

UNIOESTE – Universidade Estadual do Oeste do Paraná

CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

Colegiado de Ciência da Computação

Curso de Bacharelado em Ciência da Computação

**Avaliando técnicas de modelagem organizacional no
contexto de desenvolvimento de sistemas
computacionais**

Fernando Luiz Grando

CASCABEL

2010

FERNANDO LUIZ GRANDO

**AVALIANDO TÉCNICAS DE MODELAGEM ORGANIZACIONAL NO
CONTEXTO DE DESENVOLVIMENTO DE SISTEMAS
COMPUTACIONAIS**

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel.

Orientador: Prof. Dr. Victor Francisco Araya Santander

CASCADEL

2010

FERNANDO LUIZ GRANDO

**AVALIANDO TÉCNICAS DE MODELAGEM ORGANIZACIONAL NO
CONTEXTO DE DESENVOLVIMENTO DE SISTEMAS
COMPUTACIONAIS**

Monografia apresentada como requisito parcial para obtenção do Título de *Bacharel em Ciência da Computação*, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Dr. Victor Francisco Araya Santander

(Orientador)

Colegiado de Ciência da Computação,
UNIOESTE

Prof. Msc. Anibal Mantovani Diniz

Colegiado de Ciência da Computação,
UNIOESTE

Prof. Dr. Clodis Boscarioli

Colegiado de Ciência da Computação,
UNIOESTE

Cascavel, 03 de Dezembro de 2010.

DEDICATÓRIA

*Aos meus pais e minha irmã, que sempre me incentivaram, me apoiaram, acreditaram nos meus sonhos e me deram forças para continuar o meu caminho. Sem vocês eu nada seria.
Muito obrigado!*

“Seu tempo é limitado, então não percam tempo vivendo a vida de outro. Não sejam aprisionados pelo dogma – que é viver com os resultados do pensamento de outras pessoas. Não deixe o barulho da opinião dos outros abafar sua voz interior. E mais importante, tenha a coragem de seguir seu coração e sua intuição. Eles de alguma forma já sabem o que você realmente quer se tornar. Tudo o mais é secundário”.

Steve Paul Jobs.

AGRADECIMENTOS

A Deus, pela saúde e disposição, permitindo mais esta evolução pessoal e profissional na caminhada da vida.

Aos meus pais, Agemiro Grando e Semilda Maria Huppers, pelo amor, incentivo e dedicação em todos os momentos da minha vida. Agradeço eternamente a vocês pela alegria e cuidados que me proporcionaram.

A minha irmã, Deisy Daniella Grando, que nunca mediu esforços para me ajudar. Sou muito grato pelo carinho, amizade e constante apoio que me dedicou ao longo dos anos.

A minha namorada, Adrieli Tailini Lothermann, pela compreensão, compartilhando das minhas angústias e alegrias, pelo seu amor e carinho incondicionais. Agradeço por ser essa luz que sempre brilha na minha vida. É com quem quero compartilhar essa e outras conquistas.

Aos meus amigos, Alessandro Rodrigo Franco, Daniel Bordignon Cassanelli, Fernando Martins, Marco Antonio da Rosa, Pedro Patitucci Finamore e Rafael Almeida de Oliveira, pela convivência, amizade e companheirismo, tendo a certeza de que me ajudaram a evoluir como ser humano e também como profissional.

Ao professor Victor Francisco Araya Santander, pelo grande apoio, confiança, estímulo e valiosa orientação neste trabalho. Meus sinceros agradecimentos.

Aos professores Anibal Mantovani Diniz e Clodis Boscarioli, membros da banca, responsáveis por importantes comentários e sugestões, contribuindo para melhorar este trabalho.

A empresa Tecinco – Tecnologia em Informática Corporativa, pelo estágio fornecido, disponibilizando dessa maneira subsídios tecnológicos e estudo de caso para a realização desta monografia.

Enfim, lembro e sou grato a todos aqueles que direta ou indiretamente contribuíram para a realização deste trabalho, não havendo aqui, infelizmente, condições de a todos nominar. Que Deus retribua a cada um com saúde e paz.

Lista de Figuras

2.1: Entradas e Saídas do Processo de Engenharia de Software.....	14
2.2: O processo principal da abordagem GQM	22
3.1: Submodelos que compõem o Modelo Organizacional	28
3.2: Parte de um MO de uma Biblioteca.....	30
3.3: Parte de uma interação entre o MRN e componentes do MO	31
3.4: O processo de verificação do endereço do cliente não decomposto.....	33
3.5: O processo de verificação do endereço do cliente decomposto	33
3.6: Exemplo de MAR de uma Biblioteca.....	34
3.7: Exemplo de parte de um MRCT de uma Biblioteca.....	36
3.8: Exemplo de subprocesso de forma “fechada”	39
3.9: Exemplo de subprocesso de forma “aberta”	39
3.10: Exemplo de utilização de <i>pools</i>	40
3.11: Exemplo de utilização de <i>lanes</i>	41
3.12: Exemplo de segmento de processo utilizando artefatos	42
3.13: Exemplo de Processo Interno	42
3.14: Exemplo de Processo Abstrato	43
3.15: Exemplo de Processo de Colaboração.....	43
3.16: Relacionamentos de dependência entre atores em i^*	45
3.17: Notação gráfica dos elementos e ligações em i^*	47
3.18: Modelo de Dependências Estratégicas para Agendamento de Reuniões	47
3.19: Modelo de Razões Estratégicas para Agendamento de Reuniões	49
3.20: Metamodelo de <i>Workflow</i>	51
3.21: Exemplo de <i>Workflow</i> representando um sistema de atendimento <i>on-line</i>	53
3.22: O ambiente de Regras de Negócio.....	56
3.23: Exemplo de <i>Map</i>	58

3.24: Notações básicas para Casos de Uso em UML	59
3.25: Exemplo de Caso de Uso de Negócio.....	59
4.1: Ciclo de vida de projeto da empresa desenvolvedora de software	66
4.2: Diagrama BPMN para o módulo Cotação de Compras.....	72
4.3: Diagrama SD para o módulo Cotação de Compras	73
4.4: Diagrama SR para o módulo Cotação de Compras	74
4.5: Diagrama SD para o módulo Cotação de Compras, com a inclusão do Sistema	75
4.6: Diagrama SR para o módulo Cotação de Compras, com a inclusão do Sistema.....	76
4.7: Diagrama <i>Map</i> para o módulo Cotação de Compras	78
4.8: Diagrama <i>Map</i> para o objetivo Analisar Preços dos Itens.....	79
4.9: Diagrama de Casos de Uso de Negócio para o módulo Cotação de Compras	80
4.10: Diagrama de Casos de Uso de Sistema para o módulo Cotação de Compras	81
4.11: MO e MRN para o módulo Cotação de Compras.....	81
4.12: MC para o módulo Cotação de Compras.....	82
4.13: MPN para o módulo Cotação de Compras	82
4.14: MAR para o módulo Cotação de Compras	83
4.15: MRCT para o módulo Cotação de Compras.....	83

Lista de Tabelas

2.1: Entradas e Saídas do Processo de Engenharia de Requisitos	13
2.2: Comparação das estratégias experimentais	19
2.3: Estrutura para definição das metas GQM	21
3.1: Objetos de Fluxo.....	38
3.2: Objetos de Conexão	39
3.3: <i>Swimlanes</i>	40
3.4: Artefatos	41
4.1: Características das técnicas.....	84

Lista de Abreviaturas e Siglas

B2B	<i>Business to Business</i>
BPD	<i>Business Process Diagram</i>
BPEL	<i>Business Process Execution Language</i>
BPMI	<i>Business Process Management Initiative</i>
BPMN	<i>Business Process Modeling Notation</i>
BPMN-WG	<i>Business Process Modeling Notation Working Group</i>
BRG	<i>Business Rules Group</i>
DRP	Documento de Requisitos de Projeto
ECA	Evento/Condição/Ação
ECAA	Evento/Condição/Então-Ação/Senão-Ação
EKD	<i>Enterprise Knowledge Development</i>
ER	Engenharia de Requisitos
ES	Engenharia de Software
ESE	Engenharia de Software Experimental
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ISO	<i>International Standards Organization</i>
MAL	<i>Modal Action Logic</i>
MAR	Modelo de Atores e Recursos
MC	Modelo de Conceitos
MO	Modelo de Objetivos
MPN	Modelo de Processos de Negócio
MPS.BR	Melhoria de Processos do Software Brasileiro
MRCT	Modelo de Requisitos e Componentes Técnicos
MRN	Modelo de Regras de Negócio
OCL	<i>Object Constraint Language</i>

OMG	<i>Object Management Group</i>
QGM	<i>Goal/Question/Metric</i>
QIP	<i>Quality Improvement Paradigm</i>
RF	Requisito Funcional
RNF	Requisito Não Funcional
RUP	<i>Rational Unified Process</i>
SD	<i>Strategic Dependency</i>
SI	Sistema de Informação
SR	<i>Strategic Rationale</i>
TA	Teoria da Atividade
UML	<i>Unified Modeling Language</i>
WF	<i>Workflow</i>
WFMC	<i>Workflow Management Coalition</i>
XPDL	<i>XML Process Definition Language</i>

Sumário

Lista de Figuras	vi
Lista de Tabelas.....	viii
Lista de Abreviaturas e Siglas.....	ix
Sumário	xi
Resumo	xiii
1 Introdução.....	1
1.1 Contexto	1
1.2 Motivações.....	4
1.3 Proposta	6
1.4 Contribuições Esperadas.....	7
1.5 Estrutura do Trabalho	7
2 A Engenharia de Requisitos e a Engenharia de Software Experimental	9
2.1 Engenharia de Requisitos	9
2.1.1 Classificação de Requisitos.....	11
2.1.2 Processo de Engenharia de Requisitos.....	13
2.1.3 Problemas na área de Engenharia de Requisitos.....	15
2.2 Engenharia de Software Experimental	17
2.2.1 Objetivos da Experimentação	18
2.2.2 Tipos de Experimentos	19
2.2.3 Metodologia e Experimentação	20
2.3 Considerações Finais	23
3 Modelagem Organizacional.....	25
3.1 Introdução.....	25
3.2 A Técnica EKD	28
3.2.1 O Modelo de Objetivos (MO).....	29
3.2.2 O Modelo de Regras de Negócio (MRN)	30
3.2.3 O Modelo de Conceitos (MC).....	31
3.2.4 O Modelo de Processos de Negócio (MPN)	32

3.2.5 O Modelo de Atores e Recursos (MAR).....	33
3.2.6 O Modelo de Requisitos e Componentes Técnicos (MRCT).....	35
3.3 <i>Business Process Modeling Notation</i> (BPMN)	36
3.3.1 <i>Business Process Diagram</i> (BPD)	37
3.3.2 Tipos de submodelos BPMN	42
3.4 A Técnica i*.....	44
3.4.1 O Modelo de Dependências Estratégicas (SD)	45
3.4.2 O Modelo de Razões Estratégicas (SR)	48
3.5 Fluxos de Trabalho (<i>workflows</i>)	50
3.6 Regras de Negócio.....	53
3.6.1 Representação de Regras de Negócio	54
3.7 <i>Maps</i>	57
3.8 Casos de Uso de Negócio	58
3.9 Análise das Técnicas	61
3.10 Considerações Finais	63
4 Estudo Comparativo e Avaliação dos Modelos Organizacionais.....	64
4.1 Estudo Comparativo	64
4.1.1 Contexto do Estudo de Caso – Cotação de Compras	65
4.1.2 Utilizando a Engenharia de Software Experimental	68
4.1.3 Utilizando as Técnicas de Modelagem Organizacional	71
4.1.4 Resultados Obtidos	83
4.2 Avaliação	85
4.3 Considerações Finais	89
5 Considerações Finais.....	92
Referências Bibliográficas	95

Resumo

Hoje em dia, o desenvolvimento de softwares complexos e de baixo custo é um grande desafio para a Computação, em especial para a área de Engenharia de Software. Geralmente os requisitos de software são mal analisados e especificados, o que resulta em programas de baixa qualidade. No entanto, observa-se que o trabalho da Engenharia de Requisitos pode ser melhorado se modelarmos aspectos organizacionais visando entender melhor as intenções e motivações organizacionais que incorporam o desejo do desenvolvimento de um sistema computacional. Verifica-se nesse contexto a necessidade de realizar um estudo comparativo entre técnicas de modelagem organizacional existentes, de forma a apresentar uma avaliação das mesmas, destacando as suas principais características e qualidades no processo de elicitação de requisitos de software. Para tanto, neste trabalho realiza-se uma avaliação destas técnicas utilizando princípios da Engenharia de Software Experimental. Como resultado, obtém-se uma análise crítica dos modelos organizacionais estudados, visando apoiar engenheiros de requisitos em uma melhor escolha da técnica a ser aplicada. A técnica i^* pode ser considerada a que mais se destacou na avaliação realizada, permitindo, de forma abrangente, representar vários aspectos do contexto organizacional, além de possibilitar a alocação de responsabilidades ao sistema computacional inserido como um novo ator na modelagem da organização.

Palavras-chave: Engenharia de Requisitos, Modelagem Organizacional, Desenvolvimento de Software.

Capítulo 1

Introdução

Esse capítulo tem como objetivo a apresentação do trabalho, indicando as motivações para o seu desenvolvimento, bem como a sua estruturação. Inicialmente, na Seção 1.1, a proposta é contextualizada no escopo da Engenharia de Software, destacando a importância do processo de Engenharia de Requisitos na obtenção de produtos de software de qualidade. Na Seção 1.2, são descritas as principais motivações que levaram a execução deste estudo. Na Seção 1.3, a proposta em si é explanada, bem como seus principais objetivos. Já na Seção 1.4, são relatadas as contribuições esperadas com o desenvolvimento do presente trabalho. Por fim, na Seção 1.5 a organização desse documento é apresentada.

1.1 Contexto

Atualmente, o desenvolvimento de softwares cada vez mais complexos, passíveis de certificação e com menor custo possível, é um desafio constante para a comunidade de Engenharia de Software (ES) (Santander e Silva, 2006).

Nota-se que muitas pessoas envolvidas na área de desenvolvimento de software ainda não estão amadurecidas o suficiente para lidar com sistemas complexos e de grande porte, utilizando muitas vezes técnicas que não são apropriadas para esses tipos de sistemas. Como resultado, obtém-se uma série de problemas, que vão desde a estimativa equivocada dos custos envolvidos e cronogramas mal planejados, até a baixa produtividade e, principalmente, a baixa qualidade dos produtos.

Diversas técnicas 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 (ER), cujo processo é definido por (Kotonya e Sommerville, 1998), como sendo “*um*

conjunto estruturado de atividades que são seguidas para derivar, validar e manter um documento de requisitos de sistema”.

Para se realizar um bom trabalho de Engenharia de Requisitos, faz-se necessário o entendimento dos aspectos sociais, organizacionais, técnicos, econômicos e jurídicos envolvidos no domínio do desenvolvimento, de forma a minimizar riscos e custos (Alencar, 1999).

Freqüentemente, requisitos de software são inadequadamente elicitados, analisados e especificados, sendo estes fatores decisivos para o desenvolvimento de softwares de baixa qualidade. Desta forma, boa parte da atenção da comunidade acadêmica e industrial em nível nacional e internacional está concentrada em encontrar caminhos que permitam desenvolver softwares que considerem adequadamente os requisitos organizacionais, funcionais e não funcionais (Santander e Silva, 2006).

Quando usuários precisam ou desejam um determinado software, em grande parte dos casos não existe uma ideia concreta do que se espera que esse sistema computacional faça. Geralmente, o que se tem, são intenções e desejos de facilitar a execução de atividades no ambiente organizacional.

Por outro lado, no processo de desenvolvimento de software, atenta-se inicialmente à necessidade de se elicitar os requisitos do sistema junto ao usuário, procurando obter o que o mesmo deseja do sistema. Para tanto, faz-se necessário uma compreensão mais ampla, procurando focar todos os atores (com suas especializações em agentes, posições e papéis) e objetos que se inter-relacionam dentro da empresa onde o sistema a ser desenvolvido se localizará.

A modelagem dos processos de negócio é essencial, pois são atividades previamente estabelecidas que tem como objetivo determinar a forma como o trabalho é realizado numa organização. Constituem um conjunto de ações, relacionadas entre si de forma lógica e coerente a fim de promover uma saída favorável à organização, tanto em nível interno como externo. Uma estrutura de processos de negócio mal concebida pode por em risco a eficiência e a eficácia de uma empresa.

A finalidade de modelar um negócio é criar uma abstração, que é uma visão simplificada do negócio (Eriksson e Penker, 2000). Um modelo de negócio mostra qual é o ambiente da

organização e como a organização age em relação a este ambiente¹ (Jacobson, Ericson e Jacobson, 1994).

Já a modelagem organizacional, em um sentido mais amplo, é a representação da estrutura, das atividades, dos processos, das informações, dos recursos, do pessoal, do comportamento, dos objetivos e das restrições das empresas comerciais, governamentais ou de outra natureza. Esse modelo ajuda a compreender as complexas interações entre as organizações e as pessoas.

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

É fato que os modelos tradicionalmente utilizados, considerando a Ciência da Computação, são intimamente ligados com aspectos das funcionalidades dos sistemas, ou seja, estão preocupados com o “que” e “como” fazer, permitindo apenas descrições de entidades e atividades envolvidas nos processos. Já sob o ponto de vista da Administração, as técnicas visam modelar a estrutura interna da organização, evidenciando suas funções, dados e estratégias, descrevendo assim o “porque” de realizar determinado processo (Alencar, 1999).

Neste sentido, a modelagem organizacional permite não só melhor entender requisitos organizacionais que interferirão nos sistemas, mas também identificar alternativas para os vários processos da organização, facilitando os esforços durante o desenvolvimento do sistema computacional e permitindo que a análise organizacional seja mais bem integrada aos processos de desenvolvimento do sistema. O entendimento dos aspectos sociais, organizacionais, técnicos, jurídicos e econômicos é fundamental para a realização de um bom trabalho de Engenharia de Requisitos.

Segundo Yu (1995a), são necessários modelos capazes de reconhecer motivações, intenções e razões sobre as características dos processos existentes, a fim de que, ao final do processo de desenvolvimento, tenha-se uma documentação apropriada que permita mais facilmente a compreensão das características essenciais do sistema.

Desta forma, este trabalho tem como foco estudar as técnicas atuais utilizadas para modelar organizações, porém, com uma visão voltada para sua utilização juntamente com modelos funcionais. Uma expectativa e necessidade de engenheiros de requisitos e engenheiros de software é que estes modelos organizacionais que já possuem necessidades expressas na

¹ Por ambiente entende-se tudo que a organização interage para realizar os seus processos de negócio, como clientes, empregados e parceiros.

forma de satisfação de objetivos, realização de tarefas e obtenção de recursos, possam servir de base para capturar a funcionalidade de sistemas computacionais pretendidos pela organização. Tipicamente deseja-se capturar essa funcionalidade utilizando elementos integrantes da UML (*Unified Modeling Language*) (Booch, Jacobson e Rumbaugh, 1999), como casos de uso, diagrama de classes e outros diagramas da orientação a objetos.

A próxima seção apresenta as principais motivações para a realização desse trabalho.

1.2 Motivações

Observa-se na literatura (Yu, 1995a), (Castro, Kolp e Mylopoulos, 2002), (Castro *et al.*, 2010), que o trabalho da ES, mais especificamente da ER, pode ser melhorado se os aspectos organizacionais forem modelados visando entender melhor as intenções e motivações da organização que incorporam o desejo do desenvolvimento de um software.

Neste sentido, várias propostas objetivando a modelagem organizacional têm sido apontadas, como a Técnica *i** (Yu, 1995a), a Técnica EKD (*Enterprise Knowledge Development*) (Bubenko Jr., Stirna e Brash, 1998), Fluxos de Trabalho (*Workflows*) (Estrada *et al.*, 2001), Regras de Negócio (Rosca *et al.*, 1997), Notação de Modelagem de Processos de Negócios (*Business Process Modeling Notation - BPMN*) (BPMN, 2010a), *Maps* (Rolland, Prakash e Benjamin, 1999), (Rolland, 2007), Casos de Uso de Negócio (Jacobson, Booch e Rumbaugh, 1999), (Jacobson, 1992).

Contudo, poucos são os estudos a fim de comparar essas técnicas visando avaliar suas características e qualidades no processo de desenvolvimento de software. Assim, é bastante motivador o fato de poder realizar um estudo comparativo entre técnicas de modelagem organizacional existentes com a finalidade de apresentar uma avaliação crítica das mesmas, destacando as suas características, propriedades, comportamentos e qualidades no que diz respeito ao processo de elicitação e especificação de requisitos para o sistema computacional pretendido pela organização.

Cabe ressaltar que alguns estudos como (Santander e Castro, 2002), (Vicente *et al.*, 2009) e (Castro *et al.*, 2010) apontam algumas vantagens da técnica de modelagem organizacional *i** em relação às demais, pois a mesma permite expressar além da descrição das atividades de processos de negócio, os aspectos intencionais, motivações e possíveis alternativas a serem consideradas em processos de negócio apoiados por sistemas computacionais. Entretanto, outras técnicas como BPMN e *Maps* vêm recebendo atenção especial na área de engenharia

de software, em parte pela simplicidade que ambas as técnicas oferecem para modelar ambientes organizacionais, bem como pela rápida assimilação das técnicas por engenheiros de software.

Deve-se considerar o fato de a grande maioria das empresas desenvolvedoras de software em nível nacional considerarem a adoção do programa para Melhoria de Processos do Software Brasileiro (MPS.BR) como fundamental para sobrevivência no mercado. Neste contexto, a modelagem organizacional é considerada como uma atividade essencial para obter maiores níveis de maturidade nos processos de desenvolvimento adotados, bem como para desenvolver produtos de qualidade. Mais especificamente, observando o MPS.BR verifica-se que não se pode gerenciar requisitos e/ou gerenciar projetos se não se pode estabelecer uma relação clara e direta entre as metas organizacionais expressas em modelos organizacionais e metas associadas aos sistemas computacionais pretendidos. Além disso, requisitos são muito mais facilmente elicitados, analisados, validados e rastreados a partir da análise de modelos organizacionais.

É importante frisar que para a realização deste trabalho, serão utilizados princípios da Engenharia de Software Experimental (ESE), pois a mesma fornece mecanismos para o desenvolvimento de experimentos reais abordando as técnicas organizacionais estudadas. Uma grande percentagem da pesquisa em Engenharia de Software exhibe uma recorrente limitação, a validação insuficiente das propostas dos pesquisadores.

Frequentemente apenas são fornecidas provas de conceito elementares, mas não evidências concretas de que as propostas permitem reduzir os problemas identificados. A ESE vem mitigar esta situação, através da aplicação do método científico, que é uma abordagem fundamental usada por cientistas de todas as áreas do conhecimento para testar hipóteses e sustentar teorias. É uma abordagem que inclui uma série de passos, onde é possível identificar relações de causa-efeito.

Ressalta-se que somente por meio de estudos de caso reais é que se pode destacar as dificuldades vividas nas empresas inerentes ao processo de desenvolvimento de software (Wohlin *et al.*, 2000).

Não se pode evoluir com qualidade em direção às demais etapas do processo de desenvolvimento de software se o ambiente organizacional no qual o sistema computacional

será inserido não foi modelado e adequadamente entendido pelos *stakeholders*². Neste sentido, cabe destacar os diversos relatos na literatura associados a grandes perdas de investimentos no desenvolvimento de sistemas de software que não atentaram adequadamente a este aspecto e por isso não atenderam as expectativas dos seus clientes e usuários (Kotonya e Sommerville, 1998).

Na seção seguinte será apresentada a proposta e os objetivos do presente trabalho.

1.3 Proposta

Nesse trabalho, primeiramente será realizado um estudo entre diferentes técnicas de modelagem organizacional, com intuito de compreendê-las tanto do ponto de vista teórico quanto prático.

Após esse estudo inicial, será desenvolvido um estudo comparativo entre essas técnicas, a fim de avaliar as características e qualidades existentes no processo de desenvolvimento de software. Para este fim, serão utilizados princípios da Engenharia de Software Experimental. O estudo de caso adotado envolve o desenvolvimento de um novo módulo para o sistema de compras de um software para gerenciamento comercial e de concessionárias, disponibilizado por uma empresa desenvolvedora de software da região oeste do Paraná.

Com base nesse experimento, será realizada uma avaliação a respeito das técnicas de modelagem organizacional estudadas, considerando suas características e qualidades identificadas no processo de experimentação.

Entre os trabalhos relacionados, pode-se citar o trabalho realizado por Pádua, Cazarini e Inamasu (2004), onde os autores apresentam a modelagem organizacional como uma forma de capturar requisitos organizacionais para melhorar a compreensão do domínio, interagir com usuários, para que eles entendam o que o sistema pode fazer para melhorar o negócio, e adquirir conhecimento da estrutura organizacional e estratégica. Nesse sentido, algumas técnicas de modelagem organizacional foram apresentadas, um breve estudo comparativo entre elas foi realizado e o modelo EKD foi detalhado.

