para diagramas *Use Case* não é trivial, pois ambas as técnicas permitem focar requisitos de software sob perspectivas diferentes.

Como os modelos organizacionais podem representar alternativas para a obtenção de objetivos organizacionais, uma das vantagens deste mapeamento é que o Engenheiro de Requisitos, além de estar envolvido conscientemente com os objetivos organizacionais, pode também fazer, juntamente com o cliente, opções a serem descritas em *use cases* que atinjam da melhor maneira possível as metas e objetivos da organização e/ou cliente. Assim, aspectos antes não considerados na descoberta de *use cases* podem ser avaliados, analisados e incluídos na descrição do mesmo.

A seguir serão apresentadas algumas diretrizes de forma a auxiliar no trabalho de descoberta de *use cases* para cada ator a partir da modelagem organizacional

1- Inicialmente é necessário analisar o Modelo de Requisitos e Componentes Técnicos e desenvolver um conjunto de objetivos e requisitos para o sistema de informação, sendo fundamental que esses objetivos e requisitos sejam derivados e consistentes com os outros modelos da técnica EKD.

Por ex: É objetivo do sistema de informação: “Manter todos os tipos de informações dentro da biblioteca sobre recursos dos livros, transações, empréstimos dos clientes, pedidos e listas de esperas dos clientes”.

É requisito do sistema de informação: “Biblioteca deveria usar software sempre que for possível”, “Prover 24h de catálogo de pesquisa” e “Catálogo de pesquisa deveria ser exportável em CD-ROM”.

2- Como os objetivos e requisitos do sistema de informação devem ser consistentes com os outros modelos da técnica EKD, é fundamental analisar as principais ligações entre os modelos que serão apoiados pelo MRCT. As análises destas ligações podem levar a descoberta de *use cases* ou a passos de um *use case* para cada ator analisado no item 5.4.1.

- **Ligação entre o Modelo de Objetivos e o Modelo de Processos do Negócio**

Os componentes do Modelo de Processos do Negócio são motivados pelos componentes do Modelo de Objetivos, possibilitando que os objetivos do Modelo de Objetivos sejam alcançados.

Esta ligação tipicamente relaciona objetivos do Modelo de Objetivo a processos do Modelo de Processos do Negócio, através do relacionamento “motiva”.

Por ex: O objetivo “Minimizar lista de espera”, motiva o processo “Gerenciamento de retornos de empréstimo” o qual pode ser decomposto em “registro de retorno” e “checar se o livro está reservado”.

Quando um processo de negócio necessita ser decomposto, devido sua complexidade, e que normalmente inclua outros passos para leva-lo a sua execução, este poderá ser considerado como um *use case*. Se o processo for bastante simples, geralmente este poderá ser agregado aos demais processos com o mesmo fim em comum, sendo então este conjunto de processos também considerado um *use case*.

Desta forma, o processo “gerenciamento de retorno”, poderá ser candidato a *use case* do ator atendente, sendo que o nome para o *use case* deve representar a melhor idéia de junção dos processos.

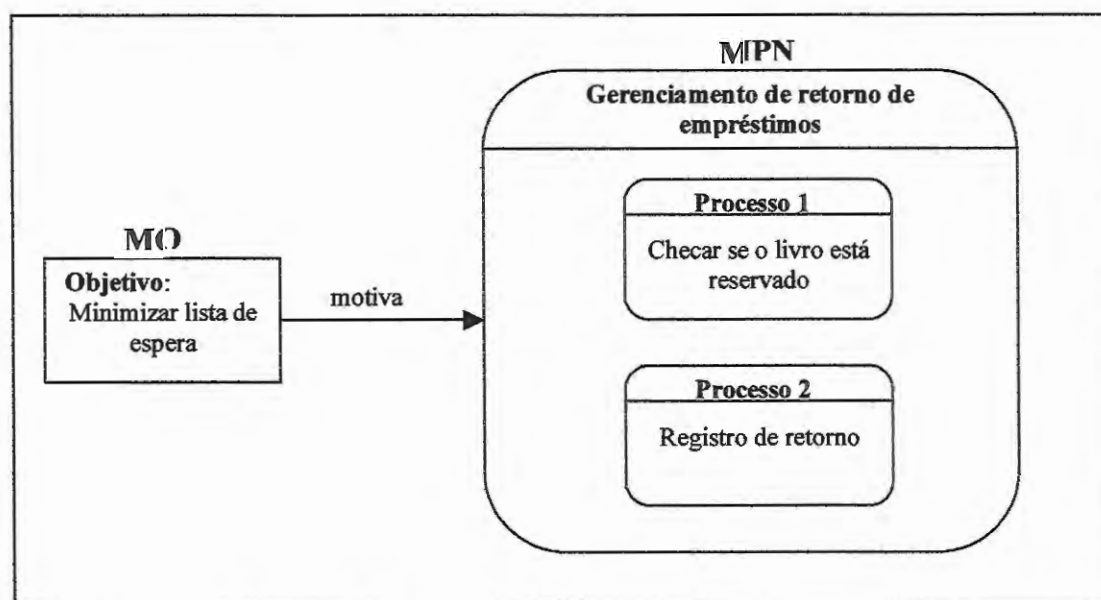


Figura 31: Ligação entre o Modelo de Objetivos e o Modelo de Processos do Negócio