Na seção seguinte são relatadas as contribuições esperadas com o desenvolvimento desse trabalho.

² Em português, parte interessada ou interveniente. É um termo usado para se referir a qualquer pessoa ou entidade que afeta ou é afetada pelas atividades de uma organização.

1.4 Contribuições Esperadas

Espera-se com este trabalho contribuir com a pesquisa na área de Engenharia de Requisitos através de um estudo comparativo entre diversas técnicas de modelagem organizacional existentes.

A partir da avaliação das técnicas estudadas via experimento realizado em uma empresa de desenvolvimento de software de médio porte, apresenta-se uma avaliação crítica dos pontos positivos e negativos das respectivas técnicas. Cabe salientar que a análise crítica realizada também pode ser utilizada em contextos de empresas de maior porte, auxiliando engenheiros de requisitos a realizar uma melhor escolha da técnica a ser aplicada.

De maneira complementar, deseja-se que o estudo desenvolvido facilite a obtenção e captura das funcionalidades de sistemas computacionais pretendidos pela organização através de diagramas orientados a objetos. Inclusive, este aspecto é considerado na definição das métricas adotadas para avaliar as diferentes características de cada modelo estudado.

1.5 Estrutura do Trabalho

Este trabalho se divide em quatro capítulos além do presente, estruturados da maneira apresentada a seguir.

No Capítulo 2 são abordados os principais conceitos da Engenharia de Requisitos e da Engenharia de Software Experimental. Com relação à Engenharia de Requisitos, é apresentada uma introdução ao tema, a Classificação de Requisitos de Software, o Processo de Engenharia de Requisitos e os problemas atuais da área. Da mesma forma, uma introdução à Engenharia de Software Experimental é apresentada, citando posteriormente os principais Objetivos da Experimentação, os Tipos de Experimentos e a Metodologia utilizada na presente proposta.

O Capítulo 3 apresenta uma visão geral a respeito da importância da modelagem de requisitos organizacionais no contexto de desenvolvimento de sistemas computacionais. São apresentadas também diversas técnicas atuais de modelagem organizacional, tais como: técnica EKD, notação BPMN, técnica i^* , Fluxos de trabalho (*workflows*), Regras de Negócio, *Maps*, Casos de Uso de Negócio.

Já no Capítulo 4 o estudo comparativo das técnicas é descrito, auxiliado pelos princípios da Engenharia de Software Experimental. Posteriormente, uma avaliação a respeito dos modelos estudados é apresentada.

Por fim, o Capítulo 5 relata as conclusões obtidas no desenvolvimento deste trabalho, assim como as principais contribuições que ele fornece para Engenharia de Software, particularmente ao Processo de Engenharia de Requisitos. Ao final, possíveis trabalhos futuros são apresentados.

Capítulo 2

A Engenharia de Requisitos e a Engenharia de Software Experimental

Primeiramente, esse capítulo aborda os principais conceitos com relação a Engenharia de Requisitos, ressaltando a importância da elicitação e análise dos requisitos organizacionais no contexto de desenvolvimento de sistemas. Uma introdução à Engenharia de Requisitos é apresentada na Seção 2.1. De forma complementar, esse capítulo apresenta uma introdução à Engenharia de Software Experimental, que servirá de apoio aos estudos comparativos realizados no Capítulo 4. A Seção 2.2 introduz o conceito de Engenharia de Software Experimental. Por fim, a Seção 2.3 contém as considerações finais.

2.1 Engenharia de Requisitos

A Engenharia de Requisitos (ER) tem sido reconhecida como uma das mais importantes fases do processo de Engenharia de Software (ES). Este reconhecimento decorre da descoberta que geralmente a maior parte dos problemas no desenvolvimento de software é originada nas etapas iniciais do desenvolvimento. Estas etapas constituem o processo de Engenharia de Requisitos, a qual tem como principais atividades: elicitação, análise, negociação, especificação, gerenciamento e validação de requisitos (Kotonya e Sommerville, 1998).

No processo de desenvolvimento de software, estabelecer requisitos que sejam compreensíveis por todas as partes envolvidas é um fator básico, ao tempo que é, também, um problema de difícil solução. Dessa forma, faz-se necessária uma abordagem sistemática de obtenção dos requisitos que assegure a compreensão dos requisitos do usuário e a produção de um sistema utilizável a um bom custo/benefício (Alencar, 1999).

Uma incorreta identificação de requisitos de um software pode levar ao desenvolvimento de um produto que não atende aos reais objetivos para o qual foi planejado, resultando em um software de baixa qualidade.

A literatura possui várias definições para requisitos de software. Sommerville (2001) define requisitos de software como *“um conjunto de atividades que o software deve desempenhar, com suas limitações e restrições, além de características não ligadas diretamente às funções desempenhadas pelo software”*.

De acordo com (Sommerville e Sawyer, 1997), um requisito pode descrever uma:

- a) facilidade no nível do usuário: por exemplo, um corretor de gramática e ortografia;
- b) propriedade muito geral do sistema: por exemplo, o sigilo de informações;
- c) restrição específica no sistema: por exemplo, o tempo de varredura de um sensor;
- d) restrição no desenvolvimento do sistema: por exemplo: a linguagem C++ deverá ser utilizada para o desenvolvimento do sistema.

A IEEE (1997) define requisito como sendo:

- a) uma condição ou uma capacidade de que o usuário necessita para solucionar um problema ou alcançar um objetivo;
- b) uma condição ou uma capacidade que deve ser alcançada ou possuída por um sistema ou componente do sistema para satisfazer um contrato, um padrão, uma especificação ou outros documentos impostos formalmente;
- c) uma representação documentada de uma condição ou capacidade, conforme os itens (a) e (b).

Macaulay (1996) possui uma definição bastante simples, na qual ele diz que requisito é simplesmente algo que o cliente necessita. A visão de requisitos, segundo Davis (1993), não se restringe a apenas o "quê" o sistema deva fazer, mas também a "como" deva fazer. Para esse autor, a definição do IEEE não é muito boa, pois só trata da visão do “quê”.

Já a IEEE (1984) define a Engenharia de Requisitos como *“um processo de aquisição, refinamento e verificação das necessidades do cliente para um sistema de software, objetivando-se ter uma especificação completa e correta dos requisitos de software”*.

Segundo Boehm (1989), a Engenharia de Requisitos pode ser definida como *“uma atividade que objetiva desenvolver uma especificação completa, consistente, não ambígua e correta dos requisitos, que sirva, inclusive, de base para um acordo entre as partes*

envolvidas no processo de desenvolvimento do software, onde se pactue, de forma concisa, "o que" o produto irá fazer".

Segundo Castro (1995), Engenharia de Requisitos é um campo amplo que compreende todas as atividades necessárias para criar e manter a documentação de requisitos. Sua função principal é aperfeiçoar os processos para o gerenciamento do ciclo de vida dos requisitos e aborda um ponto fundamental do desenvolvimento de software: a definição do que produzir.

Para Kotonya e Sommerville (1998), a Engenharia de Requisitos é um termo relativamente novo que foi criado para cobrir todas as atividades envolvidas em descobrimento, documentação e manutenção de um conjunto de requisitos para um sistema baseado em computador.

A obtenção dos requisitos de software é uma tarefa complexa, visto que demanda de um conjunto de ideias que podem estar incompletas, sendo necessário transformar essas ideias em uma elaboração completa e consistente de técnicas de requisitos para um sistema de software. Dificilmente as especificações permanecem inalteradas durante o processo. Esta volatilidade pode ocasionar um custo e tempo maior.

Segundo Kotonya e Sommerville (1998), erros nos requisitos são responsáveis pelos seguintes problemas:

- a) pelo atraso na entrega do sistema e pelo aumento do custo do desenvolvimento do sistema;
- b) pela insatisfação do cliente e dos usuários finais com o sistema. Eles podem não usar suas facilidades ou podem ainda descartar por completo o sistema;
- c) pela falta de confiabilidade no uso do sistema, devido a erros regulares do sistema e falhas que levam à interrupção da operação normal do sistema;
- d) pelo aumento do custo de manutenção e evolução do sistema.

Na seção seguinte, os diferentes tipos de requisitos de software serão classificados.

2.1.1 Classificação de Requisitos

Durante a fase de especificação dos requisitos, surge a necessidade de se estabelecer o tipo de requisito de que se esteja tratando, a fim de melhorar a compreensão das necessidades do cliente, bem como melhor modelá-las. Numa visão mais tradicional, os requisitos são classificados como funcionais e não funcionais, conforme apresentado por Sommerville (2001):

- a) **Requisitos Funcionais (RF):** são as declarações das funções que o sistema deve oferecer, como ele se comporta com entradas particulares e como o sistema deve se comportar em situações específicas. O termo “função” é usado no sentido genérico da operação que pode ser realizada pelo sistema, seja por meio de comandos dos usuários, seja pela ocorrência de eventos internos ou externos ao sistema. Em alguns casos, os requisitos funcionais podem também, explicitamente, definir o que o sistema não deve fazer;
- b) **Requisitos Não-Funcionais (RNF):** são as restrições nas funções oferecidas pelo sistema. Incluem restrições de tempo; restrições no processo de desenvolvimento; padrões; e de qualidades globais de um software, como custo, confiabilidade, manutenibilidade, portabilidade, usabilidade, desempenho, dentre outras. Alguns desses requisitos são provavelmente traduzidos em funções (operacionalizados), ao longo do processo de desenvolvimento de software (Chung *et al.*, 2000).

Resumidamente, os requisitos funcionais descrevem “o que” o sistema deve fazer, enquanto que os requisitos não funcionais fixam restrições sobre “como” os requisitos funcionais serão implementados.

Entretanto, devido às novas concepções a respeito de estratégias na gestão das organizações, vários estudiosos afirmam que a classificação tradicional não mais apresenta respostas à compreensão de todas as facetas que compõem o ambiente das atuais organizações. Para melhor compreensão da organização, é necessário não analisar somente requisitos funcionais, mas também os aspectos organizacionais e sociais.

Dessa maneira, a classificação mais moderna de requisitos agrega a conceituação de requisitos organizacionais (Loucopoulos e Karakostas, 1995), (Kotonya e Sommerville, 1998), (Davis, 1993):

- c) **Requisitos Organizacionais:** dizem respeito às metas da empresa, suas políticas estratégicas adotadas, os relacionamentos entre os seus atores junto com seus respectivos objetivos.

Segundo essa nova visão, a Engenharia de Requisitos pode ser dita como a área do conhecimento preocupada na comunicação com agentes organizacionais, com respeito a suas visões, intenções e atividades relativas às suas necessidades de suporte de computadores, desenvolvimento e manutenção de uma especificação de requisitos adequada a um sistema (Loucopoulos e Karakostas, 1995).

Pode-se afirmar que os requisitos na visão moderna não se preocupam apenas com "como" o sistema deve fazer, mas sim com "o que" o sistema deve fazer, associado com "o porque" fazer, compreendendo a intencionalidade dos fatos organizacionais.

A interdependência entre as organizações, entre as unidades internas de uma organização e entre os membros de uma equipe de trabalho é cada vez maior no atual estágio de desenvolvimento da sociedade. A principal razão para se considerar os requisitos organizacionais, está na ajuda à compreensão de interações complexas entre as organizações e as pessoas envolvidas, onde projetistas e desenvolvedores devem dar atenção especial aos componentes organizacionais (razões, motivações, porquês) no desenvolvimento de seus produtos.

2.1.2 Processo de Engenharia de Requisitos

Segundo a ISO (1988), processo é definido como *"um grupo de atividades inter-relacionadas que se caracterizam por uma série de entradas específicas que agregam valor e fornecem uma série de saídas específicas para clientes externos e internos"*.

O Processo de Engenharia de Requisitos, segundo Sommerville e Sawyer (1997), é um conjunto estruturado de atividades para extrair, validar e manter um documento de requisitos. Uma completa descrição do processo poderá incluir quais atividades são realizadas, a estruturação ou particionamento, quem é responsável, as entradas e saídas desta atividade e as ferramentas usadas para suportar a engenharia de requisitos.

Tabela 2.1: Entradas e Saídas do Processo de Engenharia de Requisitos. (Kotonya e Sommerville, 1998)

ENTRADA	DESCRIÇÃO
Informações dos sistemas existentes	Informação sobre a funcionalidade dos sistemas a serem substituídos ou de outros sistemas que interagem com o sistema que está sendo especificado.
Necessidades dos stakeholders	Descrição do que os <i>stakeholders</i> necessitam do sistema para suportar seus trabalhos.
Padrões organizacionais	Padrões usados na organização relativos às práticas de desenvolvimento de sistemas, gerenciamento da qualidade, etc.
Normas e regulamentos	Regulamentações externas tais como regulamentações de saúde e segurança que se aplicam ao sistema.
Informações do domínio	Informações gerais sobre o domínio da aplicação do sistema.
SAÍDA	DESCRIÇÃO
Requisitos definidos	Uma descrição dos requisitos do sistema que seja entendida pelos <i>stakeholders</i> e que tenha sido acordado por estes.
Especificação do sistema	Esta é uma especificação mais detalhada da funcionalidade do sistema.
Modelos do sistema	Um conjunto de modelos, tais como, modelo de fluxo de dados, modelo de objeto, modelo de processo, etc., que descreve o sistema sob diferentes perspectivas.

Kotonya e Sommerville (1998) descrevem as principais entradas e saídas do processo de Engenharia de Requisitos na Tabela 2.1. Com o objetivo de entender melhor o processo de Engenharia de Requisitos, recorre-se à Figura 2.1.

Pode-se notar na Figura 2.1 que os autores consideram como relevante não só buscar informações sobre a funcionalidade dos sistemas ou descrever as necessidades dos usuários do sistema a ser desenvolvido, mas entender melhor o domínio da aplicação, considerar os aspectos relativos à própria estrutura da organização (padrões adotados, gerenciamento de qualidade, metas, estratégias, etc.) e inclusive considerar os aspectos legais (regulamentações externas) que se apliquem ao sistema a ser produzido. Desse ponto de vista, não cabe só ao administrador modelar os requisitos organizacionais, mas também ao desenvolvedor do sistema a ser adotado pela organização.

A Engenharia de Requisitos é um processo que envolve todas essas atividades para criar e manter o documento de requisitos do sistema. O Processo de Engenharia de Requisitos possui basicamente as seguintes atividades:

- a) **Elicitação de requisitos:** requisitos são descobertos através da consulta com as partes interessadas;
- b) **Análise e negociação de requisitos:** requisitos são analisados e os conflitos resolvidos através de negociação;
- c) **Documentação e modelagem de requisitos:** um documento de requisitos é produzido;
- d) **Validação de requisitos:** é checada a consistência do documento de requisitos;
- e) **Gerenciamento de requisitos:** envolve a coleta, armazenamento e manutenção de grande quantidade de informação.

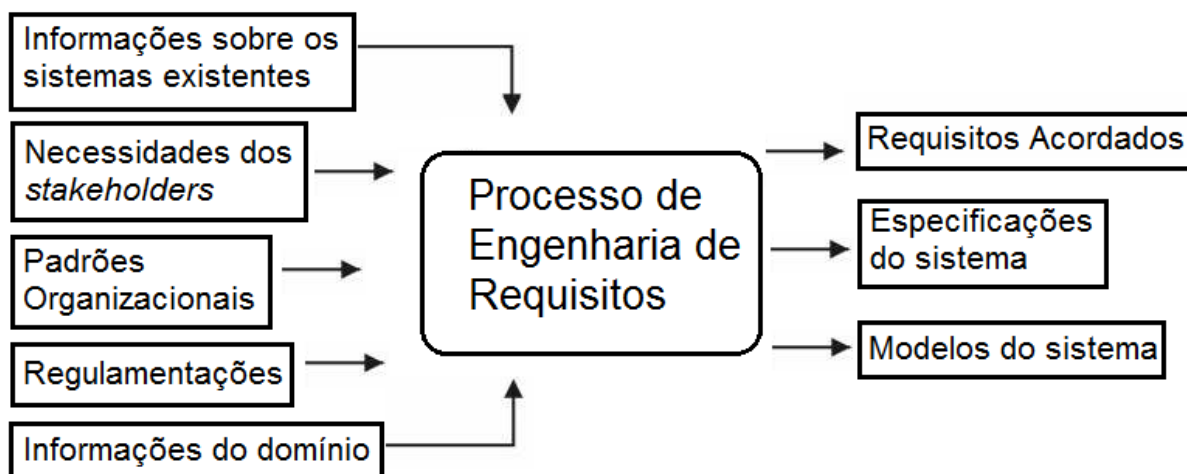


Figura 2.1: Entradas e Saídas do Processo de Engenharia de Software. (Kotonya e Sommerville, 1998)

2.1.3 Problemas na área de Engenharia de Requisitos

Apesar de todos os esforços da Engenharia de Requisitos, o que se vê atualmente são usuários insatisfeitos com softwares que não se adéquam às suas reais necessidades, o que resulta no aumento dos custos em correções de problemas associados a requisitos mal especificados. Tendo em vista esse panorama, Castro, Kolp e Mylopoulos (2002) citam os seguintes problemas na área:

- a) **Aspectos organizacionais não são considerados:** a maior preocupação da ER é em torno da obtenção das características técnicas referentes ao sistema que se deseja especificar. Deixando de lado os aspectos relacionados com as organizações, deixa-se de compreender amplamente as reais necessidades e intenções dessas organizações, resultando em requisitos que não satisfazem as reais necessidades do cliente. Santander (2002) afirma que isso implica em avaliar objetivos e metas organizacionais e apontar de que forma sistemas de software podem satisfazer estes objetivos;
- b) **Pouco envolvimento das partes participantes:** apesar dos esforços da ER na obtenção das necessidades dos clientes, quando não há envolvimento completo e total dos participantes no contexto de desenvolvimento de um sistema, as chances dos requisitos serem mal formulados e especificados são grandes;
- c) **Problemas de comunicação com as partes participantes:** quando o número de pessoas envolvidas é elevado, as partes podem não se comunicar eficientemente. Dessa forma, o resultado é a elaboração de um documento de requisitos que não é compreendido por todos, além da possibilidade de haver requisitos incompletos ou mesmo incorretos.

O presente trabalho está relacionado com a deficiência apresentada no item (a), onde vários trabalhos na área já foram propostos. Santander (2002) mostra a viabilidade e as vantagens de integrar a técnica de modelagem organizacional i^* e com uma técnica baseada em cenários, os Casos de Uso em UML (*Unified Modeling Language*). Tipicamente, Diagramas de Casos de Uso têm sido usados para capturar requisitos funcionais do sistema a ser desenvolvido. Infelizmente, a UML e técnicas baseadas em cenários em geral não estão equipadas para modelar os requisitos organizacionais. Através da proposta em (Santander, 2002), engenheiros de requisitos podem desenvolver Diagramas de Caso de Uso em UML a partir dos modelos organizacionais propostos na técnica i^* .

Já Alencar (1999), propõe diretrizes para o mapeamento dos modelos organizacionais, que explicitam os requisitos organizacionais, em componentes estruturais das linguagens de especificação precisas. Dessa forma, o objetivo é acrescentar às especificações precisas os componentes organizacionais que lhes faltem. Na proposta, a autora utiliza a técnica de modelagem *i**, para levantar os aspectos organizacionais, e as linguagens *Structured MAL* (*Structured Modal Action Logic*) e *OCL* (*Object Constraint Language*) associada com UML, para a modelagem precisa.

No trabalho de Teixeira (2009), é apresentado um conjunto de diretrizes que auxiliam o processo de construção de modelos organizacionais *i** para a fase de requisitos detalhados derivando-os a partir de modelos de atividades usando a Teoria da Atividade (TA). Considera-se como base a proposta apresentada em Cruz Neto (2008), direcionando o foco para a especificação dos requisitos do sistema computacional pretendido e seu impacto no ambiente organizacional.

No estudo de Estrada *et al.* (2001), é ressaltado a importância de incorporar informações sobre os processos de negócio na especificação de requisitos, onde o software é parte ativa deste modelo de negócio. Estrada *et al.* (2001) apresentam uma abordagem para criação de modelos de fluxos de trabalho (como ponto de partida para o desenvolvimento de modelos organizacionais *i**) a partir de um modelo de requisitos, o qual permite descrever e modificar um processo de negócio sem considerar os detalhes da implementação. O autor cita que vários trabalhos têm tentado utilizar técnicas de modelagem conceitual de sistemas de software, tais como a UML, para modelar processos de negócios, o que considera uma abordagem ineficiente, visto que essas técnicas não permitem especificar de forma clara relações organizacionais entre seus atores.

No entanto, os trabalhos mencionados estão focados em determinadas técnicas de modelagem organizacional, não se preocupando em realizar um estudo comparativo entre as técnicas existentes e, a partir dessa análise, propor uma solução melhorada. Assim, acredita-se que a presente avaliação possa auxiliar nas atividades de elicitação, análise e documentação de requisitos, visto que engenheiros de requisitos poderão compreender melhor o ambiente organizacional e seus relacionamentos, bem como considerar os requisitos e objetivos organizacionais previamente estabelecidos.

Nota-se que a motivação para pesquisas na área de Engenharia de Requisitos tem crescido consideravelmente nos últimos anos. A causa disso é o fato de que a maioria dos problemas

atribuídos a sistemas computacionais que não satisfazem clientes adequadamente são originados nas atividades realizadas nesta fase.

2.2 Engenharia de Software Experimental

A experimentação é uma disciplina muito importante para diversas áreas de pesquisa, que inclui a Engenharia de Software. Há uma compreensão cada vez maior na comunidade de ES que os estudos empíricos são necessários para avaliar, desenvolver ou melhorar processos, métodos e ferramentas para desenvolvimento de software e manutenção dos mesmos.

Os experimentos são o centro do processo científico, pois somente os experimentos podem validar as teorias ou explorar os fatores críticos para que as teorias possam ser formuladas e corrigidas. Novos métodos, técnicas, linguagens e ferramentas não deveriam ser apresentadas para venda sem experimentação e validação. É necessário que as novas invenções sejam avaliadas em comparação com as existentes. É importante ressaltar que experimentos não provam nada, nenhum experimento oferece prova com certeza, eles apenas verificam a previsão da teoria junto à realidade (Travassos, Gurov e Amaral, 2002).

Segundo Travassos, Gurov e Amaral (2002), a Engenharia de Software possui aspectos explícitos de produção, pois considera o processo de criação do software. Por outro lado, características relacionadas à competição e ao *time-to-market* exigem a melhoria contínua da qualidade do processo e do produto, apresentando a parte científica da Engenharia de Software.

Wohlin *et al.* (2000) afirmam que existem quatro métodos relevantes para a condução de experimentos na área da Engenharia de Software: o método científico, o método de engenharia, o método experimental e o método analítico.

Travassos, Gurov e Amaral (2002) supõem que a abordagem mais aceita para a experimentação na Engenharia de Software seja o método experimental, que considera a proposição e avaliação do modelo com os estudos experimentais. O método experimental, que será utilizado nesse trabalho, sugere o modelo, após desenvolve o método qualitativo e/ou quantitativo, aplica um experimento, mede e analisa, avalia o modelo e repete o processo. Dessa forma, essa abordagem é considerada orientada à melhoria revolucionária. Os autores citam que os outros métodos podem ser utilizados para a resolução de problemas na Engenharia de Software.

Informações mais completas a respeito da Engenharia de Software Experimental podem ser encontradas nos trabalhos (Wohlin *et al.*, 2000), (Basili, Selby e Hutchens, 1986), (Basili, 1992), (Fenton, 1993), (Basili *et al.*, 2006), (Basili, 1996).

2.2.1 Objetivos da Experimentação

Conradi *et al.* (2001) listam algumas definições dos objetivos da experimentação em Engenharia de Software. São elas:

- a) Para compreender a natureza dos processos da informação, os pesquisadores devem observar o fenômeno, encontrar explicação, formular a teoria, e verificá-la;
- b) A experimentação pode ajudar a construir uma base de conhecimento confiável, reduzindo assim incertezas sobre quais teorias, ferramentas e metodologias são adequadas;
- c) A observação e experimentação podem levar a novos e úteis meios de introspecção, e abrir novas áreas de investigação. A experimentação pode encontrar novas áreas onde a engenharia age lentamente;
- d) A experimentação pode acelerar o processo eliminando abordagens inúteis e suposições errôneas. A experimentação ajuda também a orientar a engenharia e a teoria nas direções promissoras de pesquisa;
- e) Os experimentos podem ser custosos, mas um experimento significativo geralmente pode se encaixar no orçamento de um pequeno laboratório. Por outro lado, um experimento caro pode valer a pena muito mais do que seu custo e, por exemplo, oferecer à companhia liderança de três, quatro ou cinco anos sobre a competição;
- f) O crescimento do número de trabalhos científicos com uma validação empírica significativa possui a boa chance de acelerar o processo de formação da Engenharia de Software como ciência. As ideias duvidosas serão rejeitadas mais rapidamente e os pesquisadores poderão concentrar-se nas abordagens promissoras;
- g) A tecnologia vem se modificando rapidamente. As mudanças sempre trazem ou eliminam as suposições. Os pesquisadores devem então antecipar as mudanças nas suposições e aplicar os experimentos para explorar as consequências dessas mudanças.

Já Travassos, Gurov e Amaral (2002) dizem que os principais objetivos para a execução de experimentos na área de Engenharia de Software, com respeito a produtos, processos, recursos, modelos, teorias, entre outros, são:

- a) **Caracterização:** “o que está acontecendo?”;
- b) **Avaliação:** “quão bom é isto?”;
- c) **Previsão:** “posso estimar algo no futuro?”;
- d) **Controle:** “posso manipular o evento?”;
- e) **Melhoria:** “posso melhorar o evento?”.

Os principais tipos de experimentos são descritos na próxima seção, no intuito de estudar qual é mais aplicável em determinada situação experimental.

2.2.2 Tipos de Experimentos

Existem atualmente inúmeros tipos de classificação dos experimentos. Acredita-se que esse grande número é devido ao fato de que a experimentação ainda é uma abordagem nova na área de Engenharia de Software. Um tipo de experimento é mais apropriado para determinada situação de acordo com, por exemplo, os objetivos do estudo, ou os resultados finais esperados.

A princípio, destacam-se três estratégias experimentais, as quais podem ser diferenciadas de acordo com o controle de execução, o controle de medição, o custo de investigação e a facilidade de repetição. São elas: *survey*, estudo de caso e experimento. Uma comparação entre as estratégias se encontra na Tabela 2.2.

Tabela 2.2: Comparação das estratégias experimentais. (Travassos, Gurov e Amaral, 2002)

Fator	Survey	Estudo de caso	Experimento
O controle da execução	Nenhum	Nenhum	Tem
O controle da medição	Nenhum	Tem	Tem
O controle da investigação	Baixo	Médio	Alto
Facilidade da repetição	Alta	Baixa	Alto
Custo	Baixo	Médio	Alto

O *survey* é uma pesquisa conduzida quando algumas técnicas ou ferramentas já tenham sido utilizadas. O principal meio para coletar as informações, sejam elas qualitativas ou quantitativas, é o questionário. Essa estratégia experimental possui os seguintes objetivos: descritivo, explanatório e explorativo.

O experimento normalmente é realizado em laboratório, oferecendo o maior nível de controle. O principal objetivo dessa estratégia é manipular uma ou mais variáveis e manter as outras fixas, medindo o efeito do resultado. Geralmente os experimentos são utilizados para

confirmar teorias ou validar medidas. Esses experimentos podem ser *in-vitro* (sob condições de laboratório) ou *in-vivo* (sob condições normais).

Já o estudo de caso, que será o tipo de experimento utilizado na proposta do trabalho, é empregado para monitorar os projetos, atividades e atribuições. Essa estratégia tem como objetivo observar um atributo específico e estabelecer o relacionamento entre atributos diferentes. O nível de controle em um estudo de caso é baixo, porém, de maneira contrária ao *survey*, o estudo de caso possui o controle sobre a medição das variáveis do experimento.

Na seção a seguir, será estudada uma metodologia de experimentação, o Paradigma da Melhoria da Qualidade (QIP), e uma abordagem relacionada, chamada *Goal/Question/Metric* (GQM).

2.2.3 Metodologia e Experimentação

Com o propósito de que um experimento forneça resultados válidos, o mesmo deve ser organizado e controlado ou, pelo menos, acompanhado. Um bom exemplo de metodologia de experimentação avançada é o Paradigma da Melhoria da Qualidade (*Quality Improvement Paradigm* – QIP) (Basili, Caldeira e Rombach, 1994). Essa metodologia possui seis passos, resultando em um ciclo da melhoria do processo completo. O ponto forte da metodologia está na melhoria contínua do processo de desenvolvimento de software. Os seis passos são descritos a seguir:

- a) 1º Passo: caracterização do processo de negócio necessária para a compreensão do ambiente e a definição dos objetivos básicos;
- b) 2º Passo: os objetivos quantitativos são estabelecidos com o propósito de demonstrar as expectativas da experimentação;
- c) 3º Passo: com base nos passos anteriores, o processo de melhoria apropriado é escolhido, tomando em consideração a consistência entre os objetivos;
- d) 4º Passo: o processo de desenvolvimento de software oferece o *feedback* do projeto que representa a informação recolhida;
- e) 5º Passo: a informação do passo anterior serve como base para análise, desde a avaliação das práticas atuais, a determinação de problemas, até a proposição da melhoria futura;
- f) 6º Passo: toda informação relevante está empacotada para utilização futura.

Uma abordagem ligada ao QIP e que será utilizada neste trabalho, é a *Goal/Question/Metric* (GQM) (Solingen e Berghout, 1999), que é uma abordagem orientada a metas e utilizada em Engenharia de Software para a medição de produtos e processos de software. O GQM é baseado no requisito de que toda a coleta dos dados deve ser baseada num fundamento lógico, em um objetivo ou meta, que é documentado explicitamente.

O paradigma GQM se apresenta como um mecanismo muito bom para planejamento, definição de objetivos da medição e avaliação. O objetivo do método GQM é caracterizar e fornecer um melhor entendimento dos processos, produtos, recursos e ambientes e, assim, estabelecer bases para comparação com trabalhos futuros (Basili, Caldeira e Rombach, 1994).

O GQM se baseia na suposição de que para se medir de maneira eficaz, alguns objetivos devem ser estabelecidos para que estes sirvam de rota para o estabelecimento de questões que orientarão a definição de métricas em um contexto particular. É muito importante para o sucesso da aplicação do GQM que os objetivos estejam bem traçados, pois somente assim a escolha das métricas e posterior avaliação dos dados serão bem sucedidas (Gomes, Oliveira e Rocha, 2001).

O primeiro passo nessa abordagem é definir metas (objetivos) a serem alcançadas no programa de medição. Após a identificação das metas, um plano GQM é elaborado para cada meta selecionada. O plano consiste, para cada meta, em um conjunto de questões quantificáveis que especificam as medidas adequadas para sua avaliação (Basili e Rombach, 1988). As questões identificam a informação necessária para atingir a meta e as medidas definem operacionalmente os dados a serem coletados para responder as perguntas. As metas GQM devem ser formuladas da forma apresentada na Tabela 2.3.

Tabela 2.3: Estrutura para definição das metas GQM. (Gresse e Hoisl, 1995)

Analisar o... →	<i><objeto de estudo></i>	Identifica o que será analisado. Ex: processo de software, projeto, documento, sistema, etc.
com a finalidade de... →	<i><objetivo></i>	Porque o objeto será analisado. Ex: avaliar, melhorar, monitorar, controlar, etc.
com respeito a... →	<i><enfoque></i>	Identifica o atributo que será analisado. Ex: confiabilidade, custos, correção, etc.
do ponto de vista de... →	<i><ponto de vista></i>	Identifica quem utilizará as métricas coletadas. Ex: equipe de desenvolvimento, gerente de projeto, etc.
no contexto de... →	<i><contexto></i>	Identifica o ambiente onde o programa de medição está localizado. Ex: projeto A, departamento X.

Sendo o GQM orientado a objetivos ou metas, pode-se enumerar seus passos nos três níveis de realização:

- a) **Conceitual:** definição do escopo da avaliação, ou seja, do objeto a ser medido (processos, produtos ou recursos);
- b) **Operacional:** definição de um conjunto de questões que auxiliem na caracterização do objeto de estudo e como ele deve ser enxergado dentro do contexto da qualidade;
- c) **Quantitativo:** definição de um conjunto de dados a serem obtidos, relacionado a cada uma das questões definidas anteriormente, a fim de respondê-las de forma quantitativa, ou seja, as métricas.

Dessa forma, Solingen e Berghout (1999) definem as fases do método GQM da seguinte forma (Figura 2.2):

- a) **Planejamento** - nesta fase são realizadas as seguintes atividades: relacionar a equipe que participará do GQM, selecionar a área que se deseja melhorar, apontar os projetos que farão parte da aplicação do método e treinamento da equipe nos conceitos necessários para a aplicação do GQM;
- b) **Definição** - nesta fase devem-se definir os objetivos do GQM, produzir ou adaptar modelos de software, definir as questões a serem respondidas, definir e refinar as métricas, além de promover a revisão dos planos do GQM;
- c) **Coleta de Dados** - nesta fase os dados são coletados, com base nas métricas definidas;
- d) **Interpretação** - os dados coletados anteriormente são absorvidos e conclusões acerca dos mesmos são extraídas pela equipe de GQM. Com base neles as questões definidas podem ser respondidas.

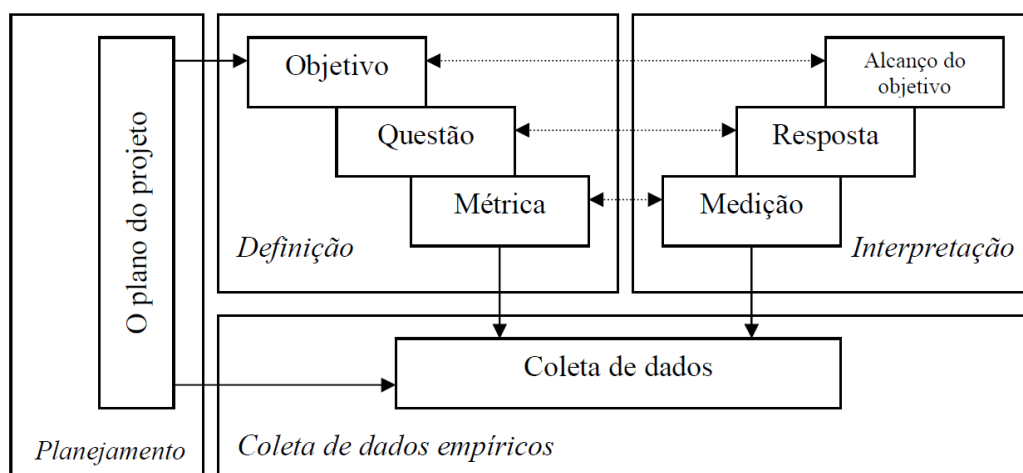


Figura 2.2: O processo principal da abordagem GQM. (Solingen e Berghout, 1999)

2.3 Considerações Finais

Neste capítulo, apresentou-se uma breve retrospectiva dos principais conceitos da Engenharia de Requisitos, subárea da Engenharia de Software, e foram introduzidas algumas definições a respeito da Engenharia de Software Experimental.

Primeiramente uma seção introdutória apresentou as principais definições de Engenharia de Requisitos, resumindo-se em uma disciplina que está inserida no contexto da Engenharia de Software e está relacionada com o estudo da elicitação, análise, documentação, validação e gerenciamento de requisitos.

Logo após foi apresentada a Classificação de Requisitos, onde foram estudados os Requisitos Funcionais, Requisitos Não-Funcionais e Requisitos Organizacionais.

Em seguida, foi apresentado o Processo de Engenharia de Requisitos, citando as principais atividades envolvidas.

Posteriormente, foram apresentados alguns problemas atuais na área de Engenharia de Requisitos, tais como a falta de consideração dos aspectos organizacionais durante a fase de coleta e especificação de requisitos de software.

Com relação à Engenharia de Software Experimental, pode-se dizer que a experimentação é uma disciplina muito importante para a Engenharia de Software, havendo uma compreensão cada vez maior na comunidade de ES que os estudos empíricos são necessários para avaliar, desenvolver ou melhorar processos, métodos e ferramentas para desenvolvimento de software e manutenção dos mesmos.

De forma resumida, tem-se que os principais objetivos da experimentação, com respeito a produtos, processos, recursos, modelos, teorias, entre outros, são: caracterização, avaliação, previsão, controle e melhoria.

De acordo com a literatura, destacam-se três principais tipos de experimentos, os quais podem ser diferenciados de acordo com o controle de execução, o controle de medição, o custo de investigação e a facilidade de repetição. São eles: *survey*, estudo de caso (utilizado na presente proposta) e experimento.

A metodologia de experimentação GQM (*Goal/Question/Metric*), que será utilizada neste trabalho, é uma abordagem orientada a metas e utilizada em Engenharia de Software para a medição de produtos e processos de software. Essa abordagem apresenta um mecanismo muito bom para planejamento, definição de objetivos da medição e avaliação.

No próximo capítulo são apresentadas técnicas para a modelagem de requisitos organizacionais. A modelagem serve para entender melhor a organização e os seus requisitos e permitir a construção de sistema de software com melhor qualidade. Assim, serão descritas algumas das principais técnicas de modelagem organizacional, tais como: técnica EKD, notação BPMN, técnica i*, Fluxos de Trabalho (*workflows*), Regras de Negócio, *Maps*, Casos de Uso de Negócio. A compreensão das técnicas de modelagem organizacional apresentadas no Capítulo 3 é essencial para a elaboração da nossa proposta.

Capítulo 3

Modelagem Organizacional

O presente capítulo fornece uma visão geral a respeito da importância da análise, elicitação e modelagem de requisitos organizacionais no contexto de desenvolvimento de software. Esta visão geral é apresentada na Seção introdutória 3.1. A Seção 3.2 aborda a técnica EKD, apresentando suas principais características. Da mesma forma, a Seção 3.3 estuda a notação BPMN, seguida pela técnica *i**, na Seção 3.4. Na Seção 3.5, o assunto estudado é o modelo de Fluxos de Trabalho (*workflows*). Já na Seção 3.6, o foco é as Regras de Negócio, seguida pela técnica *Maps* (Seção 3.7) e os Casos de Uso de Negócio (Seção 3.8). Uma análise crítica das técnicas estudadas é realizada na Seção 3.9. Por fim, a Seção 3.10 é destinada às considerações finais.

3.1 Introdução

A importância da informação para as organizações é universalmente aceita, constituindo, senão o mais importante, pelo menos um dos recursos cuja gestão e aproveitamento estão diretamente relacionados com o sucesso desejado. É fato comprovado que os processos de negócios sempre estarão presentes sob a estrutura organizacional das empresas. A principal diferença entre uma empresa de sucesso e uma empresa com sérios problemas, na maioria dos casos, pode estar relacionada ao gerenciamento dos processos de negócios (Trova, 2004).

O objetivo principal de um sistema de informação é automatizar tarefas ou atividades de um processo de negócio, permitindo que os atores da organização atinjam os seus objetivos específicos e, conseqüentemente, as metas gerais da empresa. Esta é a razão pela qual o estudo do ambiente organizacional em que o software será introduzido tem sido reconhecido como parte fundamental da Engenharia de Requisitos (Anton, 1996).

Essa análise deve ser conduzida antes da aquisição do sistema, pois o resultado terá reflexo em todo o processo de implantação, tendo consequências no tempo de duração da implantação, na contratação de consultoria externa, nas customizações a serem realizadas, na profundidade da mudança, no treinamento dos usuários e, principalmente, no custo final do projeto (Trova, 2004).

Neste contexto, existem pesquisas que destacam a importância do uso das metas do negócio para impulsionar o desenvolvimento de um produto de software (Estrada *et al.*, 2002).

Muitas técnicas tradicionalmente aplicadas no desenvolvimento de software tratam de aspectos relacionados à funcionalidade do sistema, à descrição de atividades e entidades, às entradas que deverão ser transformadas e às saídas que deverão ser produzidas. Porém, não consideram aspectos mais amplos como: os objetivos da organização, as regras de negócio, as restrições, os aspectos não funcionais relacionados à qualidade, à confiabilidade e à usabilidade (Pádua, Cazarini e Inamasu, 2004).

De acordo com Alencar (1999), essas técnicas não ajudam, portanto, a buscar soluções alternativas para problemas da organização, não adicionam valor ao negócio e, na maioria das vezes, os processos manuais são automatizados sem nenhuma modificação. Esses modelos têm como objetivo a descrição de sistemas técnicos, em vez de fornecer descrições mais ricas sobre as organizações sócio humanas.

Dessa forma, o que se tem são técnicas capazes de obter requisitos funcionais de um sistema, de descrever atividades e entidades, porém, sem considerar aspectos importantes da organização, como os seus objetivos, suas estratégias, sua política, suas restrições, etc. Assim, faz-se necessário uma abordagem mais rica, que facilite os esforços da Engenharia de Requisitos para obter uma melhor compreensão sobre os relacionamentos da organização entre os vários atores do sistema, e entender as razões envolvidas nos processos de decisões.

O analista de sistemas tem habilidade para descrever a empresa em termos de estrutura de dados que esta utiliza e a organização das funções que realiza, mas tendem a negligenciar as restrições com as quais a organização opera (BRG, 1997). Os requisitos organizacionais não devem ser considerados como uma simples descrição da funcionalidade do sistema, pois tratam do domínio no qual o sistema está inserido e das restrições que podem existir no ambiente, no sistema e no desenvolvimento, diminuindo ambiguidades e incertezas.

Para garantir que os sistemas cumpram com sua finalidade, os desenvolvedores devem possuir uma compreensão mais aprofundada sobre a organização. O modelo organizacional é uma representação da estrutura, das atividades, dos processos, das informações, dos recursos, do pessoal, do comportamento, dos objetivos e das restrições das empresas comerciais, governamentais ou de outra natureza (Pádua, Cazarini e Inamasu, 2004).

Segundo Santander e Castro (2000), a ideia é subsidiar os engenheiros de requisitos com metas estratégicas de negócios na organização as quais devem ser analisadas e ponderadas no momento do desenvolvimento de um sistema computacional. Além disso, também permitir que engenheiros de requisitos, a partir de uma observação mais detalhada dos modelos organizacionais, possam optar pela melhor alternativa para o desenvolvimento do software.

Para Bubenko Jr. e Wangler (1993), um bom sistema de informação deve atender às necessidades do negócio, caso contrário, o sistema estará impedindo o desenvolvimento do negócio. Este é o motivo do início de um novo paradigma, que muda da engenharia da informação orientada à tecnologia e abordagens orientadas a objetos, para estruturas que focam em modelagem de regras do negócio.

Segundo Alencar (1999), o principal obstáculo à captura correta dos requisitos de um sistema tem sido a dificuldade em se obter uma compreensão mais aprofundada do domínio da aplicação. Neste contexto, a modelagem organizacional facilita a compreensão do ambiente empresarial e é reconhecida como atividade valiosa pela Engenharia de Requisitos. Para a autora, a modelagem organizacional possui o objetivo de:

- a) fornecer um objeto, que seja uma representação compartilhável e reusável da cadeia de fornecimento de informação e conhecimento;
- b) suportar tarefas da cadeia de fornecimento, pela habilitação de respostas a questionamentos, que não estão explicitamente representados no modelo;
- c) definir os objetos de maneira precisa, de forma que sejam consistentemente aplicados, por meio dos domínios e interpretados pelos usuários;
- d) suportar visualização do modelo, de forma intuitiva, simples e consistente.

Existem diversas técnicas de modelagem organizacional. Algumas delas são realizadas com múltiplas visões, analisando metas e objetivos da organização (Bubenko Jr. e Kirikova, 1994). A organização, segundo os autores, é representada por meio de modelos, que facilitam a realização de especificações de requisitos mais próximas à realidade da organização.

Nas seções seguintes, serão apresentadas algumas técnicas de modelagem organizacional, suas características, qualidades e restrições.

3.2 A Técnica EKD

Segundo Rolland, Nurcan e Grosz (2000), o EKD (*Enterprise Knowledge Development*) 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. Os autores enfatizam que 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.

O modelo organizacional contém um número de submodelos inter-relacionados. Cada um representa algum aspecto da organização. Os submodelos são: Modelo de Objetivos (MO), Modelo de Regras de Negócio (MRN), Modelo de Conceitos (MC), Modelo de Processos de Negócio (MPN), Modelo de Atores e Recursos (MAR), e Modelo de Requisitos e Componentes Técnicos (MRCT) (Bubenko Jr., Stirna e Brash, 1998).

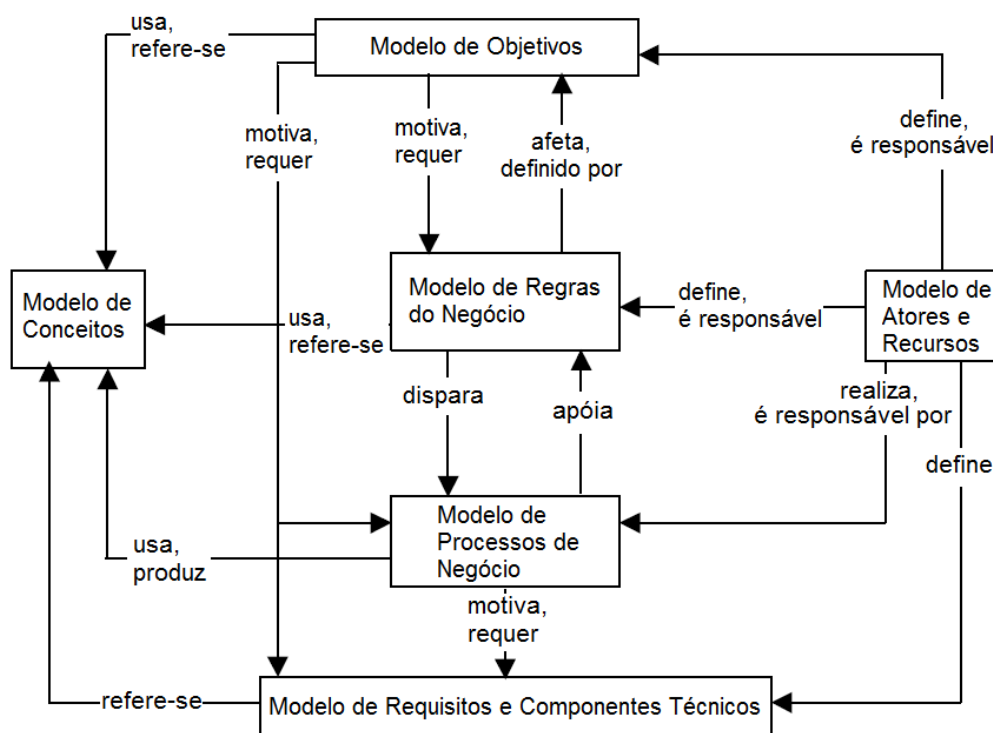


Figura 3.1: Submodelos que compõem o modelo organizacional. (Bubenko Jr., Stirna e Brash, 1998)

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) e com componentes de outros submodelos (relacionamento intermodelos). A Figura 3.1 mostra o relacionamento entre modelos. Cada um desses submodelos será descrito e exemplificado nas próximas seções.

3.2.1 O Modelo de Objetivos (MO)

O Modelo de Objetivos 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 à sua razão de existir. Os componentes do modelo de objetivos são:

- a) **Objetivos:** são usados para expressar intenções;
- b) **Restrições:** são restrições em componentes e ligações existentes na organização;
- c) **Problemas:** expressam que a organização está, ou pode ficar em um estado não desejado;
- d) **Causas:** expressam razões de problemas;
- e) **Oportunidades:** expressam possíveis situações que não são consideradas objetivos, mas que podem ser vantajosas;
- f) **Pontos Fracos:** são situações que não contribuem positivamente para a satisfação dos objetivos da organização.

As ligações dentro do Modelo de Objetivos podem ser:

- a) **Relacionamento de Apoio:** usado para refinar ou decompor objetivos e outros componentes;
- b) **Relacionamento de Impedimento:** usado para mostrar influências negativas entre componentes do Modelo de Objetivos, e pode ser considerado como o oposto do Relacionamento de Apoio;
- c) **Relacionamento de Conflito:** usado para definir situações que ao alcançar um objetivo haverá um conflito com outro.

A seguir, um exemplo de parte de um Modelo de Objetivos de uma Biblioteca (Figura 3.2).