- **Ligação entre o Modelo de Regras do Negócio e o Modelo de Processos do Negócio**

Através desta ligação é possível determinar quais são os processos apoiados e/ou disparados pelas regras do negócio do Modelo de Regras do Negócio e, por conseguinte determinar quais são os requisitos do sistema de informação gerados pela ligação destes modelos.

Por ex: A regra “Clientes tornam se maus-clientes se demorarem mais do que quatro semanas para entregarem o livro”, é apoiado pelo processo “Checar por demora na entrega” e dispara o processo “Relatar mau-cliente”, a qual motiva os seguintes requisitos do sistema de informação:

- Registrar maus clientes
- Gerar relatórios de livros atrasados

Desta forma, ambos requisitos podem ser considerados como *use cases* para o ator atendente.

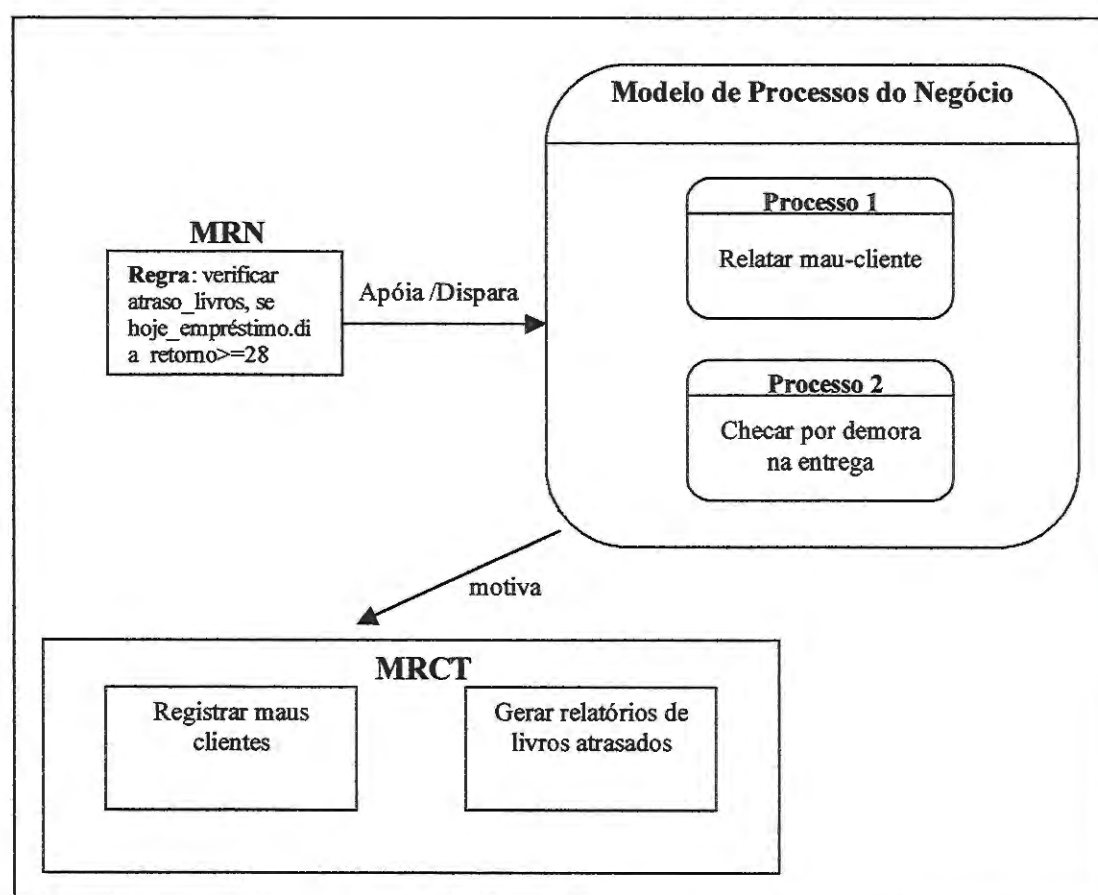


Figura 32: Ligação entre o Modelo de Regras do Negócio e o Modelo de Processos do Negócio

- **Ligação entre o Modelo de Processos do Negócio e o Modelo de Atores e Recursos**

Analisando o Modelo de Atores e Recursos podemos identificar como diferentes atores são relacionados entre si e como são relacionados com os componentes do Modelo de Objetivos, e de que forma os objetivos são relacionados aos processos do Modelo de Processos do Negócio.