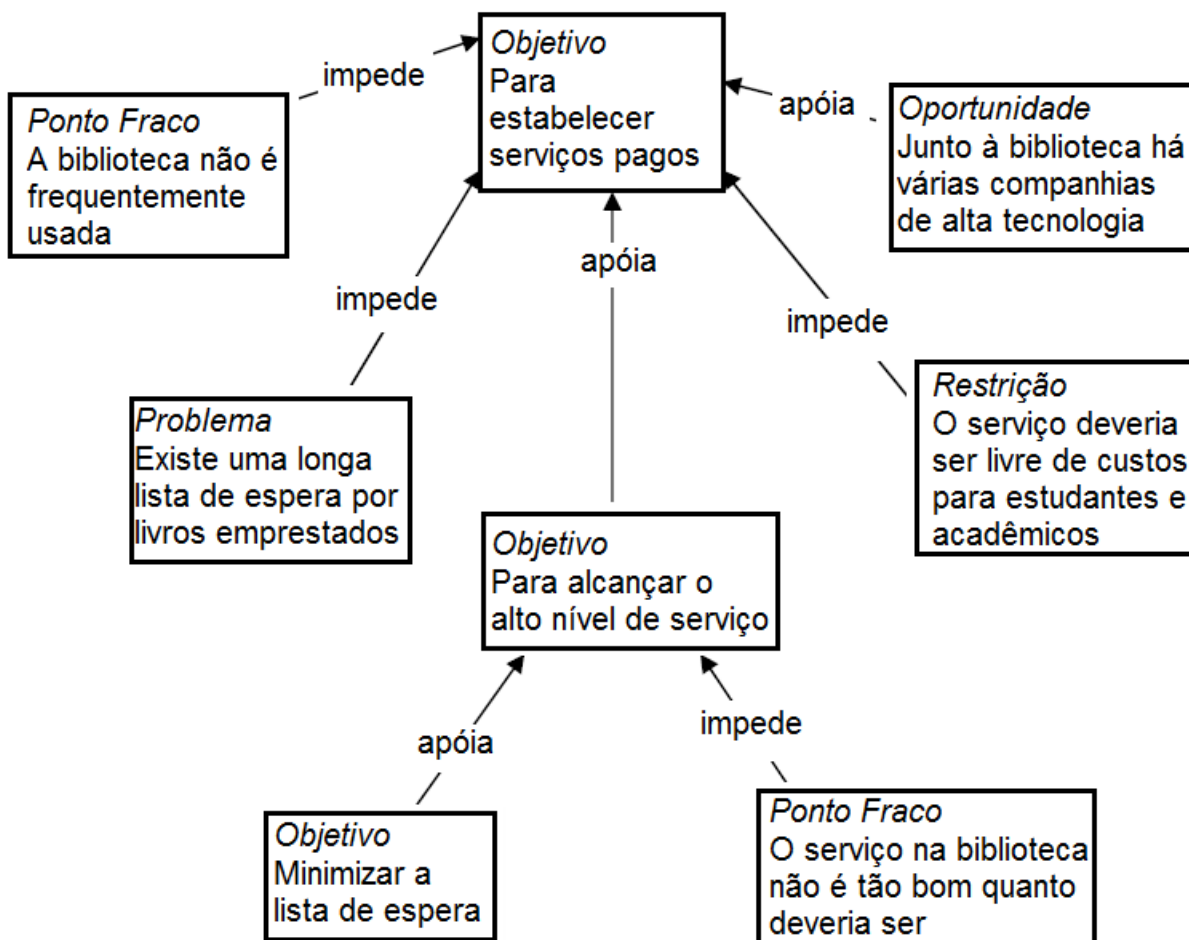


Figura 3.2: Parte de um MO de uma Biblioteca. (Bubenko Jr., Stirna e Brash, 1998)

3.2.2 O Modelo de Regras de Negócio (MRN)

O Modelo de Regras de Negócio, de acordo com Bubenko Jr., Stirna e Brash (1998), é usado para definir e manter explicitamente regras do negócio formuladas, consistentes com o Modelo de Objetivos. Regras do Negócio podem ser vistas como operacionalizações 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. Podem ser categorizadas em:

- a) **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.
- b) **Regras de Evento-Ação:** estão relacionadas com a invocação de atividades. Expressam condições sobre as quais as atividades devem ser realizadas;

- c) **Regras de Restrição:** estão relacionadas à integridade da informação, à estrutura dos componentes, ou às atividades e comportamentos permitidos na organização.

Os relacionamentos podem ser do tipo:

- a) **Relacionamento de apoio:** usado para refinar ou decompor regras;
 b) **Relacionamento de impedimento:** usado para mostrar influências negativas entre componentes do Modelo de Regras de Negócio.

A seguir, um exemplo de parte de um Modelo de Regras de Negócio de uma Biblioteca (Figura 3.3).

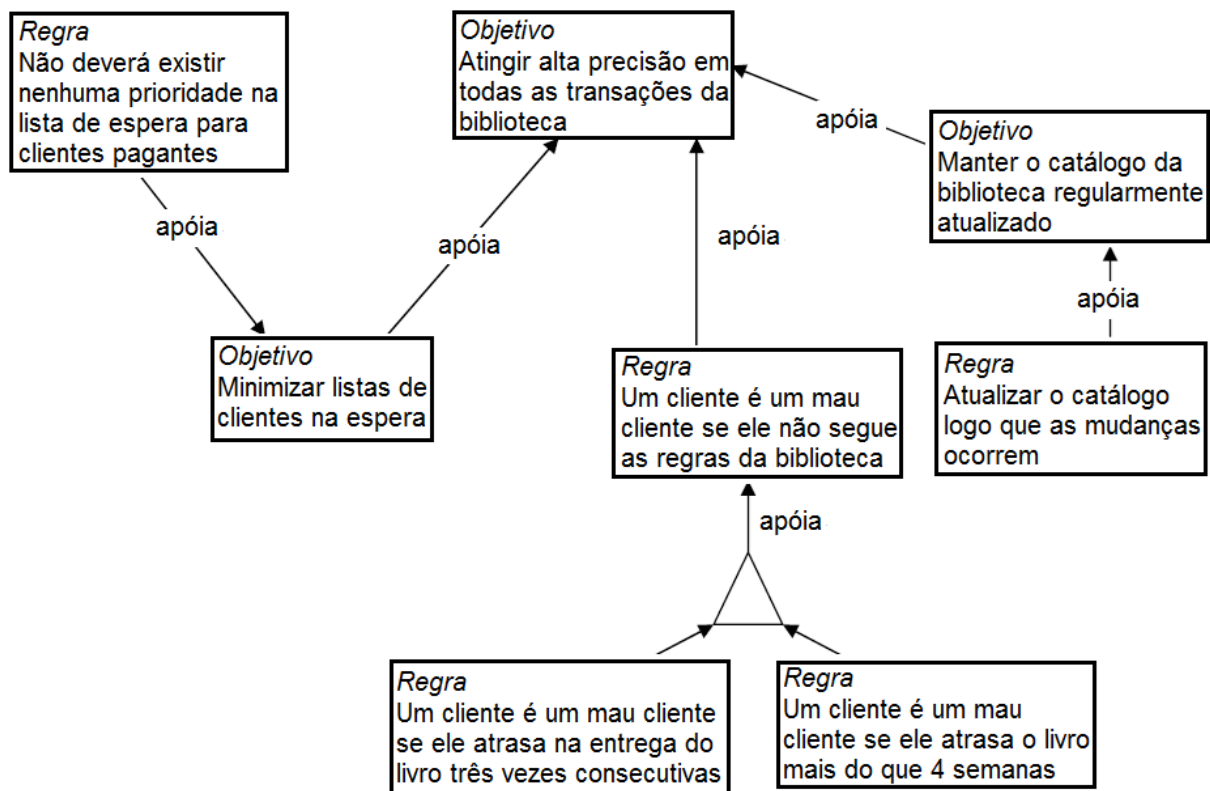


Figura 3.3: Parte de uma interação entre o MRN e componentes do MO. (Bubenko Jr., Stirna e Brash, 1998)

3.2.3 O Modelo de Conceitos (MC)

De acordo com Bubenko Jr., Stirna e Brash (1998), o Modelo de Conceitos é utilizado 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 para os *stakeholders*.

O Modelo de Conceitos inclui componentes tais como:

- a) **Entidade:** é alguma coisa do domínio de interesse e aplicação que deve ser entendida, caracterizada e definida usando relacionamento para outras entidades;
- b) **Atributo:** é uma entidade usada apenas para caracterizar outra entidade;
- c) **Grupos de Componentes de Modelo de Conceitos:** é uma visão de uma parte de um modelo de conceitos, e inclui um subconjunto de entidades do Modelo de Conceitos, relacionamentos e atributos.

Os relacionamentos podem ser do tipo:

- a) **Relacionamento binário:** é uma relação semântica entre duas entidades;
- b) **Relacionamento ISA:** expressa generalização;
- c) **Relacionamento Parte-de:** expressa agregação.

Um exemplo de Modelo de Conceitos será mostrado no Capítulo 4, quando o assunto for a comparação das técnicas de modelagem organizacional.

3.2.4 O Modelo de Processos de Negócio (MPN)

O Modelo de Processos de Negócio destina-se a analisar o processo e fluxo de informação e material da organização, onde processos podem ser decompostos em subprocessos. O Modelo de Processos de Negócio descreve as atividades organizacionais (funções e processos da organização). Os componentes do Modelo de Processos de Negócio são (Bubenko Jr., Stirna e Brash, 1998):

- a) **Processos:** uma coleção de atividades que consome entradas e produz saídas em termos de informação ou material;
- b) **Processos Externos:** coleção de atividades que são localizadas fora do escopo da área da atividade organizacional;
- c) **Conjunto de materiais ou informações:** é um conjunto de informação ou material enviado de um Processo ou Processo Externo para outro.

A seguir, um exemplo de processo não decomposto para verificação do endereço do cliente (Figura 3.4).

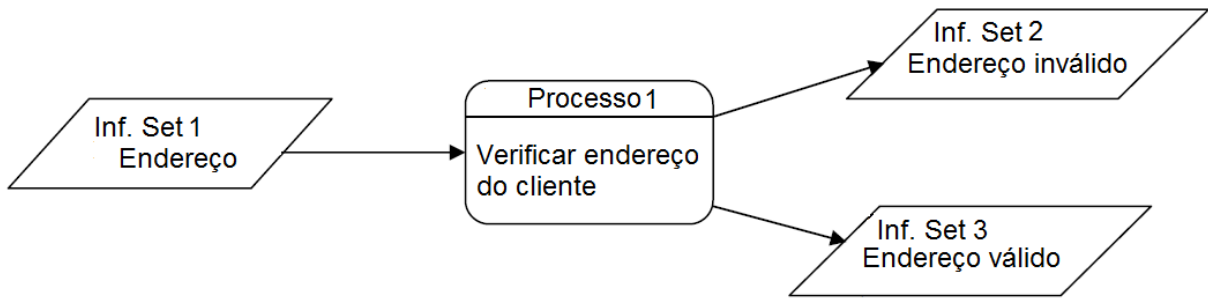


Figura 3.4: O processo de verificação do endereço do cliente não decomposto. (Bubenko Jr., Stirna e Brash, 1998)

O próximo exemplo decompõe o processo da Figura 3.4 em quatro subprocessos (Figura 3.5).

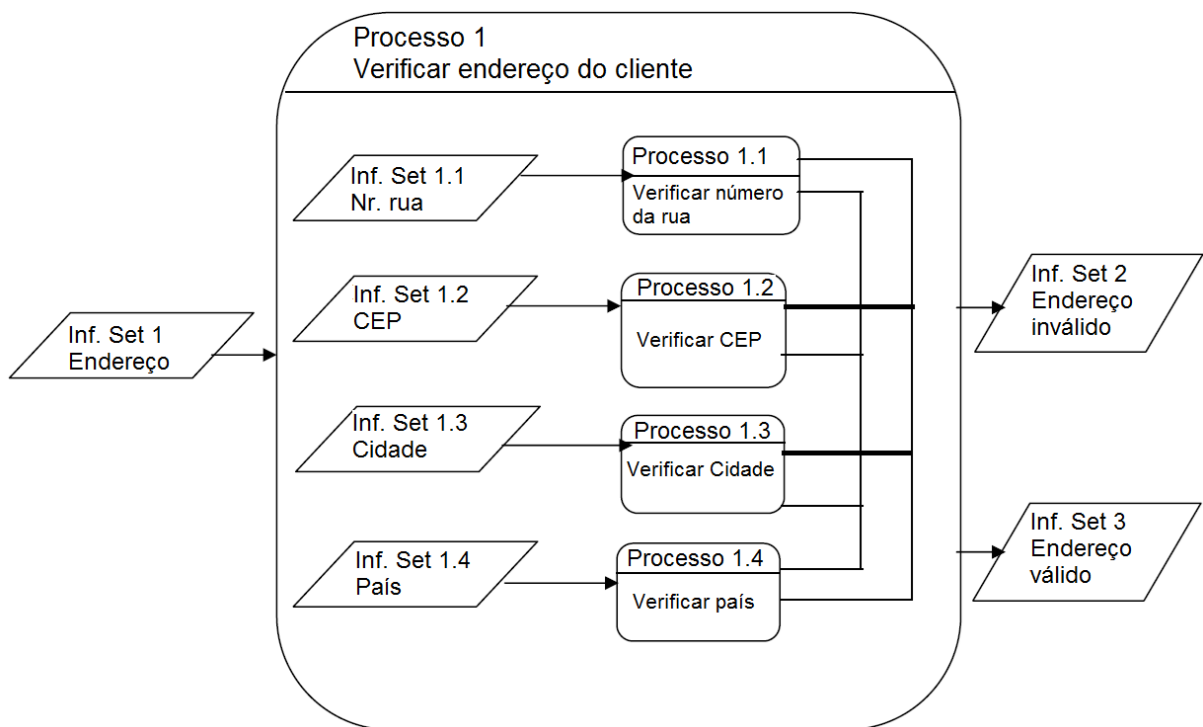


Figura 3.5: O processo de verificação do endereço do cliente decomposto. (Bubenko Jr., Stirna e Brash, 1998)

3.2.5 O Modelo de Atores e Recursos (MAR)

O Modelo de Atores e Recursos define que tipo de atores e recursos, ou quais atores individuais estão envolvidos nas atividades organizacionais. Esse modelo descreve: como diferentes atores e recursos são relacionados entre si e como são relacionados com

componentes do Modelo de Objetivos; de que forma os objetivos estão relacionados aos processos do Modelo de Processos de Negócio. Ator e Recurso podem ser:

- a) **Indivíduos:** são pessoas da organização;
- b) **Recursos não humanos:** são máquinas ou equipamentos;
- c) **Unidades Organizacionais:** representam divisões da organização, como: departamentos, projetos e grupos de trabalho;
- d) **Papel:** indivíduos e unidades organizacionais podem desempenhar papéis diferentes em contextos diferentes; recursos não humanos podem desempenhar um papel também.

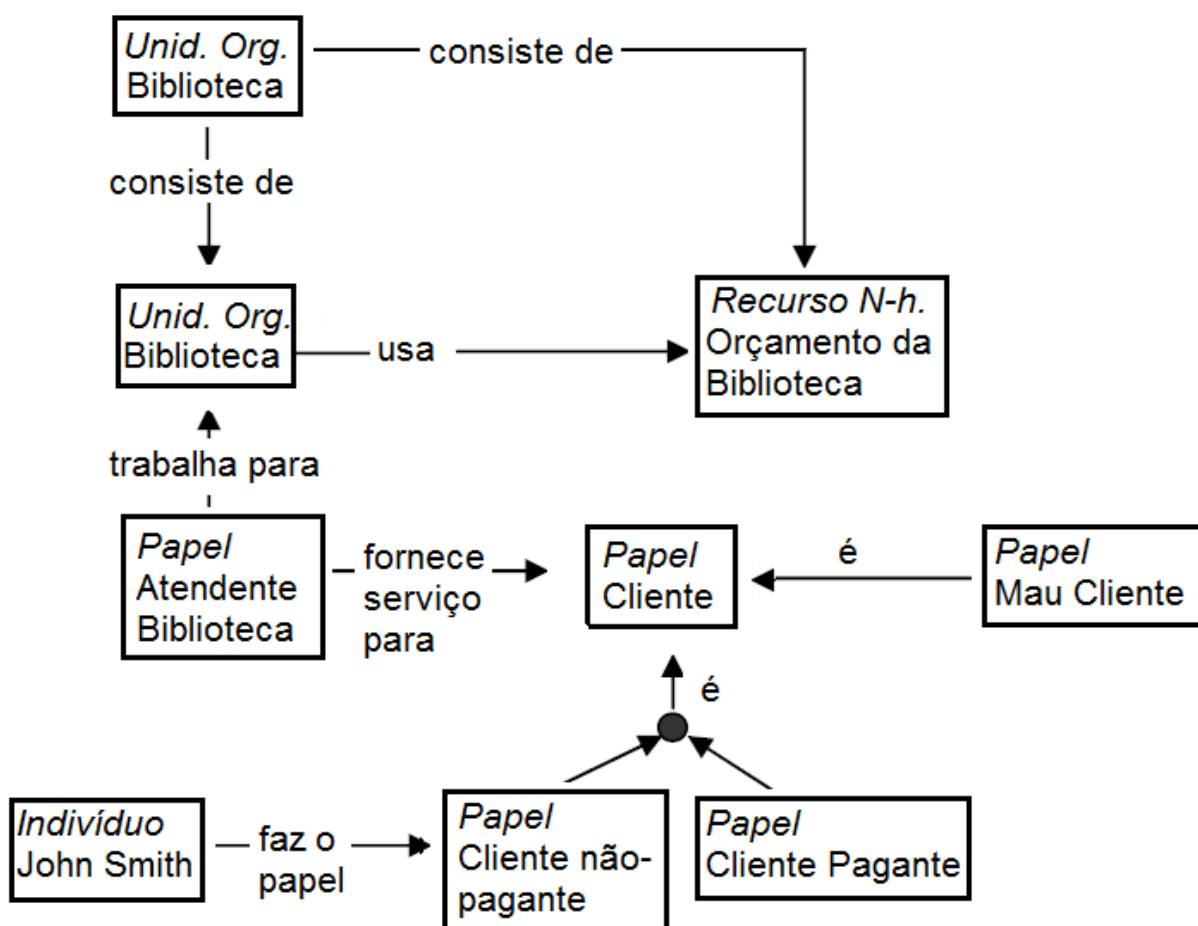


Figura 3.6: Exemplo de MAR de uma Biblioteca. (Bubenko Jr., Stirna e Brash, 1998)

Os relacionamentos podem ser do tipo:

- a) **Relação Binária:** expressa relação entre dois atores;
- b) **Relação ISA:** expressa generalização entre tarefas do Modelo de Atores e Recursos;
- c) **Relação Parte-de:** expressa relação de agregação.

Um exemplo de Modelo de Atores e Recursos de uma Biblioteca pode ser visualizado na Figura 3.6.

3.2.6 O Modelo de Requisitos e Componentes Técnicos (MRCT)

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 Processo do Negócio. Primeiramente, é necessário desenvolver um conjunto de requisitos ou objetivos de alto nível para o sistema computacional como um todo. Baseado nesse conjunto, o sistema de informação é estruturado em um número de subsistemas ou componentes técnicos. Para cada subsistema, 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.

De acordo com Bubenko Jr., Stirna e Brash (1998), os componentes presentes no Modelo de Requisitos e Componentes Técnicos são:

- a) **Objetivos do Sistema de Informação:** usados para expressar o alto nível de objetivos em relação ao sistema de informação e/ou subsistema ou componentes;
- b) **Problemas do Sistema de Informação:** usados para expressar estados não desejáveis do negócio ou do ambiente, ou fatos problemáticos;
- c) **Requisitos do Sistema de Informação:** expressam requisitos a serem designados para propriedades do sistema de informação, podendo ser funcional ou não funcional.

A seguir, um exemplo de parte de um Modelo de Requisitos e Componentes Técnicos de uma Biblioteca (Figura 3.7).

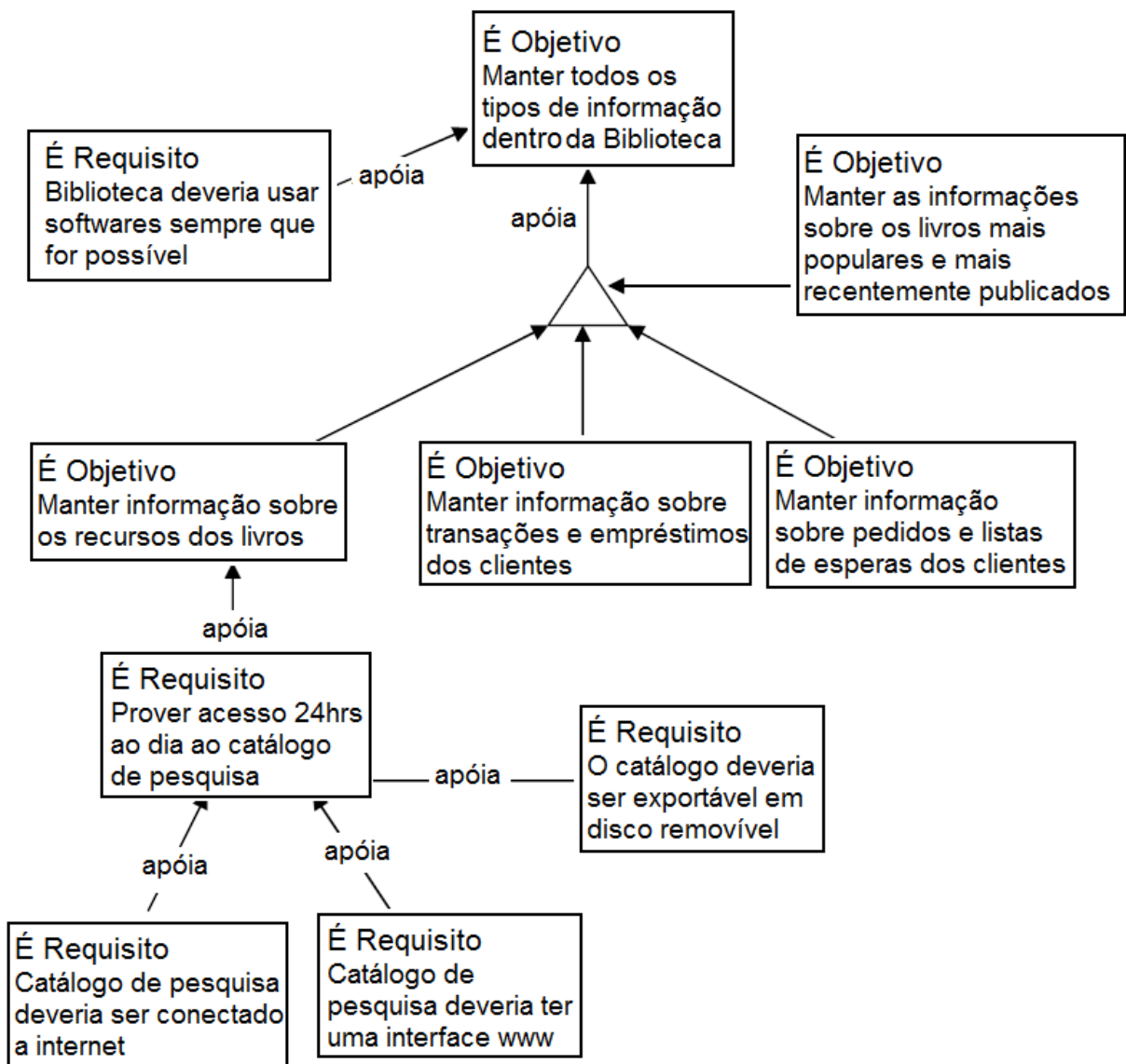


Figura 3.7: Exemplo de parte de um MRCT de uma Biblioteca. (Bubenko Jr., Stirna e Brash, 1998)

3.3 Business Process Modeling Notation (BPMN)

Modelos de processos de negócios desempenham um papel fundamental na gestão de organizações (Smith e Fingar, 2003), (Hammer e Champy, 1993) e nas empresas de desenvolvimento de sistemas de informação (Dumas, Aalst e Hofstede, 2005). Muitas notações desenvolvidas para a tarefa de modelagem de processos de negócio têm o seu próprio foco de aplicação e público adequado (Bider e Johannesson, 2002), (Kavakli e Loucopoulos, 1999).

Modelos conceituais de alto nível proporcionam a compreensão da organização a partir de uma perspectiva intencional e social para apoiar o raciocínio durante o desenvolvimento (Yu,

1995a). Em comparação, modelos técnicos de baixo nível são especialmente adequados para aplicações na descrição, execução e simulação de processos de negócios (Yu, 1995b).

Com o objetivo de criar padrões e uma arquitetura comum para gerenciamento de processos de negócio, foi criada a BPMI (*Business Process Management Initiative*) (BPMI, 2010), uma organização sem fins lucrativos, iniciada pela Intalio Inc. em 2000 e que recebeu o suporte de gigantes da indústria como a IBM, SAP, BEA, Fujitsu, WebMethods e IDS Scheer.

Em agosto de 2001, um grupo de trabalho (*Business Process Modeling Notation Working Group* - BPMN-WG) da BPMI.org foi formado por 35 empresas e iniciou os trabalhos para criar a BPMN, uma Notação de Modelagem de Processos de Negócio.

A versão 1.0 da especificação escrita por Stephen White da IBM surgiu em maio de 2004 e se estabeleceu como notação padrão para modelar processos executáveis de negócio. Em junho de 2005, a BPMI anunciou sua junção a OMG (*Object Management Group*), associação sem fins lucrativos que desde 1989 desenvolve e mantém padrões e especificações, dentre elas, a notação UML.

Segundo Koliadis *et al.* (2007), a BPMN é essencialmente uma técnica orientada a modelagem de processos de negócio que suporta o controle de execução da atividade de entidades de dentro de uma organização através de *swimlanes* (raias). Essa notação tem a capacidade de mapear diretamente as linguagens de execução de processo, incluindo XPDL (*XML Process Definition Language*) (Fischer, 2005) e BPEL (*Business Process Execution Language*), definidas em (White, 2004) e (Ouyang *et al.*, 2006).

O principal objetivo da BPMN é fornecer uma notação que seja facilmente compreensível por todos os usuários de negócios, desde os analistas de negócio até os clientes. Assim, a intenção da BPMN é padronizar a notação para modelagem de processos de negócios em face das muitas notações de modelagem existentes de diferentes pontos de vista.

A BPMN oferece um Diagrama de Processos de Negócio (*Business Process Diagram* - BPD), que é um diagrama desenvolvido para uso por pessoas que projetam e gerenciam processos de negócios.

3.3.1 Business Process Diagram (BPD)

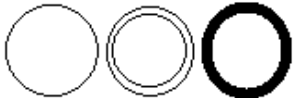
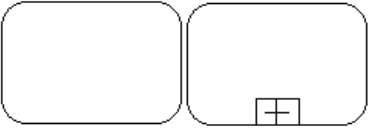
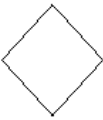
Deve ser enfatizado que um dos condutores para o desenvolvimento da BPMN é a criação de um mecanismo simples para a criação de modelos de processos de negócios, mas que ao

mesmo tempo seja capaz de lidar com a complexidade inerente aos processos de negócio. A abordagem para lidar com essas duas exigências conflitantes foi organizar os aspectos gráficos da notação em categorias específicas. Isto fornece um pequeno conjunto de categorias de notação, de modo que o leitor de BPD possa facilmente reconhecer os tipos básicos de elementos e compreender o diagrama (BPMN, 2010b).

A lista dos elementos gráficos em um BPD é apresentada em dois grupos. Primeiro, há a lista dos principais elementos que irão apoiar a exigência de uma notação simples. Estes são os elementos básicos que definem a BPMN. A maioria dos processos de negócio é modelada de forma adequada com esses elementos. Por outro lado, há toda a lista de elementos, incluindo os elementos do núcleo, que ajudam no desenvolvimento de uma notação mais poderosa, onde se lida com situações de modelagem mais avançadas. Neste trabalho são apresentadas apenas as categorias básicas de elementos, destacadas a seguir.

- **Objetos de Fluxo** (Tabela 3.1): são os principais elementos gráficos para definir o comportamento de um processo de negócio;

Tabela 3.1: Objetos de Fluxo. (BPMN, 2010b)

<i>Objeto</i>	<i>Descrição</i>	<i>Figura</i>
Evento	É algo que acontece durante um processo do negócio. Estes eventos afetam o fluxo do processo e tem geralmente uma causa (<i>trigger</i>) ou um impacto (<i>result</i>). Há três tipos de eventos, baseados sobre quando afetam o fluxo: início (<i>start</i>), intermediário (<i>intermediate</i>) e fim (<i>end</i>).	
Atividade^{1,2}	É um termo genérico para um trabalho executado. Os tipos de atividades são: tarefa ¹ e subprocesso ² . O subprocesso é distinguido por uma pequena cruz no centro inferior da figura.	
Gateway	É usado para controlar a divergência e a convergência da sequência de um fluxo. Assim, determinará decisões tradicionais, como unir ou dividir trajetos.	

Nota 1 – Tarefa: a tarefa é a menor unidade de um processo, geralmente atômica (não pode ser dividida em mais objetos).

Nota 2 – Subprocesso: um subprocesso, dentro de um BPD, é como uma atividade composta por uma série de outras atividades, formando um novo fluxo; o subprocesso pode ser exibido de duas formas: “aberta” (Figura 3.8) ou “fechada” (Figura 3.9).

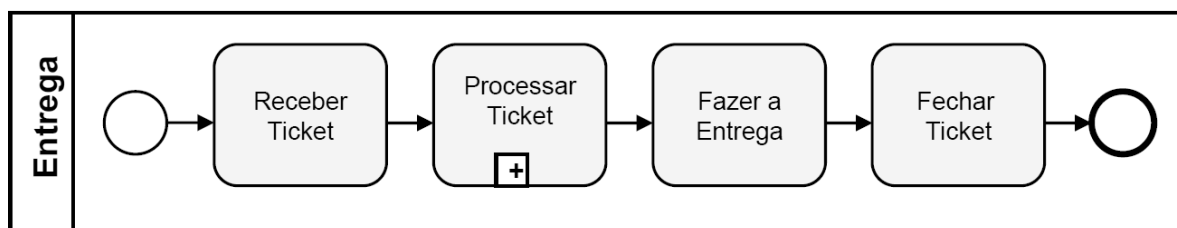


Figura 3.8: Exemplo de subprocesso de forma “fechada”. (Santos, 2010)

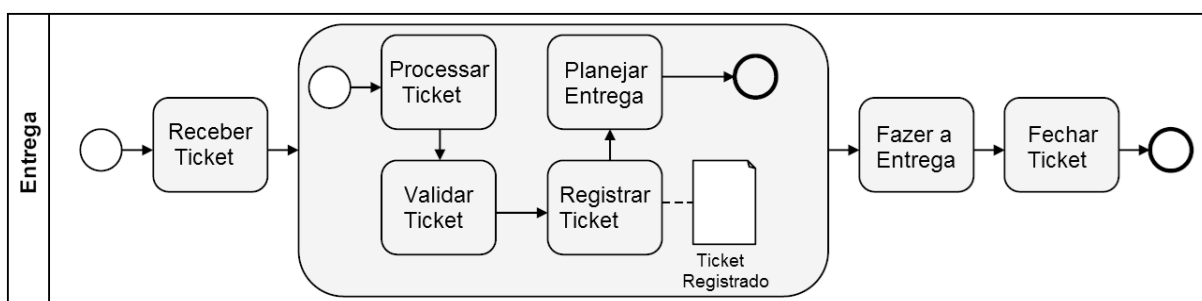


Figura 3.9: Exemplo de subprocesso de forma “aberta”. (Santos, 2010)

Observação: para representar um subprocesso aberto, foi utilizada a representação gráfica de uma atividade, contudo com o desenho do novo processo internamente. No caso de um subprocesso aberto, o desenho completo deverá estar sempre no mesmo *pool*.

- **Objetos de Conexão** (Tabela 3.2): são utilizados para conectar os objetos de fluxo entre si ou outras informações;

Tabela 3.2: Objetos de Conexão. (BPMN, 2010b)

<i>Objeto</i>	<i>Descrição</i>	<i>Figura</i>
Fluxo de Sequência	É usado para mostrar a ordem (sequência) com que as atividades serão executadas em um processo.	
Fluxo de Mensagem	É usado para mostrar o fluxo das mensagens entre dois participantes diferentes que os emitem e recebem.	
Associação	É usada para associar dados, texto e outros artefatos com os objetos de fluxo. As associações são usadas para mostrar as entradas e as saídas das atividades.	

- **Swimlanes (raias)** (Tabela 3.3): funcionam como um mecanismo de organização das atividades em categorias visuais separadas;

Tabela 3.3: *Swimlanes*. (BPMN, 2010b)

Objeto	Descrição	Figura
Pool	O <i>pool</i> representa um participante em um processo. Ele pode ser usado para representar uma unidade funcional. Exemplo: Vendas. Ele atua como um container gráfico para dividir um conjunto de atividades de outros <i>pools</i> , geralmente no contexto de situações de B2B.	
Lane	<i>Lane</i> é uma subdivisão dentro de um <i>pool</i> . Usado para organizar e categorizar as atividades.	

Pools são utilizados quando o diagrama envolve duas entidades de negócio ou participantes que estão separados fisicamente no diagrama. Especifica o "que faz o que" colocando os eventos e os processos em áreas protegidas, chamados de *pools*. A Figura 3.10 mostra um exemplo da utilização de *pools*.

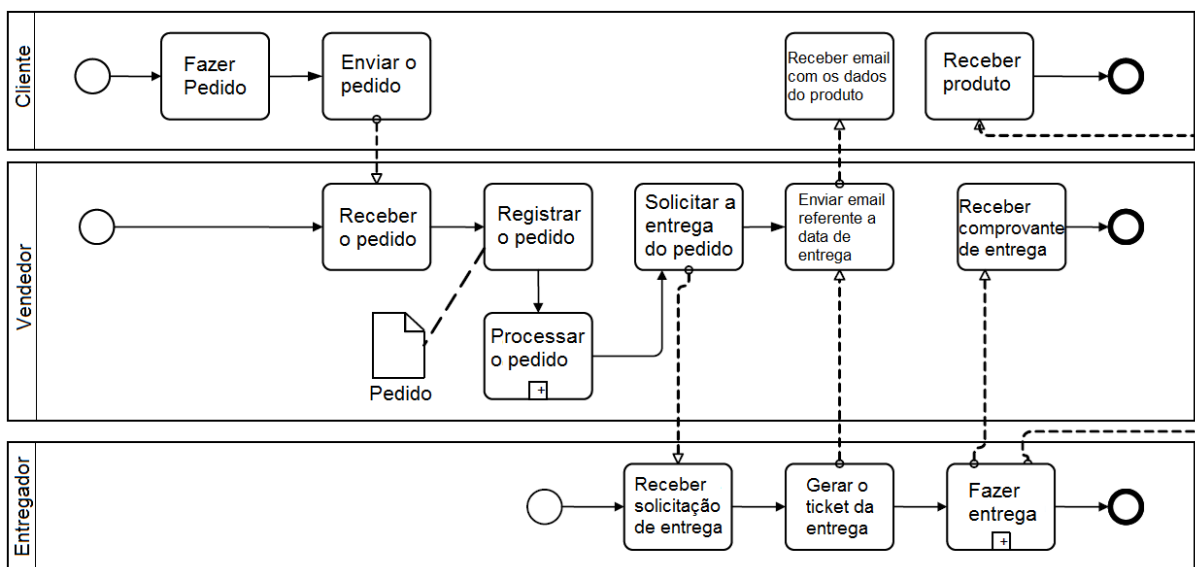


Figura 3.10: Exemplo de utilização de *pools*. (Santos, 2010)

Os objetos do tipo *lanes* são utilizados para separar as atividades associadas para uma função ou papel específico. Um *pool* representa uma organização e um *lane* representa tipicamente um departamento dentro dessa organização. A Figura 3.11 mostra um exemplo de utilização de *lanes*.

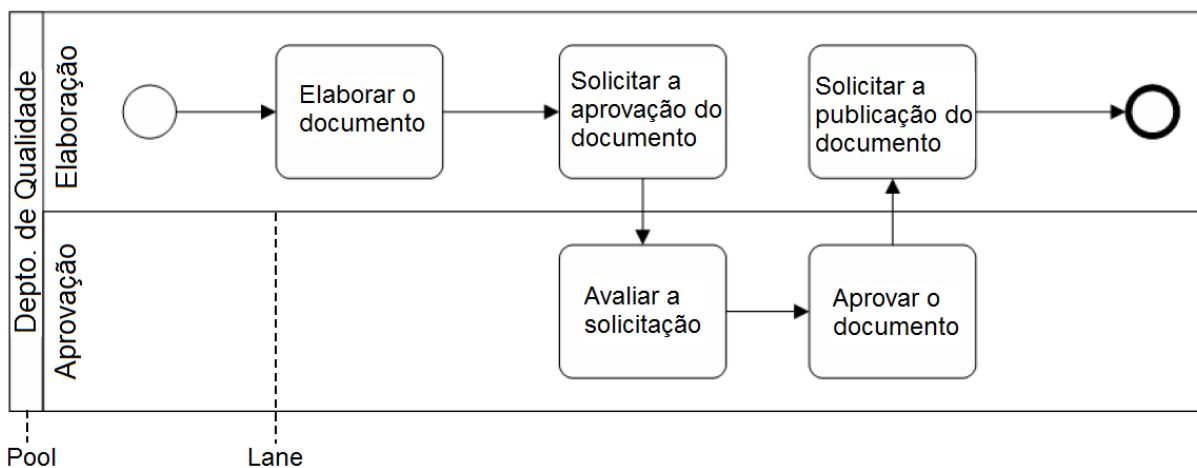


Figura 3.11: Exemplo de utilização de *lanes*. (Santos, 2010)

- **Artefatos** (Tabela 3.4): são usados para fornecer informações adicionais, ilustrando as entradas e as saídas das atividades no processo.

Tabela 3.4: Artefatos. (BPMN, 2010b)

<i>Objeto</i>	<i>Descrição</i>	<i>Figura</i>
Objetos de dados	Objeto de Dados é considerado como Artefato e não como Fluxo de Objeto. Ele não afeta o fluxo de mensagem e nem o fluxo de sequência de um processo, mas fornece informação sobre o que o processo faz. É utilizado para representar documentos, como: fatura, nota fiscal, ordem de serviço, requisição, email e etc.	
Grupo	Um grupo é representado por um retângulo. É utilizado para agrupamento de atividades e tarefas.	
Anotações	As anotações fornecem informações adicionais e comentários para o leitor de um diagrama BPMN.	

Um exemplo de segmento de processo utilizando artefatos pode ser visto na Figura 3.12.

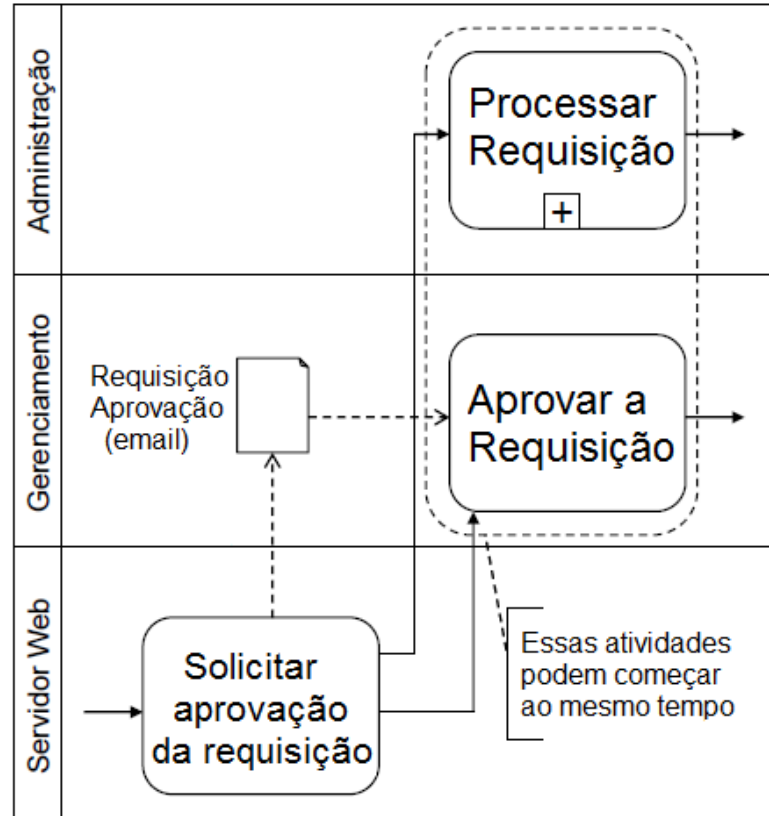


Figura 3.12: Exemplo de segmento de processo utilizando artefatos. (Santos, 2010)

3.3.2 Tipos de submodelos BPMN

Existem três tipos básicos de submodelos dentro de um modelo BPMN (2010b):

- a) **Processos Internos** (Figura 3.13): são os processos internos a uma organização específica. Definem atividades que não são geralmente visíveis ao público. O Fluxo de Sequência do processo é contido dentro do *pool*, e não pode cruzar os limites do *pool*. Fluxos de Mensagem podem cruzar a fronteira exterior para mostrar as interações que existem entre os diferentes processos de negócio internos;

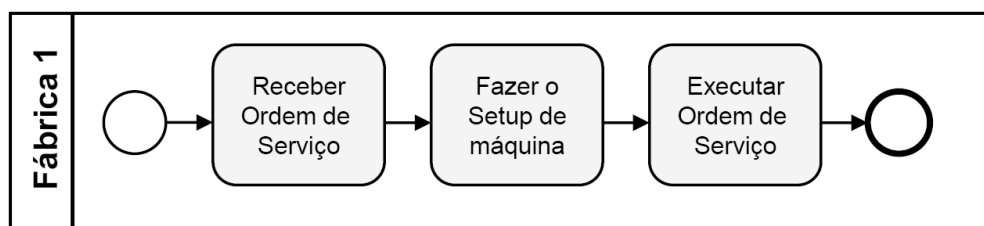


Figura 3.13: Exemplo de Processo Interno. (Santos, 2010)

- b) **Processos abstratos** (Figura 3.14): muitas vezes, o processo inclui atividades que são realizadas fora da empresa e não tem-se gerência sobre a execução destas atividades. Utiliza-se um modelo abstrato para representar uma “entidade” independente, com processos próprios, mas que não se pode modelar. No exemplo, o Fornecedor faz o beneficiamento da matéria prima, entretanto, é um processo interno do Fornecedor, o qual não é conhecido, ele deve ser modelado como um processo abstrato;

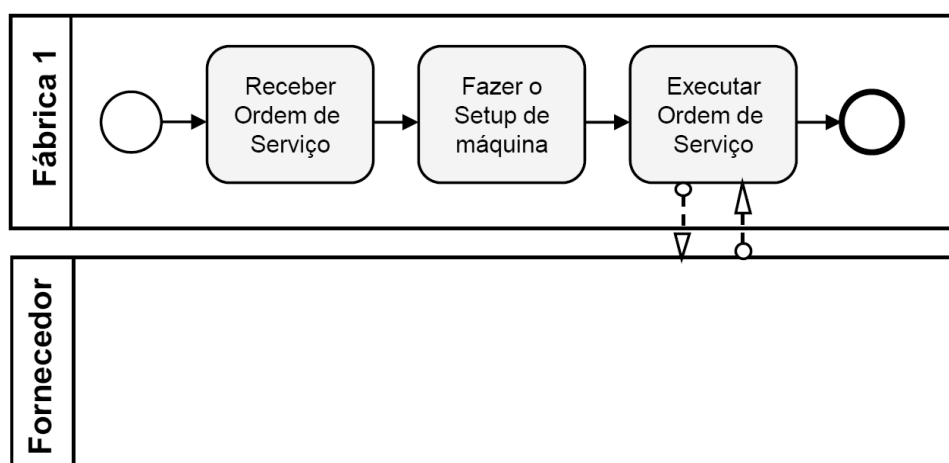


Figura 3.14: Exemplo de Processo Abstrato. (Santos, 2010)

- c) **Processos de Colaboração** (Figura 3.15): descrevem processos B2B (*business-to-business*) e as interações entre duas ou mais entidades de negócio. Os diagramas processos são geralmente de um ponto de vista global. No exemplo, o Autorizador faz a autorização de pagamento por cartão de crédito, neste caso este processo interessa a Empresa 1 (que realiza a venda), logo ele deverá ser modelado explicitamente.

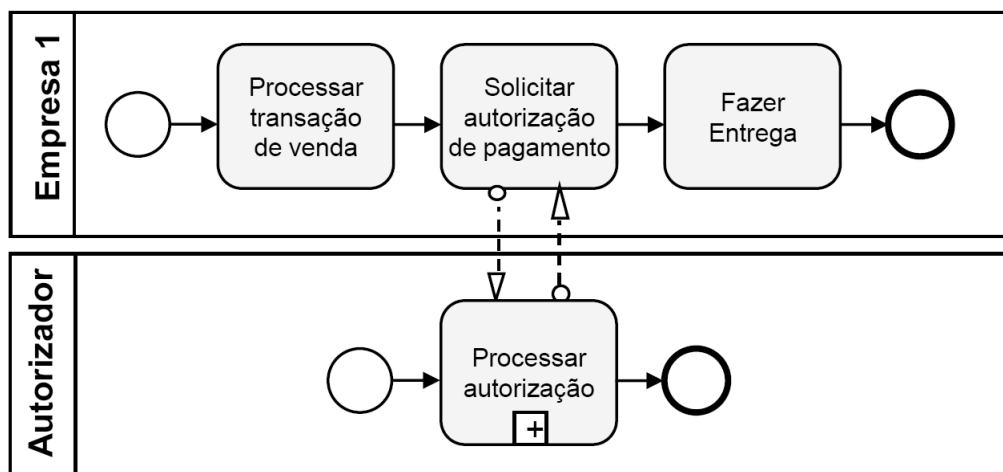


Figura 3.15: Exemplo de Processo de Colaboração. (Santos, 2010)

3.4 A Técnica i*

A técnica i*, proposta em (Yu, 1995a), permite descrever aspectos de intencionalidade e motivações envolvendo atores em um ambiente organizacional. Sendo assim, ela é capaz de reconhecer e descrever motivações, intenções e raciocínios sobre as características de um processo, oferecendo uma descrição mais rica, que facilita os esforços da Engenharia de Requisitos.

Segundo o *framework* i* (Yu, 1995a), as organizações consistem de unidades semiautônomas chamadas atores, cujo comportamento destes não é totalmente controlável ou previsível, mas regulamentado por relacionamentos sociais. Esses atores possuem a liberdade de ação, mas dependem um dos outros para alcançarem seus objetivos, executarem tarefas e fornecerem recursos.

Diferentemente de outras técnicas de modelagem, que descrevem tipicamente um processo em termos de etapas de atividades e fluxos entre entidades, a técnica i* se destaca por preocupar-se com as razões ou motivações que estão associadas a aspectos comportamentais do processo.

A técnica se baseia na premissa que um entendimento maior do processo pode ser obtido através de uma visão das intenções e estratégias. Dessa maneira, o ator é a unidade central a ser modelada. Um ator intencional não só executa atividades e produz ou consome entidades, mas tem motivações, intenções e razões associadas a suas ações (Chichinelli, 2002).

Segundo Yu (1995a), a técnica i* é utilizada para:

- a) obter uma melhor compreensão sobre os relacionamentos da organização, entre os vários atores do sistema;
- b) entender as razões envolvidas nos processos de decisões; e
- c) ilustrar as várias características de modelagem que podem ser apropriadas à Engenharia de Requisitos, principalmente, na fase inicial da especificação dos requisitos (Yu, Bois e Mylopoulos, 1995), (Yu, 1997).

Para descrever o ambiente organizacional, i* propõe dois modelos: o Modelo de Dependências Estratégicas (SD), que descreve as relações de dependências externas entre os atores da organização, e o Modelo de Razões Estratégicas (SR), que descreve interesses e conceitos dos participantes e as direções que podem seguir. Os dois modelos são descritos a seguir.

3.4.1 O Modelo de Dependências Estratégicas (SD)

O Modelo de Dependência Estratégica é uma rede de relacionamentos de dependência entre atores, e é composto por nós e ligações. Os nós representam os atores no ambiente e as ligações são as dependências entre os mesmos. Segundo o *framework* i*, um ator é uma entidade que realiza ações para obter objetivos no contexto do ambiente organizacional. Assim, atores dependem uns dos outros para obter objetivos, realizar tarefas, etc. O ator que depende de alguma forma de outro ator é chamado de *Depender*, e o ator que satisfaz o *Depender* é denominado de *Dependee*. O objeto de dependência entre *Depender* e *Dependee* é denominado de *Dependum*. Resumidamente, haverá relacionamentos do tipo *Depender*→*Dependum*→*Dependee*.

Neste modelo, distinguem-se quatro tipos de dependências (Figura 3.16): recursos, tarefas, objetivos, objetivos-*soft*. As três primeiras dependências são relacionadas à existência de intenções, e a quarta é associada com a ideia de requisitos não funcionais.

Na dependência de recurso, o *Depender* depende do *Dependee* em relação à disponibilidade de um recurso, seja uma entidade física ou de uma informação. Recurso, nesse caso, é o produto final de alguma ação, em um processo, que estará ou não disponível. Uma dependência de recurso pode ser vista na Figura 3.18, onde o ator “Participante de Reunião” depende do ator “Agendador de Reunião” para obter o recurso “Data Proposta”.