Ligações entre os componentes do Modelo de Processos do Negócio e o Modelo de Atores e Recursos descrevem o tipo de relacionamento que existe entre um particular processo de negócio e um ator, tais como:

- Ator realiza processo P;
- Ator é responsável pelo processo P;
- Ator apóia processo P.

Porém, para a realização do mapeamento dos processos do Modelo de Processos do Negócio para *use cases*, é necessário analisar os possíveis relacionamentos de responsabilidade e de dependência, referentes ao ator em questão, isto é necessário devido a visão ampla que a Modelagem Organizacional permite, podendo haver processos que não são necessários para apoiar o sistema de informação a ser desenvolvido. Estas análises incluem, certificar-se de que os relacionamentos são:

- **Responsabilidade Operacional / Responsabilidade Organizacional**

Em grande parte dos casos, esta responsabilidade pode ser considerada como um *use case* do ator.

De acordo com a figura 33, o ator atendente tem o papel de fornecer serviço para o ator cliente e também é responsável pelo objetivo “manter o catálogo da biblioteca regularmente atualizado”, a qual dispara os processos “Registrar livro”, “Registrar cliente” e “Registrar transação”. A junção desses processos pode resultar em um processo maior denominado “Prover empréstimo”.

Desta forma, como dito anteriormente, um processo que claramente inclua outros passos para levá-lo a sua execução, pode ser considerado um *use case*.

Assim, o processo “Prover empréstimo” pode ser considerado um *use case* para o ator atendente.

- **Dependência Operacional**

Igual ao relacionamento de responsabilidade, o relacionamento de dependência em grande parte dos casos pode ser considerado como um *use case* do ator. Avaliando novamente a figura 33, para que o ator atendente realize o processo “prover empréstimo” é necessário que o ator cliente forneça informações sobre os seus dados cadastrais, tais como nome, identidade e código da biblioteca. Analisando esta dependência temos que o ator atendente é dependente das informações do ator cliente. Desta forma “dados cadastrais” podem ser considerados como um

use case do ator cliente, pois o mesmo inclui uma série de informações para a execução do processo “prover empréstimo”.

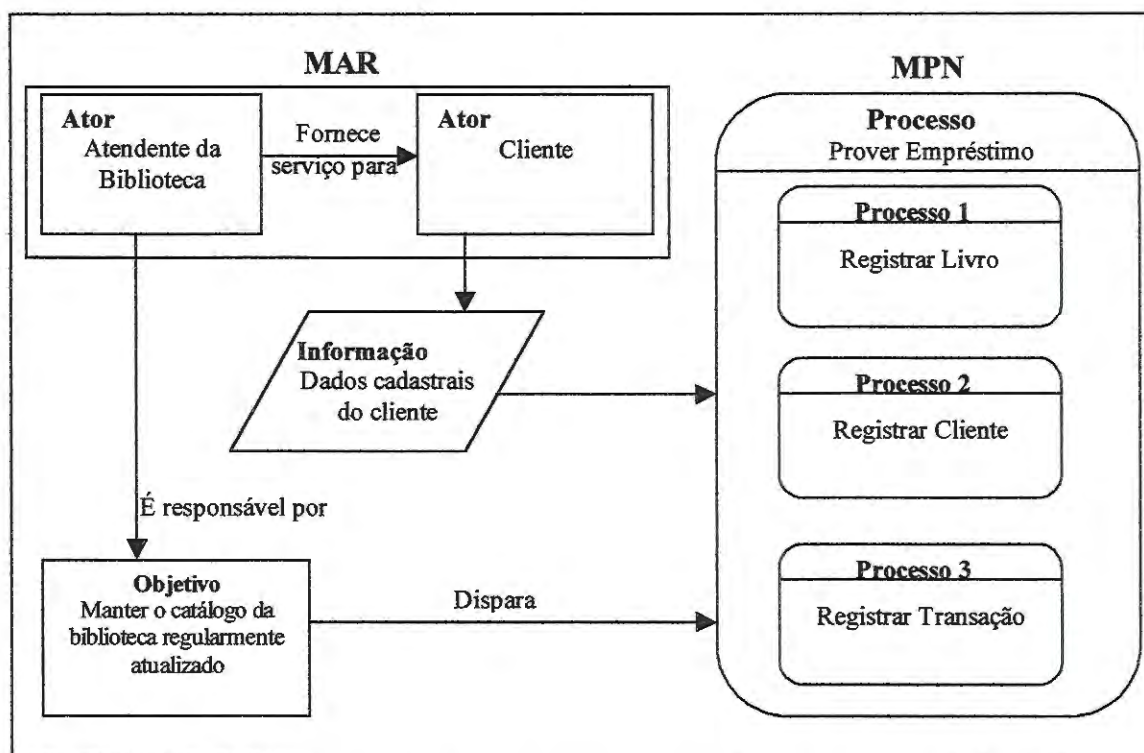


Figura 33: Ligação entre o Modelo de Processos do Negócio e o Modelo de Atores e Recursos

5.5. Aplicação das Diretrizes

Assume-se que o objetivo seja encontrar os *use cases* para um sistema computacional de controle da biblioteca.