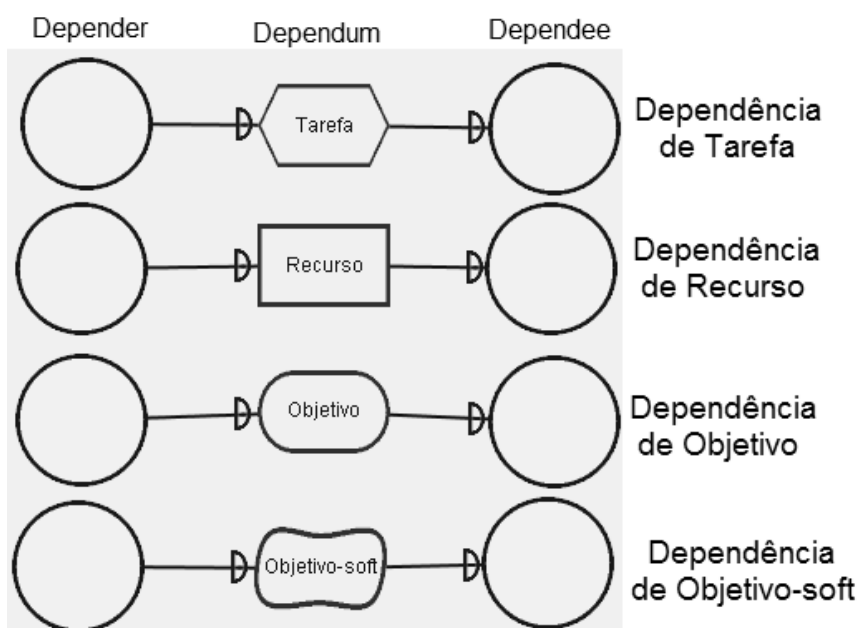


Figura 3.16: Relacionamentos de dependência entre atores em i*. (Yu, 1995a)

Na dependência de tarefa, o *Depender* depende do *Dependee* para executar uma atividade, porém é responsabilidade do *Depender* mostrar o caminho de como isto será feito. Ou seja, um agente informa ao outro o que deve ser feito, sem haver preocupação em se informar o “porque” fazer. A descrição de uma tarefa em i^* não tem por intenção ser uma completa especificação dos passos necessários à execução dessa tarefa, ela relaciona-se com as atividades a serem realizadas. Na Figura 3.18, o ator “Agendador de Reunião” depende do ator “Participante de Reunião” para executar a tarefa “Entrar Datas Disponíveis”.

Na dependência de objetivo, o *Depender* depende do *Dependee* para modificar um estado do mundo, ou seja, um agente depende de outro para que uma determinada intenção sua seja satisfeita, não importando a maneira como a mesma foi alcançada. O *Dependee* tem como papel satisfazer/alcançar um objetivo para o *Depender*. Por objetivo se entende uma condição (ou estado do mundo) que um ator gostaria de alcançar. Tem-se na Figura 3.18 três dependências de objetivo, sendo uma delas entre o ator “Iniciador de Reunião” que depende do ator “Participante Importante” para alcançar o objetivo “Comparecer reunião”.

A dependência de objetivo-*soft*, também referenciada como requisito não funcional na ER, é similar à dependência de objetivo, exceto pelo fato de que a condição de sucesso não é precisamente definida a priori, ou seja, a realização de um objetivo-*soft* é bastante subjetiva e o seu significado inicialmente não é claramente conhecido. Um bom exemplo é citado por Alencar (1999): o Cliente para voltar à Loja espera ser bem atendido, portanto atendimento bom é um objetivo-*soft*, pois não se consegue definir precisamente o que é ser bem atendido. Na Figura 3.18, uma dependência de objetivo-*soft* é “Assegurado Comparecer Reunião” entre os atores “Iniciador de Reunião” e “Participante Importante”.

Tem-se nos modelos SD, além das ligações de dependência, ligações IS-A, que descrevem uma ligação entre atores que caracterizam generalização, na qual um ator herda o comportamento (neste caso dependências) do ator pai. A notação gráfica, tanto para os elementos, quanto para as ligações em i^* , pode ser vista na Figura 3.17.

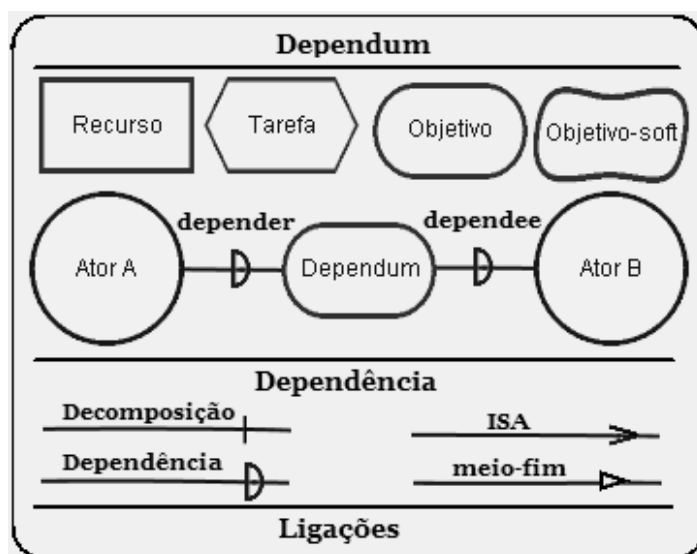


Figura 3.17: Notação gráfica dos elementos e ligações em i*. (Yu, 1995a)

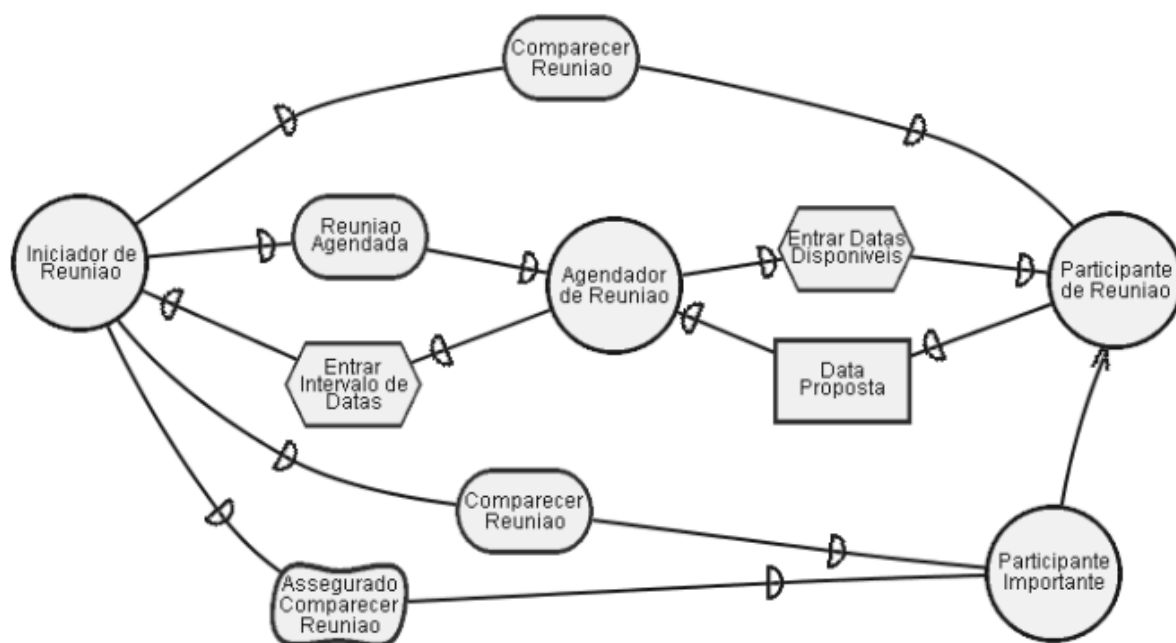


Figura 3.18: Modelo de Dependências Estratégicas para Agendamento de Reuniões. (Yu, 1995a)

Tendo como base estes elementos definidos para o modelo de SD, tanto as intenções quanto motivações e objetivos organizacionais podem ser modelados e alternativas para o desenvolvimento de sistemas computacionais podem ser avaliadas, optando-se por aquela que melhor satisfaça os objetivos de todos os *stakeholders* (Santander, 2002).

A seguir, o Modelo de Razões Estratégicas (SR) será descrito resumidamente.

3.4.2 O Modelo de Razões Estratégicas (SR)

O modelo de Razões Estratégicas (SR) pode ser considerado como um modelo complementar ao modelo de SD. O modelo SD fornece um nível de abstração, no qual são modelados apenas os relacionamentos externos entre os atores. Através do modelo SR, é possível compreender e modelar mais detalhadamente as razões estratégicas associadas com cada ator e suas dependências, auxiliando assim no processo de Engenharia de Requisitos (Yu, 1995a). De maneira geral, o modelo SR é utilizado para:

- a) descrever os interesses, relacionamentos e motivações dos participantes do processo;
- b) possibilitar a avaliação de possíveis alternativas na definição do processo; e
- c) investigar com mais detalhes as razões existentes atrás das dependências entre vários atores.

O modelo SR também é composto por nós e ligações. Nesse modelo, os nós têm como base os tipos de *Dependum*. Os elementos intencionais (objetivos, tarefas, recursos e objetivos-*soft*) aparecem no modelo SR como elementos internos ligados por elementos meio-fim e decomposição de tarefas, representando explicitamente as razões por detrás das dependências entre os atores e quais são as alternativas dos processos.

Os relacionamentos meio-fim indicam um relacionamento entre um fim – que pode ser um objetivo a ser alcançado, uma tarefa a ser realizada, um recurso a ser produzido, ou um objetivo-*soft* a ser satisfeito – e um meio, para se atender a esse fim. Este caminho ou meio para obter a meta é geralmente definido em termos de tarefas que são necessárias para atingi-la.

Nos relacionamentos de decomposição de tarefa, uma tarefa é modelada em termos de sua decomposição em subcomponentes, exprimindo o que deve ser feito para se ter a tarefa realizada. Assim, é possível a decomposição em unidades menores de um nó tarefa representando de forma mais detalhada as razões associadas com a realização da tarefa.

No modelo SR, uma rotina representa um subgrafo incluindo todas as razões, bem como os meios para se atingir um fim do ponto de vista de um ator. Através deste modelo, pode-se modelar alternativas (meios) para se obter fins estratégicos para um ator (Santander, 2002).

Com respeito aos relacionamentos que envolvem objetivos-*soft*, um atributo adicional é utilizado para designar o tipo de contribuição relacionada com esses objetivos (Yu, 1995a). A contribuição e seus respectivos atributos podem ser:

- a) **Positiva (+):** o objetivo-*soft* contribuirá positivamente;

- b) **Negativa (-)**: o objetivo-*soft* contribuirá negativamente;
- c) **Bastante (sup)**: objetivo-*soft* contribuirá de forma suficiente;
- d) **Não bastante (sub)**: o objetivo-*soft* contribuirá de forma não suficiente;
- e) **Desconhecida/indefinida (?)**: não se sabe qual será a influência do objetivo-*soft* em questão.

A seguir, a Figura 3.19 apresenta um exemplo de modelo SR, nesse caso para o Agendamento de Reuniões. Um exemplo de ligação meio-fim pode ser visualizado no ator “Iniciador de Reunião” entre o objetivo “Reunião Agendada” e a tarefa “Agendar Reunião”. Um exemplo de ligação de decomposição pode ser visualizado no ator “Iniciador de Reunião” entre a tarefa “Organizar Reunião” e o objetivo-*soft* “Rápido”.

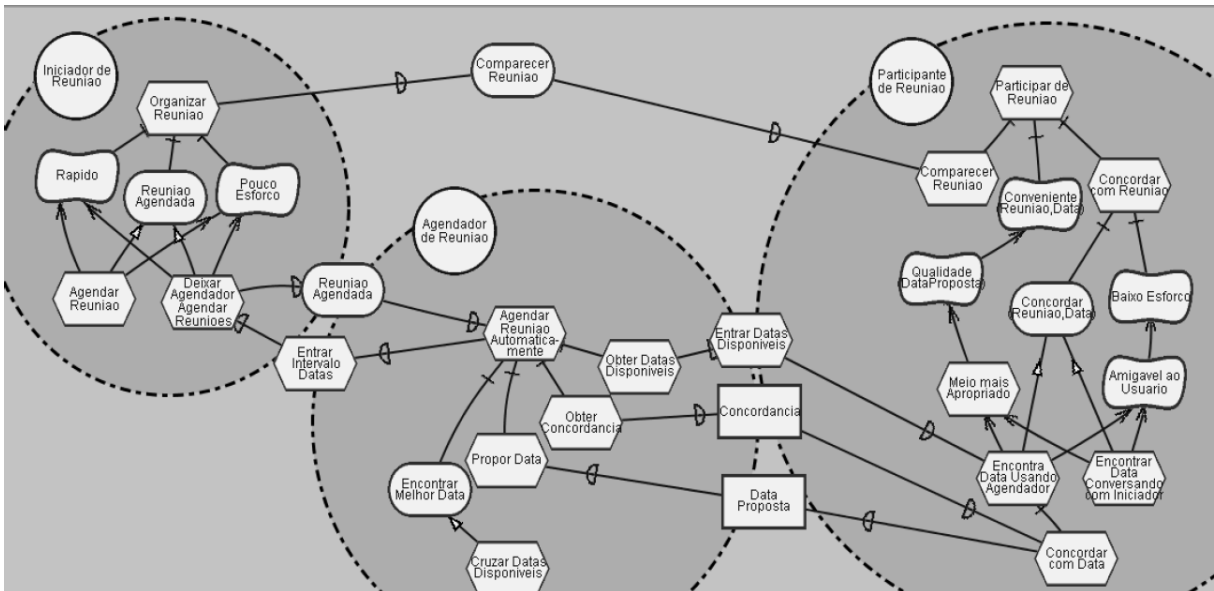


Figura 3.19: Modelo de Razões Estratégicas para Agendamento de Reuniões. (Yu, 1995a)

Em um caráter amplo, a técnica *i** é considerada de fácil entendimento pelos mais variados *stakeholders*. Quanto aos recursos gráficos, os mesmos são apontados como capazes e suficientes para representar as características mais relevantes do ambiente organizacional (Alencar, 1999).

No entanto, Santander (2002) verifica que observando a técnica mais detalhadamente, a mesma pode contribuir e ser utilizada de forma mais efetiva no processo de Engenharia de Requisitos. Para isso, os requisitos organizacionais representados em *i** devem ser relacionados aos requisitos funcionais dos sistemas pretendidos. Neste aspecto, o autor

observa nas dependências do tipo objetivo, recurso, tarefa e objetivo-*soft*, estabelecidas entre os atores em i^* , uma rica fonte de informações para descrever requisitos funcionais e detectar alguns requisitos não funcionais.

3.5 Fluxos de Trabalho (*workflows*)

O conceito de *workflow* (WF) foi desenvolvido a partir da noção de processo em sistemas de manufatura e de automação de escritórios. Segundo a WFMC (*Workflow Management Coalition*) (WFMC, 2010), entidade criada em 1993 que visa o estabelecimento e a adoção de padrões para a área de *workflow*, um processo é "*um conjunto coordenado de atividades (sequenciais ou paralelas) que são interligadas com o objetivo de alcançar uma meta comum*", sendo atividade conceituada como "*uma descrição de um fragmento de trabalho que contribui para o cumprimento de um processo*".

Dessa maneira, a WFMC (2010) define *workflow* como "*a automação total ou parcial de um processo de negócio, durante a qual documentos, informações e tarefas são passadas entre os participantes do processo*". Sendo assim, um *workflow* pode ser definido como sendo um conjunto de atividades (descrição de um fragmento de trabalho) processadas ao mesmo tempo (ou não) com uma possível especificação de controle e fluxo de dados entre atividades relacionadas.

Segundo Georgakopoulos, Hornick e Sheth (1995), um *workflow* pode descrever tarefas de processos de negócio em um nível conceitual necessário para compreender, avaliar e reprojeter o processo de negócio.

Um *workflow* também define a ordem de execução e as condições pelas quais cada tarefa é iniciada e é capaz de representar a sincronização das tarefas e o fluxo de informações. Além disso, o conceito de *workflow* está relacionado com o conceito de reengenharia e automatização de processos de negócios e de informação.

Alguns trabalhos, como (Canós, Penadés e Carsí, 1999), afirmam que muitos dos princípios da Engenharia de Software podem ser aplicados ao processo de desenvolvimento de WF. Em particular, técnicas, métodos e ferramentas utilizadas na Engenharia de Requisitos podem ser aplicadas na modelagem do WF, ajudando na construção de modelos completos e corretos.

Um sistema de gerência de *workflow*, segundo a WFMC (2010), é um software que permite a definição, criação e gerência da execução de *workflows*, sendo capaz de interpretar

Os principais elementos do modelo são:

- a) **Processo**: um processo é composto por atividades e/ou subprocessos e/ou condições de transição. O controle de fluxo conecta esses elementos e estabelece a ordem correta da execução do processo. Cada processo tem um identificador, um nome, uma descrição, uma condição de início, uma condição e um estado final;
- b) **Atividade**: é qualquer peça atômica de trabalho que constitui um passo lógico dentro de um processo. Como um processo, cada atividade tem um identificador, um nome, uma descrição, uma condição de início, uma condição de fim, um estado e um conjunto de ações específicas associadas. Uma atividade pode ser manual ou automática; atores humanos executam atividades manuais (por exemplo, tomando uma decisão), enquanto que as automáticas são executadas por um computador e, normalmente, consistem na invocação de um aplicativo externo;
- c) **Subprocesso**: um subprocesso é um processo que faz parte de outro processo, isto é, constitui um passo em um processo complexo. Isto permite a introdução de modularidade em modelos de WF;
- d) **Condição de Transição**: é possível incluir as condições de transição a seguir no fluxo de controle de um processo: junção E; junção OU; divisão E; divisão OU;
- e) **Dados**: são todas as informações necessárias para a execução do processo (ou atividade de entrada/saída ou na avaliação das condições de transição). Normalmente, eles são persistentes (armazenada em um banco de dados). Quando começa uma atividade, consultam-se os dados de entrada a partir do repositório; quando a atividade termina, ele armazena os dados de saída no mesmo repositório;
- f) **Ator**: representa a participação humana no WF.

A Figura 3.21 mostra um exemplo de *workflow* que representa um sistema de suporte “on-line” para usuários. Este *workflow*, como pode ser visto, executa acesso a bases de dados, armazenando os dados gerados em uma atividade para o uso nas atividades seguintes.

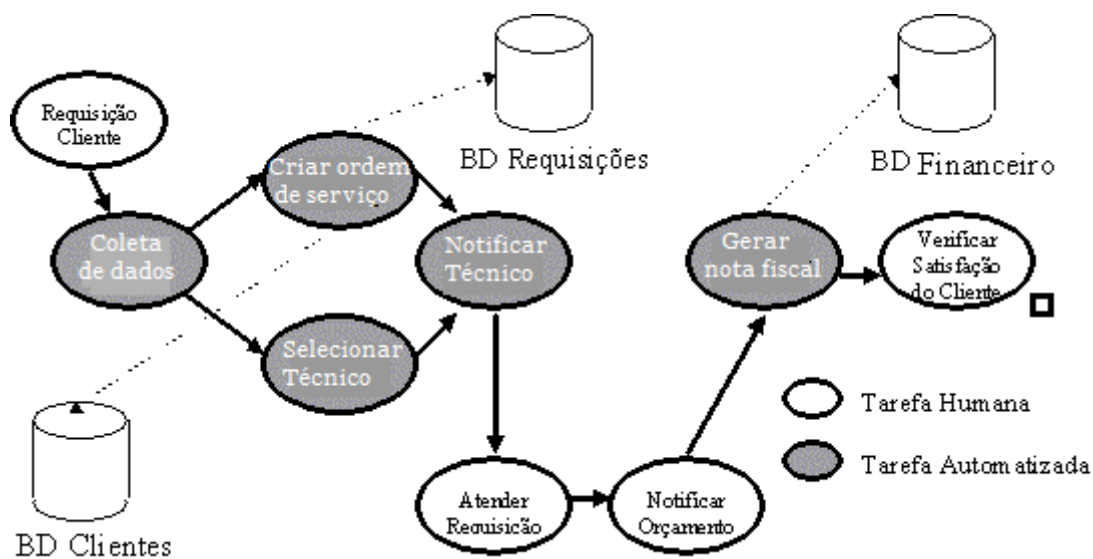


Figura 3.21: Exemplo de *Workflow* representando um sistema de atendimento on-line. (Duarte, 2010)

Um dos maiores problemas da modelagem de sistemas de *workflow* é o fato que praticamente cada sistema de gerência de *workflow* utiliza sua própria técnica de modelagem. Porém, a introdução da BPMN nesse contexto objetiva sanar essa problemática, pois trata-se de uma notação gráfica padronizada para desenhar processos de *workflow*. Antes do surgimento da BPMN, várias formas de modelagens eram encontradas na literatura e nos softwares. Por isso a necessidade de se ter um padrão de modelagem para os processos de negócio. Apoiada pela OMG, a BPMN está cada vez mais se consolidando como um padrão. Percebe-se que a cada dia cresce o número de empresas que suportam esse tipo de notação.

3.6 Regras de Negócio

As Regras de Negócio são uma forma de ajudar a resolver o problema da má definição de requisitos, mostrando que elas representam um importante conceito na fase de definição de requisitos organizacionais, facilitando modificações no sistema de informação quando as regras mudam, e proporcionam oportunidade para as pessoas do negócio avaliarem a execução de cada processo.

Segundo Rosca *et al.* (1997), regras de negócio são uma nova categoria de requisitos do sistema que representam decisões sobre como executar o negócio, e são caracterizadas pela orientação do negócio e sua tendência às mudanças.

Leite e Leonardi (1998) entendem regras de negócio diferente de requisitos. Para os autores, regras de negócio são declarações sobre a forma de a empresa fazer negócio. Essas

regras refletem políticas do negócio. Organizações têm políticas para satisfazer os objetivos do negócio, satisfazer clientes, fazer bom uso dos recursos, e obedecer às leis ou convenções gerais do negócio. Regras de negócio tornam-se requisitos, ou seja, podem ser implementados em um sistema de software como uma forma de requisitos de software desse sistema.

Gottesdiener (1997) afirma que regras de negócio podem oferecer muitos benefícios, como:

- a) rapidez no desenvolvimento de software;
- b) melhor qualidade dos requisitos;
- c) facilidade de mudança;
- d) balanceamento entre flexibilidade e controle centralizado.

Ao permitir que regras de negócio sejam definidas e gerenciadas separadamente, fazendo uma ligação com a Engenharia de Software, gerando e mantendo aplicações das regras do negócio, tem-se um excelente potencial para evoluir a qualidade do desenvolvimento de Sistemas de Informação (SI).

A propriedade essencial de regras de negócio é que elas são expressas em termos de um modelo organizacional, em vez de em termos de modelos de dados. Isso é devido ao fato que essas regras devem ser capturadas por pessoas orientadas ao negócio, não sendo necessariamente desenvolvedores de software. Outra importante propriedade de regras de negócio é que estas mudam frequentemente devido à natureza dos fatores que as influenciam, como: regulamentos governamentais, tendências do mercado, tomada de decisão. Mudanças nas regras do negócio necessitam ser apoiadas dentro dos processos de negócio para a organização reagir a essas influências (Rosca e Wild, 1996).

3.6.1 Representação de Regras de Negócio

Segundo Herbst (1996), regras de negócio podem ser representadas de acordo com as regras das bases de dados ativas, usando três componentes básicos: Eventos, Condição e Ação (ECA). Alguns estudos de caso para extrair regras do negócio foram aplicados em sistemas de informação e revelam a necessidade da extensão dessa estrutura para ECAA (Evento, Condição, Então-Ação, Senão-Ação). O evento deve ser conhecido quando a regra do negócio tem que ser processada. Na condição é importante saber o que deve ser verificado, o que tem que ser feito se caso a condição for verdadeira ou falsa. O autor afirma que a estrutura ECAA

permite especificar regras de negócio individuais e a definição de processos completos. Para Rosca *et al.* (1997), ECA tem uma melhor definição de semântica operacional.

Na pesquisa de Herbst (1996) é demonstrado um estudo de caso de uma seguradora. A fim de mostrar o escopo e o potencial do uso de regras de negócio, o autor exemplifica o registro de sinistros em nível geral. O processo se inicia quando uma pessoa faz contato com a companhia de seguros, a primeira regra de negócio executa uma verificação, se a pessoa é um assegurado e se o assegurado quer informar um sinistro.

Regra do Negócio 1: ‘PESSOA-CONTATA-NOS’

ON (chamada telefônica) ou (carta da pessoa)
IF (pessoa é um assegurado) e (assegurada relata um sinistro)
THEN inicia o registro do sinistro
chama evento ‘SINISTRO DECLARADO’

O processamento da próxima regra leva para o registro provisório de sinistro se o assegurado já forneceu todas as informações sobre o sinistro, se não ele recebe um formulário para completar.

Regra do Negócio 2: ‘REGISTRO-PROV-SINISTRO’

ON sinistro declarado
IF (informação sobre o sinistro foi avaliada)
THEN registra sinistro provisoriamente
data-registro := hoje()
chama o evento ‘SINISTRO-REGISTRADO’
ELSE manda formulário para o assegurado
chama o evento ‘FORMULÁRIO DE SINISTRO ENVIADO’

O registro de informação incompleta sobre o sinistro é seguido pela checagem preliminar no caso do sinistro ser coberto pela apólice do assegurado. Dependendo do resultado, o pedido do assegurado pode ser rejeitado e um formulário é enviado ao assegurado para completar as informações.

Regra do Negócio 3: ‘ACEITA-PROV-SINISTRO’

ON sinistro provisório registrado

pessoas do negócio entendam o que o sistema pode fazer para melhorar a qualidade de seus negócios e rever os processos atuais.

3.7 Maps

Um *Map* é um modelo de processo, expresso em termos intencionais. Ele oferece um sistema de representação baseado em uma ordenação não determinística das intenções (objetivos) e estratégias (Rolland, Prakash e Benjamin, 1999). Um *Map* permite de forma simples, porém expressiva, representar os processos em termos de objetivos e estratégias. Engenheiros de requisitos utilizam *Maps* devido à sua fácil compreensão e simplicidade (Rolland, 2007). Os usuários são capazes de compreender a técnica *Map* através de conceitos simples e uma modelagem gráfica de fácil entendimento. Considera-se que o *Map* é muito útil para explorar e modelar os objetivos e estratégias do sistema pretendido, apesar de não fornecer a compreensão dos “porquês”, que sustentam os requisitos organizacionais do sistema.

Map é uma abordagem dirigida a objetivos cuja meta é capturar os objetivos de uma empresa ou sistema e determinar as estratégias que possam contribuir para o cumprimento destes objetivos. No contexto da Engenharia de Requisitos, *Map* identifica os objetivos de negócio, fornecendo uma representação baseada em uma ordenação não determinística das intenções e estratégias. Diagramas *Map* (MD) consistem em um grafo (chamado *Map*) cujos nós são os objetivos e as arestas são as estratégias.

Um exemplo de *Map* para o processo de reserva de quarto é mostrado na Figura 3.23. Os nós do grafo identificam as diferentes intenções que os *stakeholders* podem querer alcançar. As arestas identificam as estratégias que podem ser utilizadas para alcançar esses objetivos. Uma aresta que aponta para um determinado nó indica um caminho para alcançar o objetivo correspondente, e várias arestas que apontam para o mesmo nó indicam caminhos alternativos para alcançar o mesmo objetivo. Cada *Map* tem dois objetivos especiais, “Iniciar” e “Parar”, associados ao estado inicial e final, respectivamente. Esses objetivos, assim como qualquer outro objetivo, podem aparecer apenas uma vez em um determinado *Map*.

A natureza da direção em um grafo *Map* especifica a relação de precedência entre os objetivos alcançados enquanto o *Map* for percorrido. Por exemplo, a Figura 3.23 mostra que, a partir do objetivo-fonte “Fazer Reserva de Quarto”, existem duas estratégias possíveis para

alcançar o objetivo-alvo “Realizar Pagamento”: “Por transferência eletrônica” ou “Por cartão de crédito”.

A agregação de um objetivo-fonte, um objetivo-alvo, e uma estratégia, é chamada de “Seção”. Seções alternativas fornecem rotas alternativas para o alcance dos objetivos dos *stakeholders*. Seções podem ser refinadas a um nível mais detalhado em outro *Map*.

A abordagem do *Map* é considerada vantajosa sobre as tradicionais técnicas de modelagem de objetivos, pois *Map* consegue capturar a variabilidade através da análise de requisitos. Entende-se por variabilidade a capacidade de mudar, personalizar ou configurar sistemas de softwares de acordo com os requisitos do usuário.



Figura 3.23: Exemplo de *Map*. (Rolland, 2007)

Outra vantagem importante do *Map* é que essa abordagem incentiva a participação dos *stakeholders* e ajuda a resolver problemas de comunicação entre as pessoas do negócio e os analistas de sistema. Esses são dois dos problemas atuais na área da Engenharia de Requisitos, os quais foram apresentados no Capítulo 2.

Alguns estudos propõem a integração das técnicas *Map* e *i**, de forma a reduzir a lacuna entre a modelagem organizacional e a funcional. Além disso, a integridade de ambos os modelos puderam ser melhorados quando analisadas as novas estratégias e objetivos na observação dos dois modelos. Essa proposta também ajuda a resolver um inconveniente do *Map*, que é a falta de um processo sistemático para construí-los. Na proposta, são gerados e explorados *Maps* através de modelos *i**, onde os objetivos mais importantes e as estratégias para o *Map* são decorrentes dos modelos *i**.

3.8 Casos de Uso de Negócio

Casos de Uso é uma maneira simples e natural de identificar os processos de negócio. O cliente é um “usuário” da organização, usando a organização por meio do processo de negócio. Um Caso de Uso de Negócio é um conjunto de cenários (instâncias), onde cada cenário é uma sequência de ações feitas pelo negócio que produzem um resultado para um

ator de negócio. Devem focar na interação entre o ator de negócio e o processo como o negócio é percebido externamente por quem o utiliza. Portanto, detalhes internos que não são vistos externamente não devem ser mostrados em detalhes (Jacobson, Ericson e Jacobson, 1994).

Do ponto de vista da organização, um caso de uso de negócio está associado aos objetivos e resultados organizacionais. Dessa maneira, um caso de uso do negócio:

- a) deve ser iniciado por um ator do negócio, embora haja exceções;
- b) descreve uma funcionalidade completa de um processo do negócio conforme percebida por um ator do negócio;
- c) gera como resultado algo de valor tangível para um ator do negócio (usuário);
- d) expressam os requisitos do negócio.

A Figura 3.24 apresenta as notações básicas para descrever Casos de Uso.

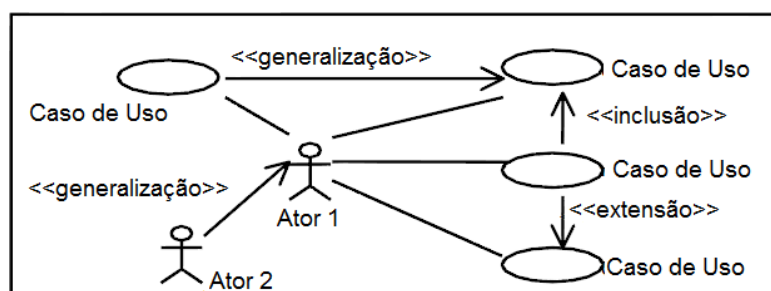


Figura 3.24: Notações básicas para Casos de Uso em UML. (Santander, 2002)

A seguir, tem-se um exemplo de Caso de Uso de Negócio (Figura 3.25). Nesse exemplo, o candidato realiza a inscrição para o vestibular.

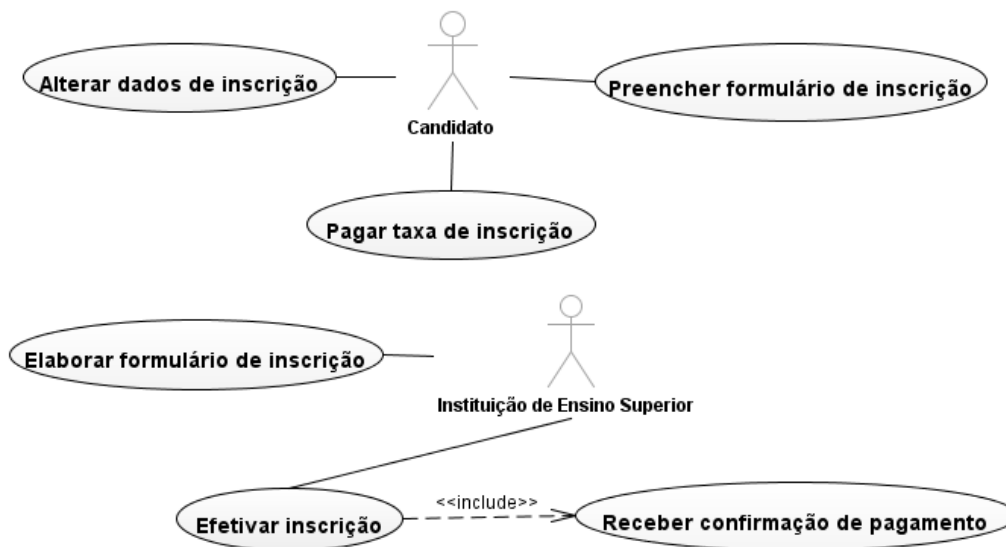


Figura 3.25: Exemplo de Caso de Uso de Negócio.

A especificação de um caso de uso do negócio pode ser feita através da descrição de sequências de eventos em formato de texto. Deve descrever como o ator do negócio e o caso de uso interagem, considerando:

- a) como e quando o caso de uso inicia e termina;
- b) quando o caso de uso interage com um ator envolvido;
- c) a sequência padrão (cenário de sucesso principal);
- d) as sequências alternativas ou de exceções (extensões).

Essas extensões caracterizam situações em que existem duas ou mais opções de continuidade no fluxo de uma determinada seção. Dentro do Cenário de Sucesso Principal de uma seção são indicados os desvios para subseções. Cada desvio deverá ter uma subseção usando novamente um Cenário de Sucesso Principal.

A título de exemplificação, para o caso de uso de negócio “Preencher formulário de inscrição”, da Figura 3.25, tem-se a seguinte descrição detalhada do mesmo, proposta por Jacobson (1992):

Nome do Caso de Uso: Preencher formulário de inscrição.

Atores participantes: Candidato.

Objetivos: Preencher o formulário de inscrição para o vestibular.

Pré-condições: O formulário de inscrição deverá ser retirado na Instituição de Ensino Superior.

Fluxo Principal:

1. O candidato preenche os dados pessoais necessários no formulário.
2. O candidato preenche os dados do curso pretendido no formulário.
3. O candidato preenche no formulário em qual campus deseja estudar.

Fluxos alternativos:

1a. O candidato desiste e devolve o formulário de inscrição sem o preenchimento para a atendente.

2a. A Instituição de Ensino Superior não oferece o curso pretendido pelo candidato.

Pós-condições: Formulário de inscrição para o vestibular preenchido.

Santander (2002) apresenta as vantagens de integrar a técnica de modelagem organizacional i* e os Casos de Uso em UML. Infelizmente, a UML e técnicas baseadas em cenários em geral não estão equipadas para modelar os requisitos organizacionais. Através da

proposta do autor, engenheiros de requisitos podem desenvolver Diagramas de Caso de Uso em UML a partir dos modelos organizacionais propostos na técnica i*.

3.9 Análise das Técnicas

Nesta seção, pretende-se apresentar algumas características percebidas (ou desejáveis) nos modelos estudados e a sua importância na modelagem eficiente de diferentes organizações.

A princípio, nota-se que os processos de negócios sempre estarão presentes sob a estrutura organizacional das empresas. A principal diferença entre uma empresa de sucesso e uma empresa com sérios problemas, na maioria dos casos, pode estar relacionada ao gerenciamento dos processos de negócios. Sendo assim, um modelo que consiga representar, gerenciar, ou até automatizar, eficientemente os processos de negócio torna-se muito importante nesse contexto.

As técnicas aplicadas no desenvolvimento de software devem tratar de aspectos relacionados à funcionalidade do sistema, à descrição das atividades e entidades. Porém, não devem se restringir a isso. Esses modelos devem ser capazes de buscar soluções alternativas para problemas da organização, através da representação dos objetivos da mesma, as razões envolvidas no processo, as regras de negócio, as restrições, os aspectos não funcionais, as estratégias, políticas, ou seja, fornecer descrições mais ricas sobre as organizações sócio humanas.

Um aspecto a ser considerado é o grau de complexidade de um determinado modelo. É desejável a representação de uma organização através de modelos fáceis de serem visualizados, com notações gráficas intuitivas, simples e consistentes. Os diversos *stakeholders* devem ser capazes de aprender e compreender o modelo de forma natural, sem esforços excessivos.

Quando trata-se de uma organização de grande porte, é inevitável um sistema com certa complexidade. Dessa maneira, modelos que permitam representar os departamentos da organização com diferentes interesses podem ser mais aplicáveis a essa situação, pois se utilizam de múltiplas visões permitindo uma melhor compreensão da estrutura organizacional.

A construção desses modelos deve ser apoiada por ferramentas computacionais eficientes, como forma de automatizar o processo e diminuir o tempo necessário para gerar os modelos organizacionais.

Analisando brevemente algumas das técnicas estudadas, nota-se que o modelo EKD é bastante rico em informações organizacionais. O modelo consegue representar os objetivos da organização, os principais conceitos, as regras e processos de negócio, os atores e recursos envolvidos, e os requisitos da organização. Para tanto, a técnica utiliza vários submodelos para representar a estrutura organizacional de diferentes pontos de vista, facilitando a compreensão de sistemas mais complexos. Porém, esse modelo pode não ser apropriado quando ocorrem muitas mudanças na organização, visto que isso requer a alteração de diferentes elementos em diversos submodelos, resultando em um esforço adicional ao analista.

Já o modelo i^* se resume em uma abordagem mais simples e que permite modelar as intenções, os relacionamentos e as motivações entre membros de uma organização. Essa técnica possui dois modelos, o modelo de Dependências Estratégicas (SD) e o modelo de Razões Estratégicas (SR). O primeiro (SD) fornece uma descrição intencional de um processo em termos de uma rede de relacionamentos de dependência entre atores. O segundo modelo (SR) possui uma representação e uma racionalização mais explícita sobre os interesses dos atores, e de como estes interesses podem ser atendidos ou afetados pelos diversos sistemas. Percebe-se que a técnica i^* é deficiente quando se trata de processos de negócio, pois a modelagem desses elementos é realizada apenas em alto nível, desprezando a sequencialidade das atividades inerentes ao processo.

Com relação à notação BPMN, fica claro que o modelo é capaz de representar os processos de negócio de forma simples e consistente, através de notações gráficas de fácil entendimento. Contudo, a técnica desconsidera aspectos intencionais, não representando os objetivos e metas presentes na organização. Outro fator negativo é quando se deseja uma modelagem mais abrangente da organização, sendo necessária a representação de todos os elementos presentes na mesma, e não apenas os processos de negócio.

Uma expectativa e necessidade de engenheiros de requisitos é que estes modelos organizacionais que já possuem necessidades expressas na forma de satisfação de objetivos, realização de tarefas e obtenção de recursos, possam servir de base para capturar a funcionalidade de sistemas computacionais pretendidos pela organização. Dessa forma, também são consideradas interessantes as técnicas que já possuam estudos publicados na área onde o foco é a integração do modelo organizacional a modelos funcionais.

3.10 Considerações Finais

Este capítulo apresentou uma visão geral a respeito da fase de modelagem organizacional no contexto de desenvolvimento de sistemas computacionais. Foram apresentadas diversas técnicas de modelagem organizacional, citando suas principais características. As técnicas abordadas foram EKD, BPMN, i*, *Workflows*, Regras de Negócio, *Maps* e Casos de Uso de Negócio.

Estas técnicas estudadas apresentam diferentes métodos e modelos para representar o conhecimento sobre processos organizacionais e sobre os relacionamentos entre os membros de uma instituição. A representação destas informações é muito útil para uma adequada compreensão da organização, resultando na elicitación mais efetiva e consistente dos requisitos de sistemas computacionais pretendidos.

A partir da compreensão da estrutura e ambiente organizacional, os engenheiros de requisitos são capazes de definir como o sistema computacional pretendido conseguirá satisfazer os objetivos da empresa, por que ele é necessário, quais as alternativas existentes, quais as implicações das alternativas para as várias partes interessadas, entre outros aspectos. Além disso, vários elementos nos modelos estudados podem ser utilizados e mapeados para requisitos funcionais e não funcionais do sistema pretendido.

Capítulo 4

Estudo Comparativo e Avaliação dos Modelos Organizacionais

Neste capítulo, tendo como base os estudos realizados sobre as técnicas de modelagem organizacional, é apresentado um estudo voltado à necessidade de comparação desses modelos, a fim de capturar as principais características e qualidades dos mesmos que são interessantes e desejáveis no contexto da Engenharia de Requisitos. Para tanto, um experimento utilizando as diversas técnicas é conduzido (Seção 4.1). Como forma de apoiar essa experimentação, são utilizados princípios da Engenharia de Software Experimental. Uma avaliação a respeito dos resultados obtidos nesse experimento é realizada na Seção 4.2, seguida pelas considerações finais (Seção 4.3).

4.1 Estudo Comparativo

Como forma de diminuir a lacuna entre negócios e tecnologia da informação, é apresentado um estudo comparativo entre as técnicas de modelagem organizacional mostradas no Capítulo 3. Para tanto, utiliza-se um estudo de caso real (detalhado na Seção 4.1.1), fornecido por uma empresa de desenvolvimento de software da região oeste do Paraná, o qual é modelado a partir de cada técnica anteriormente estudada. Assim, através das diferentes representações do contexto organizacional de onde o software será inserido, pode-se extrair na prática as características das técnicas e apontar suas principais qualidades.

Com o propósito de que um experimento forneça resultados válidos, o mesmo deve ser organizado, controlado e acompanhado. Dessa forma, a Engenharia de Software Experimental, introduzida no Capítulo 2, serve de apoio para a realização do estudo comparativo, e a sua utilização é apresentada na Seção 4.1.2.

4.1.1 Contexto do Estudo de Caso – Cotação de Compras

Na área de concessionárias de veículos e implementos agrícolas, a empresa fornecedora do estudo de caso ajuda a gerenciar revendas Volkswagen, Ford, Fiat, Iveco, Agrale, Honda, Peugeot, Toyota, Massey Ferguson, New Holland, EFFA, Sinus truck, entre outras, em quase todo o Brasil. A intimidade com o negócio de veículos transformou a organização em uma especialista na gestão de concessionárias e relacionamento gerencial com fornecedores e consumidores.

Na área comercial, a organização é parceira de distribuidoras, *auto-centers*, materiais elétricos, auto-peças, revendas de veículos multimarcas, entre outros, fornecendo ferramentas ágeis para o controle gerencial, compra, venda, custos e estoque.

O ciclo de vida de projeto na empresa consiste em cinco fases: concepção, planejamento, produção, implantação e encerramento. Nessas fases são definidas atividades de acordo com o escopo dos requisitos, as estimativas para os recursos e a natureza do projeto, de forma a possibilitar o maior controle gerencial.

O objetivo principal da fase de concepção é estabelecer o escopo do software do projeto e as condições limite, incluindo o que deve ou não estar no produto e a elaboração da proposta comercial de projeto.

A meta da fase de planejamento é criar o plano de projeto a fim de fornecer uma base estável para o esforço da fase de construção. O plano de projeto se desenvolve a partir do documento de escopo construído na fase de concepção e da avaliação dos riscos.

O fase de produção é responsável por implementar os requisitos especificados no documento de requisitos de projeto. A fase de construção é de certa forma um processo de manufatura, em que a ênfase está no gerenciamento de recursos e controle de operações para otimizar custos, programações e qualidade. Esta fase está dividida em duas etapas iterativas: desenvolvimento e teste.

Na fase de implantação o ambiente do usuário é preparado para o recebimento do produto final, ou seja, a instalação do software.

A fase de encerramento marca o fim do projeto. Nela devem-se analisar os artefatos gerados, incluindo as atas das reuniões, a fim de verificar se houveram falhas ou atrasos no cronograma. Com o decorrer do processo a empresa pode analisar os resultados obtidos e aplicar correções em setores/atividades específicas.

O ciclo de vida de projeto citado acima pode ser visualizado na Figura 4.1.

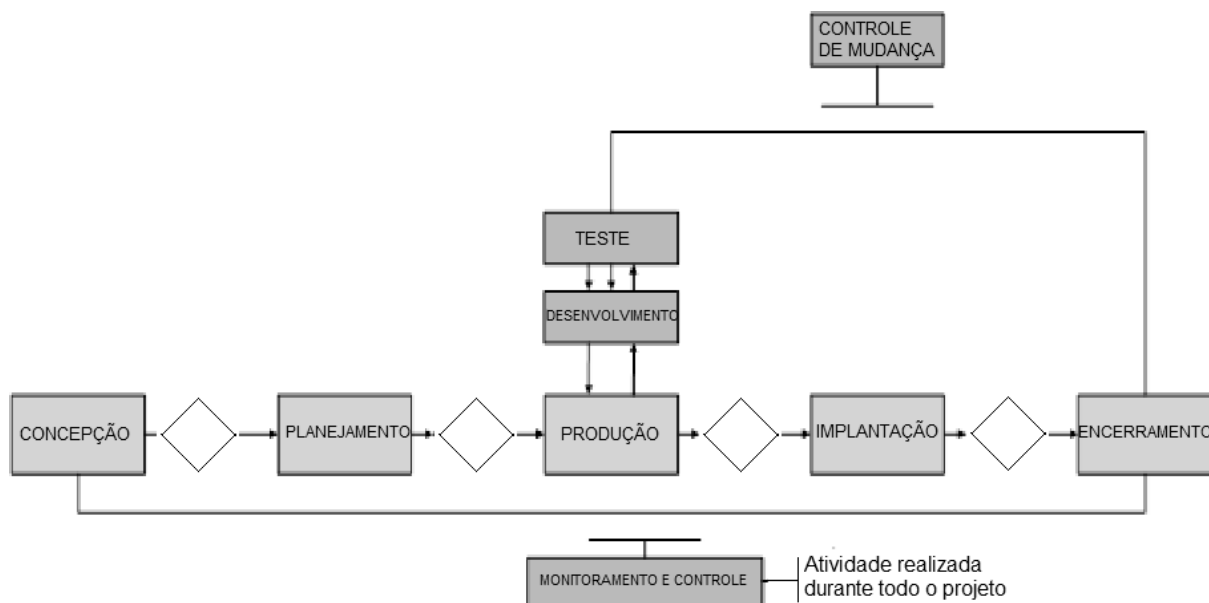


Figura 4.1: Ciclo de vida de projeto da empresa desenvolvedora de software.

Uma das maiores dificuldades da organização se concentra na fase de concepção do projeto. Nem sempre é fácil atingir o consenso entre todos os envolvidos sobre os objetivos do projeto. Segundo o analista de processos da empresa, a fase de iniciação tem muita importância principalmente para os esforços dos novos desenvolvimentos, nos quais há muitos riscos de negócios e de requisitos que precisam ser tratados para que o projeto possa prosseguir. Para projetos que visam melhorias em um sistema existente, a fase de iniciação é mais rápida, mas ainda se concentra em assegurar que o projeto seja compensatório e que seja possível realizá-lo.

Um dos artefatos gerados na fase de concepção é o Documento de Requisitos de Projeto (DRP), onde devem estar descritos textualmente os requisitos funcionais e não funcionais do sistema a ser desenvolvido. Porém, como já foi dito, identificar e especificar requisitos de um sistema computacional não é uma tarefa trivial, sendo necessária a utilização de técnicas de levantamento de requisitos para facilitar essa elicitacão.

Atualmente, a organização utiliza apenas questionários e entrevistas para a identificação dos requisitos. Embora seja uma técnica simples de utilizar e bastante eficaz numa fase inicial de obtenção de dados (e mesmo de esclarecimento de algumas dúvidas), está condicionada a alguns fatores:

- a) influência do entrevistador nas respostas do cliente: convém que o entrevistador dê margem ao entrevistado para expor as suas ideias, sem ser tendencioso;

- b) relação pessoal entre os intervenientes na entrevista;
- c) predisposição do entrevistado: caso, por exemplo, o papel do entrevistado venha a ser afetado negativamente pela introdução de um sistema na organização, este pode propositadamente dificultar o acesso à informação;
- d) capacidade de seguir um “plano” para a entrevista: na ausência destes planos é natural que haja tendência para que os intervenientes se dispersem um pouco, levando a que a entrevista demore mais tempo do que seria suposto. Caso a entrevista se torne demasiado longa, as pessoas podem cair na tentação de "querer despachar", sendo os últimos pontos da entrevista abordados de forma superficial (ou podem nem chegar a ser abordados).

Dessa forma, o DRP algumas vezes possui certas inconsistências, não representando as reais necessidades da organização que pretende introduzir o software. Isso resulta em esforços adicionais de desenvolvimento nas próximas fases do ciclo de vida do projeto. Exemplo disso é que, frequentemente, o próprio setor de desenvolvimento fica responsável por entrar em contato com o cliente para esclarecer possíveis dúvidas que surgem no decorrer do processo de construção do software.

Uma forma de sanar essa dificuldade da empresa e contribuir com a satisfação do cliente é introduzir técnicas de modelagem organizacional para facilitar a etapa de identificação de requisitos organizacionais e requisitos associados ao sistema computacional pretendido. Desta forma, poderíamos diminuir consideravelmente os custos de desenvolvimento nas fases do projeto posteriores a essa.

Hoje em dia, um dos principais produtos da organização é o Sistema de Gerenciamento Comercial e de Concessionárias, o qual fornece, em ambiente gráfico, um sistema ágil, seguro e fácil de usar. Esse software é responsável pelas seguintes funcionalidades:

- Controle de custos;
- Controle de estoques (entrada e saída);
- Controle de vendas (salários, comissões).

O problema é que o supracitado software não disponibiliza, no seu sistema de compras (no contexto de controle de estoques), uma função para auxiliar na determinação do menor preço para efetuar a compra de itens.

Essa nova funcionalidade foi solicitada por um cliente da empresa, e, através de entrevistas com o mesmo, os analistas puderam identificar alguns requisitos, sugerindo a seguinte

solução: desenvolver um sistema de lançamento de orçamentos, a fim de facilitar a escolha do menor preço para determinado item, através das cotações passadas pelos fornecedores. O processo proposto para a resolução do problema, intitulado Cotação de Compras, é descrito a seguir.