Na Modelagem Organizacional é comum desenvolverem-se várias visões de forma a representar melhor as intenções e motivações para um determinado processo. Diante disso a análise para a descoberta de atores e *use cases* deve considerar todos os modelos organizacionais produzidos.

Desta forma, seguindo as diretrizes definidas na seção 5.4.1., a lista de candidatos a atores em diagrama *Use case* inclui: atendente, cliente, mau-cliente. Analisando os candidatos verifica-se que mau-cliente é um tipo cliente, sendo que as funcionalidades associadas com este

ator são quase todas cobertas pelo ator cliente. Então deve-se considerar este ator uma especialização de cliente e estabelecer este tipo de relacionamento, entre os mesmos, utilizando a notação padrão em diagrama *Use Case*.

Continuando o processo de construção do diagrama *Use Case* a partir da técnica EKD, tem-se que o próximo passo é descobrir e relacionar os *use cases* para cada ator.

Assim, seguindo as diretrizes, pode-se constatar:

- **Para o ator cliente**

Analisando a dependência do ator atendente em relação ao ator cliente para a realização do processo “prover empréstimo” pode-se visualizar que para a realização do empréstimo é necessário que o ator cliente forneça seus dados cadastrais. Desta forma, este relacionamento de dependência consiste no *use case* “entrar dados cadastrais”, representando que o ator cliente necessita informar seus dados cadastrais para o sistema.

- **Para o ator atendente**

Para encontrar os candidatos a *Use Case* para o ator atendente, adotou-se a análise das ligações entre os modelos e o relacionamento entre atores. Desta forma os seguintes *use cases* foram descobertos:

- Prover empréstimo
- Gerenciamento de retorno de empréstimos
- Registrar maus clientes
- Gerar relatórios sobre os empréstimos de livros

Diante das considerações acima, tem-se na figura 34 um exemplo da representação dos atores juntamente com seus relacionamentos com os *use cases* descobertos, a partir da técnica EKD.

É lógico que o diagrama *Use Case* da figura 34 ainda pode ser alterado e complementado, pois as diretrizes para compor este diagrama foram baseadas em um simples exemplo do caso da biblioteca citado por BUBENKO.

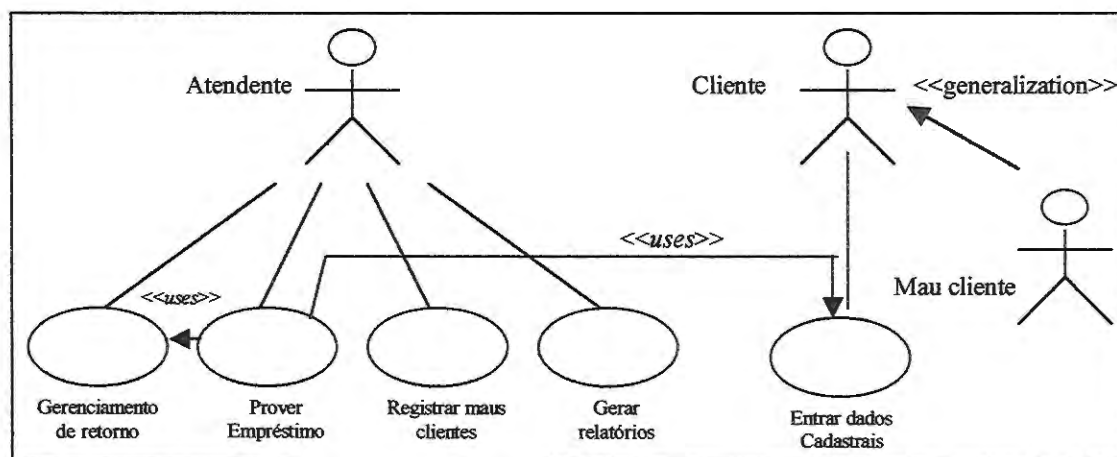


Figura 34- Diagrama *Use Case* para um sistema de biblioteca

Tendo descoberto os atores e os *use cases*, torna-se necessário definir os cenários, bem como os relacionamentos de generalização e os estereótipos do tipo `<<uses>>` e `<<extends>>` relacionados ao mesmo.

Por ex: o *use case* “prover empréstimo” representa o uso do sistema pelo ator atendente para realizar o processo “empréstimo”. Este *use case* conteria todos os passos necessários para realização do empréstimo, tendo início quando o ator fornece ao sistema, informações do livro e do cliente.

Outras tarefas que o ator espera que o sistema realize para que o *use case* seja efetuado também devem ser descritos. O sistema deve, com base nos dados fornecidos pelo ator atendente verificar a disponibilidade do livro e verificar se o cliente é registrado no sistema. Caso o livro se encontre disponível na biblioteca e o cliente esteja devidamente registrado, o sistema deverá registrar a transação.

Desta forma, para o *use case* “prover empréstimo” poder-se-ia ter o cenário com os seguintes passos:

Use case: Prover Empréstimo

Passo 1: O *use case* inicia com o ator atendente fornecendo ao sistema informações sobre o livro (o *use case* “Gerenciamento de retorno de empréstimos” é incluído `<<uses>>` neste passo, pois considera que este processo de informar ao sistema dados sobre o empréstimo é composto de vários passos, os quais estão contidos neste *use case*).

Passo 2: O sistema deve solicitar do ator cliente os dados cadastrais (tendo como informação o nome do cliente, identidade e código (o *use case* dados cadastrais é incluído <<uses>> neste passo, porque representa os passos necessários para a obtenção das informações referente ao cliente para realização do processo).

Passo 3: De acordo com os passos acima o sistema realiza o *use case* “prover empréstimo”, e retorna ao ator atendente a seguinte mensagem “empréstimo realizado com sucesso”.

5.6. Considerações Finais

O diagrama *Use Case* é considerado como uma das técnicas que mais tem se destacado no suporte as atividades de Engenharia de Requisitos. Contudo este diagrama não tem distinguido a fase inicial (captura dos requisitos organizacionais) da fase final da Engenharia de Requisitos

A fase inicial tem o objetivo de definir como o sistema pretendido irá satisfazer os objetivos da organização, porque ele é necessário, quais as alternativas existentes para as várias partes envolvidas, enquanto a fase final foca na perfeição e consistência do sistema de informação.

As atividades realizadas na fase iniciais da Engenharia de Requisitos têm objetivos e pressuposições que são diferentes da fase final.

Neste sentido este capítulo apresentou um estudo que mostrou de que forma a técnica de modelagem organizacional EKD através de seus modelos pode complementar e servir de fonte de informação para a fase inicial da Engenharia de Requisitos.

Desta forma, observando-se os modelos da técnica EKD e adotando-se as diretrizes propostas, pode-se extrair informações bastante úteis no desenvolvimento de diagrama *Use Case*.

6. Conclusões

Inicialmente este trabalho apresentou o processo de Engenharia de Requisitos e a importância de uma adequada definição de requisitos. A etapa de definição de requisitos é frequentemente negligenciada e, quando realizada configura-se como um documento longo e de difícil análise e validação.

Com base na literatura pesquisada nota-se que a grande maioria das metodologias para análise e especificação de requisitos, como Análise Estruturada e suas variações, e mais recentemente a AOO não considera a organização, ficando centrada apenas nos aspectos técnicos-funcionais, não observando a informação de uma forma mais ampla, a começar pelas necessidades do próprio negócio.

A linguagem UML, muito aceita pelos seguidores da linha OO, não é direcionada para a definição de requisitos organizacionais (fase inicial da Engenharia de Requisitos) levando a falhas na captura das necessidades do negócio. O diagrama *Use Case* tem sido adotado com notável aceitação, é considerado como parte central e fundamental da AOO e tem sido incorporado por várias metodologias recentes. Alguns autores o consideram um bom veículo para planejamento de projeto, porque controla o desenvolvimento interativo, o qual por si só é uma valiosa técnica para mostrar ao usuário o que o sistema deve fazer.

Por outro lado, alguns estudos foram apresentados, mostrando que não há captura de requisitos organizacionais no diagrama *Use Case*. Esse diagrama é tratado por muitos como sendo uma forma de focar em como os usuários vêem as operações do sistema, mas na maioria das vezes o sistema ainda não existe e se existe não é satisfatório, além disso, a visão do usuário está relacionada com os processos computadorizados existentes, que podem ser modos antiquados de operação.

Contudo o desenvolvimento de sistemas ocorre dentro de contextos onde processos organizacionais estão estabelecidos.

Portanto, é preciso capturar os requisitos organizacionais para definir como o sistema pretendido irá satisfazer os objetivos da organização, por que ele é necessário e quais as implicações das alternativas para as várias partes envolvidas.

Desta forma, a grande motivação deste trabalho está em mostrar soluções para os problemas relacionados a captura dos requisitos organizacionais (fase inicial), em geral não considerados em diagrama *Use Case*

Segundo SANTANDER & CASTRO (2000), o fato de se desenvolver um modelo organizacional no processo de Engenharia de Requisitos possibilita uma visão bastante útil no desenvolvimento de *Use Case*.

Neste sentido, este trabalho apresentou a técnica de Modelagem Organizacional EKD. Esta técnica fornece uma forma sistemática e controlada de analisar, entender, desenvolver e documentar uma organização (ROLLAND, 2000).

Os componentes do EKD são modelos conceituais que examinam a organização e seus requisitos a partir de um número de perspectivas inter-relacionadas. Esses modelos são abstrações do mundo físico. Para uma dada organização esses modelos irão constituir coletivamente o Modelo Organizacional. Verifica-se que alguns desses modelos fornecem informações relevantes que apontam a necessidade de avaliação das alternativas de situações operacionais. Como exemplo de características dos modelos podemos citar: i) são fáceis de serem elaborados e lidos porque não apresentam notação complexa; ii) podem ser compreendidos por diferentes grupos de usuários; iii) permitem extensão do escopo a partir da elaboração de novos submodelos necessários para que se tenha representação de características não cobertas pelos submodelos propostos; iv) são independentes da tecnologia, porque não se prendem a linguagens ou ferramentas, podendo ser elaborados a partir de recursos gráficos simples presentes em editores de texto comuns.

Para desenvolver um modelo organizacional completo é necessário fazer ligações entre os componentes de diferentes submodelos. Essas ligações mostraram executar uma tarefa essencial, tornando o conhecimento mais disponível e possibilitando ilustrar por que certos processos e requisitos do sistema de informação têm que ser introduzidos. A habilidade de traçar decisões, componentes e outros aspectos em todas as partes da organização depende do uso e entendimento dessas ligações. Toda ligação representa uma declaração feita sobre a organização e possivelmente seus requisitos do sistema de informação. A semântica de todas as ligações deve ser analisada cuidadosamente. Existe, entretanto, um conjunto mínimo de ligações que deveria ser definido pela representação que fosse considerada completa.

A técnica de modelagem organizacional EKD é uma metodologia que direciona o processo de conhecimento do ambiente sendo adequada para auxiliar a definição de requisitos, quando se tem pouco conhecimento da organização e do sistema de informação da empresa.

No entanto os requisitos organizacionais definidos a partir da técnica EKD precisam ser relacionados aos requisitos técnicos- funcionais representados pelo diagrama *Use Case*.

Dessa forma, o presente trabalho apresentou um estudo, que mostra de que forma o desenvolvimento de um Modelo Organizacional através da técnica EKD pode complementar e servir de fonte de informação para o desenvolvimento do diagrama *Use Case*.

Segundo SANTANDER & CASTRO (2000), esta tarefa de observar os modelos organizacionais e extrair informações para diagramas *Use Case* não é trivial, pois ambas as técnicas permitem enfocar requisitos de software sob perspectivas diferentes.

O estudo consistiu em estabelecer uma correlação entre os submodelos do EKD e os elementos componentes do diagrama *Use Case* com a finalidade de obter uma especificação de requisitos voltada para o ambiente organizacional. Assim fatores do ambiente organizacional inicialmente não considerados em diagrama *Use Case* foram integrados ao mesmo.

Uma das vantagens observadas na integração da técnica EKD e o diagrama *Use Case* foi não apenas a automação dos processos de negócio existentes, mas sim, a reformulação desses processos, a fim de encontrar objetivos do negócio mais amplos como concorrência, competitividade e metas estratégicas na organização, as quais devem ser analisadas e ponderadas no momento do desenvolvimento de um sistema computacional.

Finalmente, considera-se que a partir deste trabalho pode-se originar novos trabalhos, através da observação dos modelos organizacionais e do uso de heurísticas propostas na literatura para o desenvolvimento de diagramas *Use Case*, visando buscar dados estatísticos que comprovem os resultados esperados.

7. Referências Bibliográficas

ALENCAR, F. M. R. (1999). Mapeando a Modelagem Organizacional em Especificações Precisas. Recife. 304p. Tese (Doutorado) - Centro de Informática, Universidade Federal de Pernambuco.

ALENCAR, F. M. R.; SOUZA, F. M.; CASTRO, J. F. B.(1999). Modelagem organizacional: análise comparativa das técnicas I*. IN WORKSHOP IBEROAMERICANO DE ENGENHARIA DE REQUISITOS DE AMBIENTE DE SOFTWARE. São José, Costa Rica. Anais,[S.I.: s.n.], p.326-337.

BOOCH, G., JACOBSON, I., RUMBAUGH, J., (1999). "The Unified Modeling Language User Guide. Addison-Wesley .

BUBENKO, JR., J. A. (1993). Extending the Scope of Information Modeling. In: INTERNATIONAL WORKSHOP ON THE DEDUCTIVE APPROACH TO INFORMATION SYSTEMS AND DATABASE, 4., Lloret-Costa Brava, **Proceedings**. Department de Llenguatges Sistemes Informatics of the Universitat Politecnica de Catalunya, barcelona, Catalonia, A. Olivé (Ed). p. 73-98.

BUBENKO, JR., J. A.; KIRIKOVA, M. (1994). Enterprise Modeling: Improving the Quality of Requirements Specification. In: IRIS-17 INFORMATION SYSTEMS RESEARCH SEMINAR IN SCANDINAVA, Oulu, **Proceedings**, s.n.t.

BUBENKO, JR., J.A.; STIRNA, J.; BRASH, D. (1998). EKD User Guide, DPT of computer and systems sciences. Stockolm, Royal Institute of Technology.

BUBENKO, JR., J.A.; STIRNA, A.; PERSSON. (2001). D3: Appendix B, EKD User Guide, DPT of computer and systems sciences. Stockolm, Royal Institute of Technology.

- BUHR, R. J. A. (1998). Use case maps as architectural entities for complex systems. **IEEE Transactions on Software Engineering**, v. 24, n.12, p.1131-1155, Dec.
- CASTRO, J. F. B. (1995). Introdução à Engenharia de Requisitos. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, Celas, RS. Anais...Porto Alegre: Sociedade Brasileira de Computação, 43p.
- CERVO, A.L.; BERVIAN, P.A. (1983). Metodologia Científica. São Paulo, Makron.
- CHICHINELLI, M.(2002). Contribuição da técnica de Modelagem Organizacional I* ao Processo de Engenharia de Requisitos, com destaque aos Requisitos Funcionais. Dissertação (Mestrado em Engenharia de Produção) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos.
- COAD, P & YOURDON, E. (1992). Análise Baseada em Objetos. Editora Campus.
- DAVIS, A. (1993). Software Requirements: Objects, Functions and States. London: Prentice-Hall.
- DARDENE, A.; LAMSWEERDE, A.; FICKAS, S. (1993). Goal-directed Requirements Acquisition. Science of Computer Programming, vol. 20, p 3-50.
- DUBOIS, E. Et al. (1986). A Knowledge Representation Language for Requirements Engineering. **IEEE Press**, vol. 4, n. 10, p.1431-1444.
- DE BERTOLI, L. A. (1999). Um Método de Trabalho para Auxiliar a Definição de Requisitos. Porto Alegre. Dissertação (Mestrado em Informática) – PPGC, Universidade Federal do Rio Grande do sul.
- ELETRICAL ENTERPRISE KNOWLEDGE FOR TRANSFORMING APPLICATIONS. (2000). The ELEKTRA project programme. www.singular.gr/elektra.ekd.htm

- ERIKSSON, H *et al.* (1998). UML Toolkit. New York: John Wiley and Sons, Inc.
- FICKAS, S.; HELM, R. (1992). Knowledge Representation Reasoning in the Design of Composite System. **IEEE Transactions on Software Engineering**, vol. 18, n. 6, p. 470-482.
- FOWLER, M.; SCOTT, K. (1997). UML distilled - applying the standard object modelling language. Reading, Mass (Addison Wesley Object Technology Series).
- FURLAN, J. D. (1997). Modelagem de Negócio – uma abordagem integrada de modelagem estratégica funcional, de dados e a orientação a objetos. São Paulo: Makron Books.
- FURLAN, J. D.(1998). Modelagem de Objetos através da UML. São Paulo: Makron Books.
- GOTTESDIENER, E. (1999). Capturing business rules. *Software Development*, v.7, n. 12, p.72-76, Dec.
- GREENSPAN, S.; MYLOPOULOS, J.; BORGIDA, A. (1982). Capturing more World Knowledge in the Requirements Specification. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 6, 1982, Tokyo, Jjapan. Proceeding... [S.I.: s.n.]
- HERBST, H. (1996). Business rules in system analysis: a meta-model and repository system. *Information Systems*, v. 21, n.2, p 147-166.
- IEEE STD. 610.12.(1995). IEEE Standard Glossary of Software Engineering Terminology. The Institute of Electrical and Eletronics Engineers. New York.
- JACOBSON, I.(1995). The Use Case Construct in Object-Oriented Software Engineering. Jonh Wiley an Sons, p.309-336.
- JACOBSON, I.; BOOCH, G.; REUMBAUGH, J. (1999). The unified software development process. Reading, Mass, Addison Wesley.

JACKSON, M. (1995). *Software Requirements and Specifications: A Lexicon of Practice, Principles and Prejudices*. USA, Massachusetts: Addison – Wesley, 157p.

KILOV, H.; SIMMONDS, I. (1997). *Business rules: from business specification to design*. In: BOSCH, J.; MITCHELL, S. *Object oriented technology*. Berlin, Springer. (Lectures Notes in Computer Sciences, 1357).

KIRIKOVA, M. (2000). *Explanatory Capability of Enterprise Models*. **IEEE Transactions on Data and Knowledge Engineering**, n33, p. 119-136.

KOTONYA, G. (1994). *A Viewpoint-Oriented Method For Requirements definition*. Lancaster. Tese (Doutorado em Computação) – Departamento de Computação, Lancaster University.

KOTONYA, G.; SOMMERVILLE, I. (1995). *Requirements Engineering With Viewpoints*. Cooperative System Engineering Group – Technical Report CSEG/10/1995. Computing Department, Lancaster University.

KOTONYA, G.; SOMMERVILLE, I. (1998). *Requirements Engineering – Processes and Techniques*. Chichester, England: John Wiley & Sons Inc.

LAKATOS, E. M.; MARCONI, M. A. (2001). *Fundamentos da Metodologia Científica*, São Paulo: Atlas.

LARMAN, C. (1999). *Applying UML and patterns: an introduction object-oriented and design*. New Jersey, Prentice-Hall.

LEE, W.; J., CHA, S. D.; KNON, Y.R. (1998). *Integrating and Analyses of Use Cases Using Modular Petri Nets in Requirements Engineering*. **IEEE Transaction on Software Engineering**, v24, n12, p.1115-1130.

LEITE, J. C. S. P.; FREEMAN, P. A (1991) Requirements Validation Through Viewpoint Resolution. **IEEE Transactions on software Engineering**, vol. 12, n. 12, p.1253-1269.

LEITE, J. C. S. P. (1994). Engenharia de Requisitos. Notas de Aula, PUC-RJ.

LEITE, J. C. S. P. *et al.* (1997). Enhcing a Requirements Baseline with Scenários. In: INTERNATIONAL SIMPOSYUM ON REQUIREMENTS ENGINEERING, 3, 1997, Anápolis. Proceeding, IEEE, p.44-53.

LILLY, S. (2000). How to Avoid Use-Case Pitfalls. *Software Development*, v.8, n.1, p.40-45

LOUCOPOULOS, P. *et al* (1998). Using the EKD Approach: The Modeling Component. Manchester, UMIST.

LOUCOPOULOS, P.; KARAKOSTAS, V.(1995). System Requirements Engineering. London: McGraw-Hill.

MACAULAY, L. (1996). Requirements Engineering. Berlin: Springer Verlag.

MEYER, B. (1997). OOSC2: The Use Case Principle. *Jounal Eiffel Liberty*, v.1, n.2.

MYLOPOULOS, J. *et al.*(1990). Telos: Representing Knowledge about Information Systems. *ACM Transaction on Information Systems*, vol.8, n.4.

PÁDUA, F. S. M; CAZARINI, E.W. (2002). A importância da Modelagem Organizacional no desenvolvimento de diagramas *Use Cases*. ENEGEP - Encontro Nacional de Engenharia de Produção, Curitiba. CD ROM.

PÁDUA, S. I. D. (2001). Investigação do Processo de Desenvolvimento de Software a partir da Modelagem Organizacional, Enfatizando Regras do Negócio.144 p. Dissertação (Mestrado em Engenharia de Produção) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos.

PRADO, A.F. (1999). Desenvolvimento de Software Orientado a Objetos na pós-graduação: notas de aula. <http://www.dc.ufsc.br/~prado/ensinoepesquisa.html>).

PRESSMAN, R. (1995). Engenharia de Software. São Paulo: Makron Books.

ROLLAND, C.; NURCAN, S.; GROSZ, G. (1999). Enterprise Knowledge Development: The Process View. *Information and Management Journal*, v.36, n.3, p. 165-184.

ROLLAND, C.; GROSZ, G.; KLA, R. (1999). Experience with Goal-Scenario Coupling in RE. In **IEEE INTERNATIONAL SYMPOSIUM ON RE**, 4.; Limerich, Ireland, 7-19 June. Proceedings. Ireland, University of Limerich.

SANTANDER, V. F. A; CASTRO, J. F. B. (2000). Desenvolvendo *Use Cases* a partir de Modelagem Organizacional. In: Workshop de Engenharia de Requisitos, Rio de Janeiro. Anais, Rio de Janeiro: PUC-RIO, p.158-179.

SOUZA, F. M.; CASTRO, J. F. B. (1998). Requisitos com Modelagem Organizacional: Um Estudo de Caso no Comércio. In: Workshop de Engenharia de Requisitos. Maringá. Anais, Rio de Janeiro: PUC-RIO, p.18-26.

SCHNEIDER, G., WINTERS, J.P. (1998). *Applying Use Cases: A Practical Guide*, Addison Wesley.

SUTCLIFFE, A. G. *et al.* (1998). Supporting scenario-based requirements engineering. **IEEE Transaction on Software Engineering**, v.24, n.12, p. 1072-1088, Dec.

TURINE, M. A. S.; MASIERO, P. C. (1996). Especificação de Requisitos: Uma introdução. Relatório Técnico. ICMSC: USP.