Primeiramente, o funcionário cadastrará uma nova cotação no sistema, inserindo o número de controle interno, a descrição, a data, os fornecedores e os itens envolvidos para essa cotação.

Após o cadastro, o funcionário irá gerar relatórios de pedidos de cotação, um para cada fornecedor, contendo todos os itens cadastrados na cotação. Esses relatórios deverão ser preenchidos pelos fornecedores, informando o custo de cada item, e devolvidos a empresa.

De posse dos relatórios preenchidos, o funcionário cadastrará as propostas dos fornecedores para cada item daquela cotação. Dessa maneira, o sistema poderá realizar a análise dos preços de um determinado item, comparando-os e informando qual o menor preço e seu respectivo fornecedor. Um relatório contendo essa análise poderá ser gerado.

Com base nessa análise, é possível gerar um pedido de compra, podendo ser apresentado também em forma de relatório.

Desta forma, observando esta problemática, é essencial entender inicialmente todos os elementos estratégicos organizacionais que estão envolvidos na realização deste novo módulo para o sistema de compras. A partir deste entendimento, pode-se então elicitar de forma mais confiável e completa os requisitos detalhados deste novo módulo.

4.1.2 Utilizando a Engenharia de Software Experimental

Como o estudo realizado tem como objetivo a comparação das características de técnicas de modelagem organizacional distintas, pode-se representar a estrutura da experimentação, através da abordagem GQM apresentada no Capítulo 2, da seguinte forma:

- **Objetivo global:** estudar as técnicas de modelagem organizacional EKD, BPMN, i*, *Workflows*, Regras de Negócio, *Maps* e Casos de Uso de Negócio, objetivando compreender e comparar as suas características no contexto da Engenharia de Requisitos.

- **Objetivo do estudo:**

Analisar → *as técnicas de modelagem organizacional*

Com a finalidade de → *comparar as suas características*

Com respeito a → *representatividade dos requisitos organizacionais e futura integração com modelos funcionais*

Do ponto de vista dos → *stakeholders*

No contexto de → *pequenas e médias organizações, representadas pela organização escolhida.*

- **Questões (Q)/Métricas (M):**

Q01: A técnica possui múltiplas visões?

M01: O(s) elemento(s) da técnica que permite(m) múltiplas visões.

Q02: A técnica consegue representar diferentes aspectos da organização?

M02: O(s) elemento(s) da técnica que permite(m) representar diferentes aspectos da organização.

Q03: A técnica consegue representar os objetivos da organização?

M03: O(s) elemento(s) da técnica que permite(m) representar os objetivos da organização.

Q04: A técnica consegue representar as regras de negócio?

M04: O(s) elemento(s) da técnica que permite(m) representar as regras de negócio.

Q05: A técnica consegue representar os processos de negócio?

M05: O(s) elemento(s) da técnica que permite(m) representar os processos de negócio.

Q06: A técnica consegue representar os atores da organização?

M06: O(s) elemento(s) da técnica que permite(m) representar os atores da organização.

Q07: A técnica consegue representar os recursos da organização?

M07: O(s) elemento(s) da técnica que permite(m) representar os recursos da organização.

Q08: A técnica consegue representar as tarefas da organização?

M08: O(s) elemento(s) da técnica que permite(m) representar as tarefas da organização.

Q09: A técnica possui uma notação gráfica facilmente compreensível pelos *stakeholders*?

- M09:** As respostas coletadas de um grupo de usuários com diferentes perfis que utilizaram a técnica.
- Q10:** A técnica permite decompor um processo em subprocessos?
- M10:** O(s) elemento(s) da técnica que permite(m) decompor um processo em subprocessos.
- Q11:** A técnica possui uma descrição textual do(s) seu(s) modelo(s)?
- M11:** O(s) elemento(s) da técnica que permite(m) representar textualmente o(s) seu(s) modelo(s).
- Q12:** A técnica possui apoio em ferramentas computacionais?
- M12:** A(s) ferramenta(s) computacional(is) que fornece(m) apoio a técnica.
- Q13:** A técnica possui trabalhos já publicados onde a mesma é integrada a modelos funcionais?
- M13:** O(s) trabalho(s) publicado(s) onde o foco do estudo é a integração da técnica a modelos funcionais.
- Q14:** A técnica é de fácil aprendizado pelos *stakeholders*?
- M14:** As respostas coletadas de um grupo de usuários com diferentes perfis que utilizaram a técnica.
- Q15:** A técnica é de fácil entendimento pelos *stakeholders*?
- M15:** As respostas coletadas de um grupo de usuários com diferentes perfis que utilizaram a técnica.
- Q16:** A técnica consegue representar as relações de dependência entre os atores da organização?
- M16:** O(s) elemento(s) da técnica que permite(m) representar as dependências entre os atores da organização.
- Q17:** A técnica permite inserir (graficamente ou textualmente) um sistema computacional na modelagem da organização?
- M17:** O(s) elemento(s) da técnica que permite(m) a inserção de um sistema computacional na modelagem da organização.
- Q18:** A técnica permite representar (graficamente ou textualmente) as responsabilidades alocadas ao novo sistema computacional introduzido na organização?
- M18:** O(s) elemento(s) da técnica que permite(m) representar as responsabilidades do novo sistema computacional introduzido na organização.

Q19: A técnica propõe um processo específico, através de diretrizes, para a inserção de um sistema computacional no modelo organizacional?

M19: As diretrizes que orientam o processo de inserção de um sistema computacional no modelo organizacional.

Q20: A técnica permite relacionar requisitos não funcionais ao sistema computacional sendo inserido?

M20: O(s) elemento(s) da técnica que permite(m) relacionar requisitos não funcionais ao sistema computacional sendo inserido.

4.1.3 Utilizando as Técnicas de Modelagem Organizacional

A partir do estudo de caso do novo módulo Cotação de Compras, apresentado na Seção 4.1.1, o mesmo é modelado utilizando cada uma das técnicas apresentadas no Capítulo 3, tendo como objetivo extrair as características mais relevantes e desejáveis (no contexto da ER) de cada modelo. Porém, como citado anteriormente, a BPMN é uma notação gráfica padronizada para desenhar processos de *Workflows*, sendo assim, optou-se pela ocultação da modelagem dessa última. A seguir, inicia-se a modelagem através da técnica BPMN.

- **BPMN**

A seguir tem-se a modelagem do estudo de caso Cotação de Compras utilizando BPMN, através do diagrama apresentado na Figura 4.2.

Foram identificadas duas unidades organizacionais, o “Departamento de Compras” e os “Fornecedores”, as quais trocam informações com determinada sequencialidade através de processos específicos. Objetos de dados, como a “Lista de itens necessários”, também são transferidos entre as unidades organizacionais. No diagrama abaixo existem dois subprocessos de forma fechada, “Analisar preços dos itens” e “Processar pedido”, os quais são compostos por uma série de outras atividades que não são relevantes nessa fase da análise. Um *gateway* também foi utilizado para controlar a divergência da sequência do fluxo após o processo “Analisar preços dos itens”.

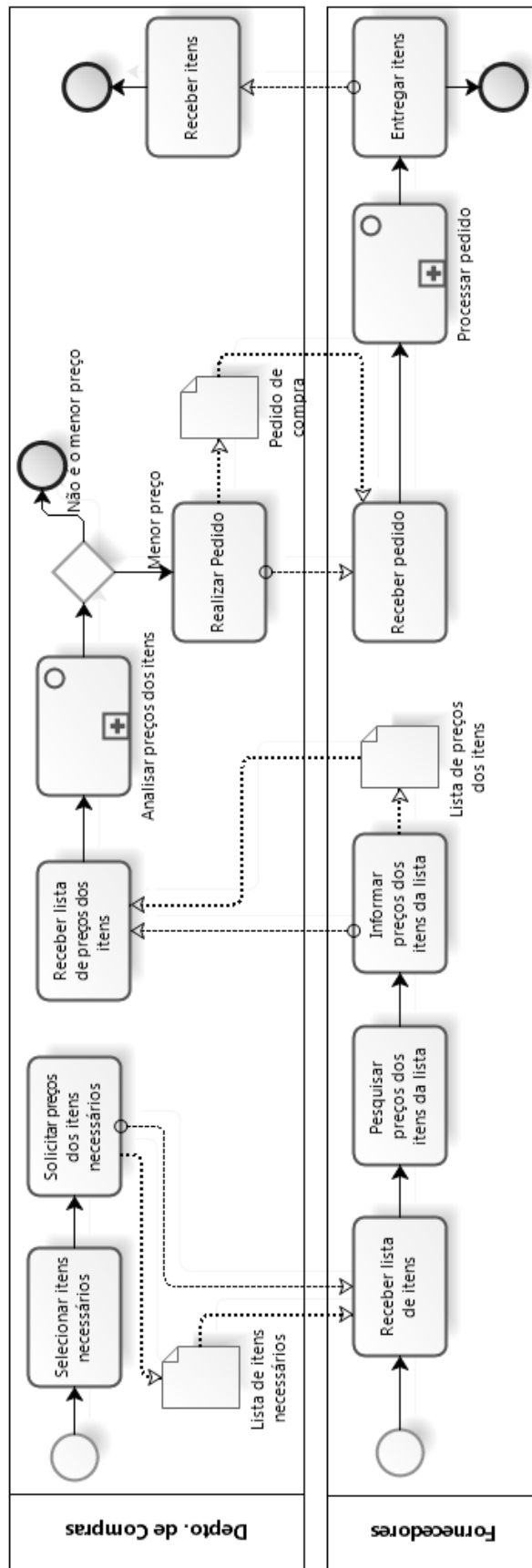


Figura 4.2: Diagrama BPMN para o módulo Cotação de Compras.

- **Técnica i***

A modelagem do novo módulo Cotação de Compras através da técnica i* pode ser visualizada na Figura 4.3 (diagrama SD) e na Figura 4.4 (diagrama SR).

Os atores identificados foram: “Departamento de Compras”, “Fornecedores” e “Setor”. Esse último foi incluído devido ao fato de a técnica i* realizar uma ampla análise da dependência entre os atores no contexto organizacional, sugerindo dessa forma a inclusão de um determinado setor da organização que solicita a compra de itens. Uma dependência de recurso pode ser identificada entre os atores “Departamento de Compras” e “Setor”, na qual o primeiro depende do recurso “Lista de itens para compra” fornecido pelo segundo. Entre os atores “Departamento de Compras” e “Fornecedores” existem quatro dependências do tipo objetivo, uma delas sendo a “Cotação de compra e pedido realizados”.

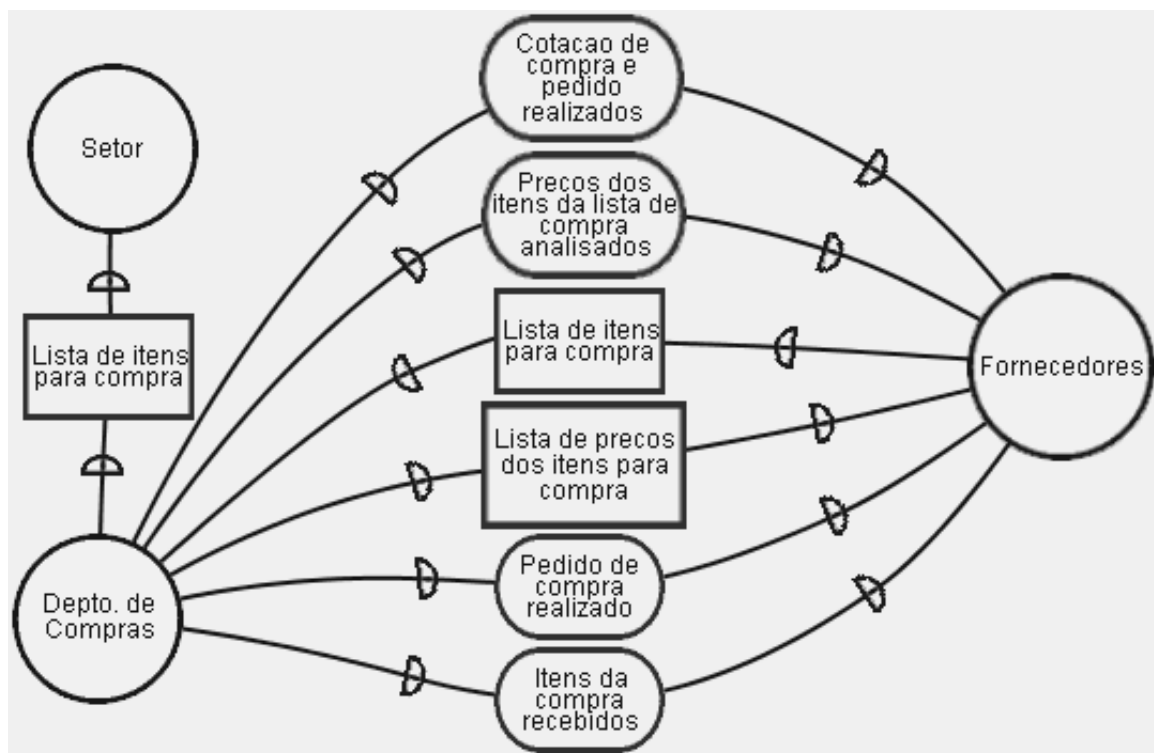


Figura 4.3: Diagrama SD para o módulo Cotação de Compras.

No diagrama SR puderam ser identificadas diversas tarefas e subtarefas internas aos atores. Exemplo disso é a tarefa “Disponibilizar serviços de orçamentos e pedidos” do ator “Fornecedores” na Figura 4.4, a qual é decomposta nas subtarefas “Disponibilizar preços dos itens da lista de compra” e “Receber pedido”.

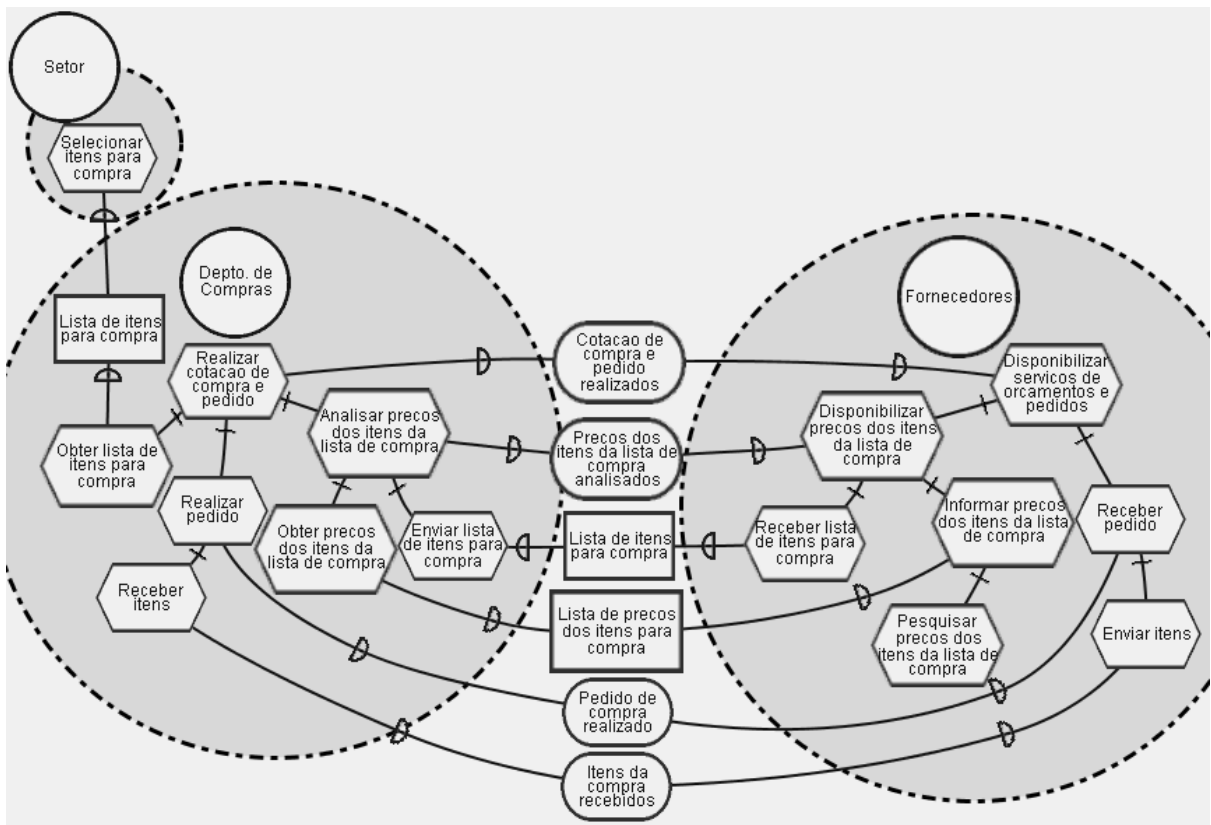


Figura 4.4: Diagrama SR para o módulo Cotação de Compras.

- **Técnica i*, com a inclusão do Sistema**

Através dos diagramas i* apresentados anteriormente (SD - Figura 4.3 e SR – Figura 4.4), pode-se inserir um sistema computacional em sua modelagem, delegando responsabilidades ao novo software introduzido na organização. Os diagramas i* com essa nova abordagem são apresentados a seguir nas Figuras 4.5 (SD) e 4.6 (SR). Agora são percebidas as dependências entre os atores “Departamento de Compras” e “Sistema”.

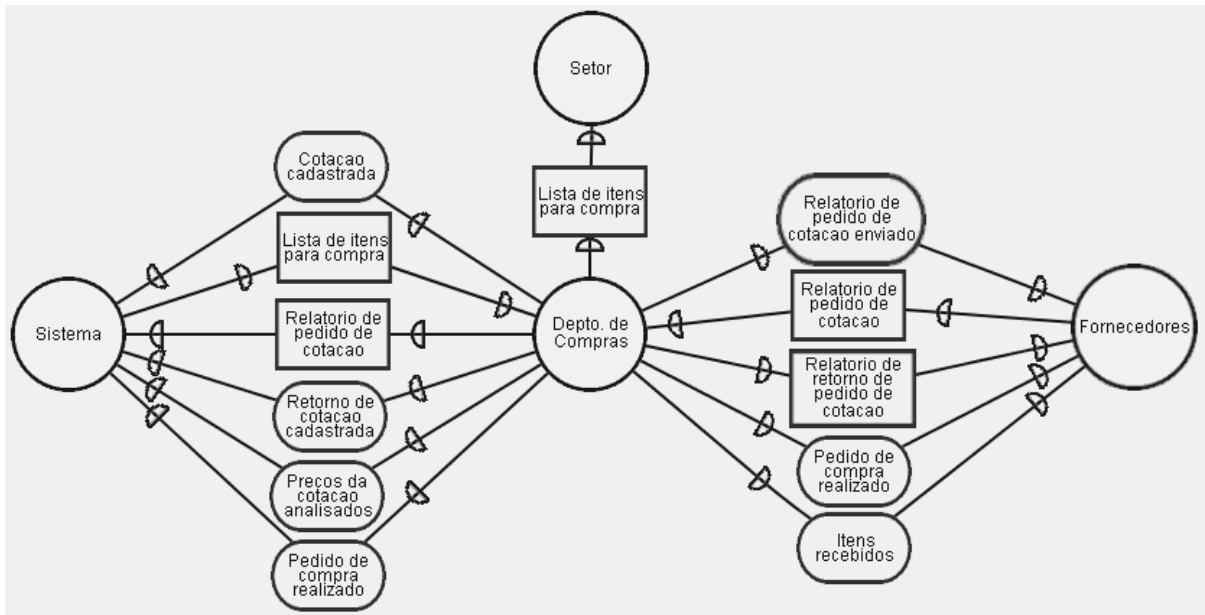


Figura 4.5: Diagrama SD para o módulo Cotação de Compras, com a inclusão do Sistema.

No diagrama SR podem ser visualizadas as tarefas internas que o Sistema será responsável em realizar para a satisfação das dependências com o Departamento de Compras, o que facilita dessa forma a identificação e especificação dos requisitos de software para esse sistema computacional.

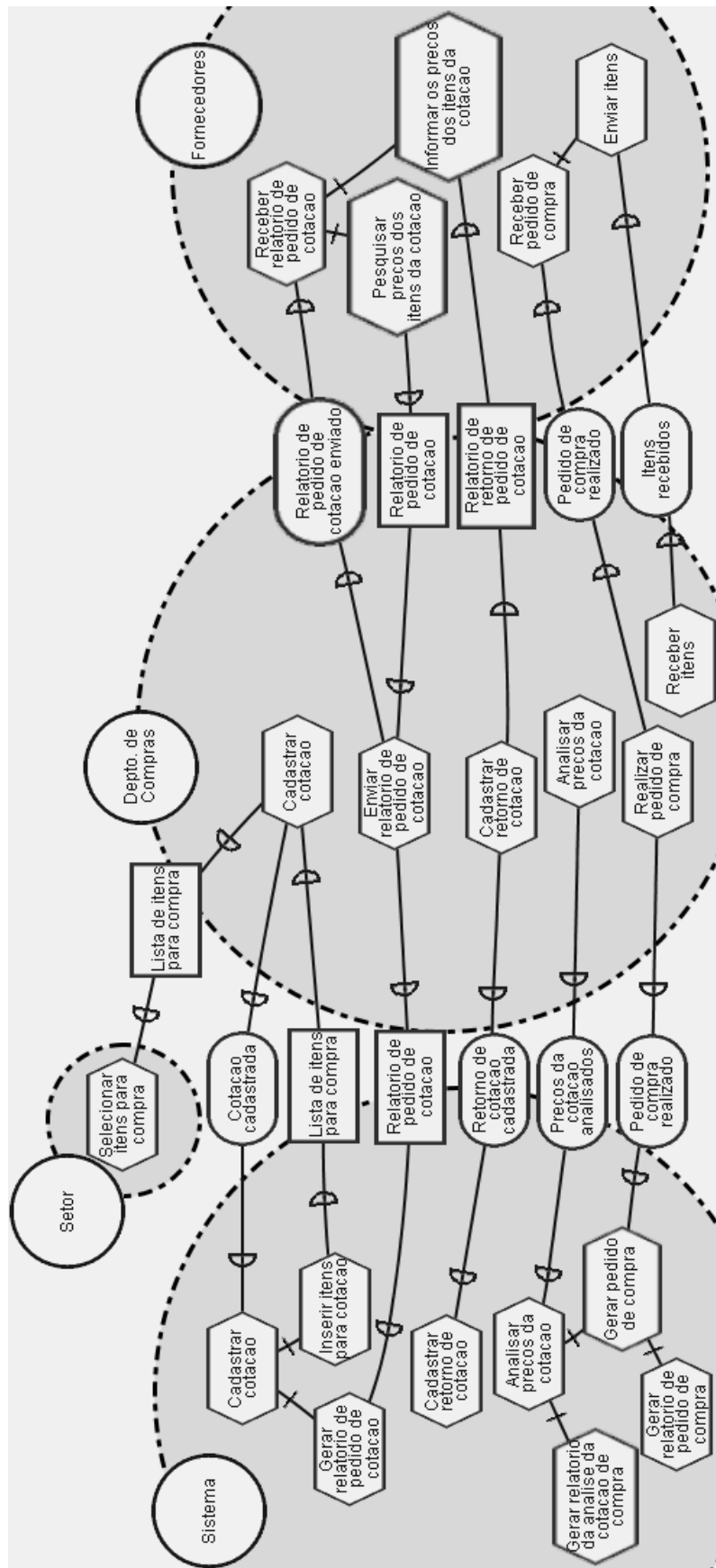


Figura 4.6: Diagrama SR para o módulo Cotação de Compras, com a inclusão do Sistema.

- **Regras de Negócio**

A utilização de Regras de Negócio para a modelagem do estudo de caso é apresentada a seguir. Essa técnica é composta apenas por uma descrição textual, não possuindo representação gráfica. Foram identificadas quatro principais regras do negócio: “Solicitar Compras de Itens”, “Solicitar Preços aos Fornecedores”, “Analisar Preços dos Fornecedores”, “Realizar Pedido de Compra”. Essa modelagem se baseia na ocorrência de um determinado evento para a regra de negócio ser processada. Uma condição então é verificada e com base nela atividades são executadas.

Regra de Negócio 01: SOLICITAR COMPRAS DE ITENS

ON (itens indisponíveis no estoque) e (necessita dos itens)

IF (não possui a lista de preços dos fornecedores)

THEN (cria uma lista com os itens necessários)

(solicita a lista de preços aos fornecedores)

chama o evento SOLICITAR PREÇOS

Regra de Negócio 02: SOLICITAR PREÇOS AOS FORNECEDORES

ON SOLICITAR PREÇOS

IF (fornecedores disponíveis) e (lista de itens necessários criada)

THEN (envia a lista de itens necessários aos fornecedores)

(aguarda o recebimento das listas dos preços de todos fornecedores)

chama o evento ANALISAR PREÇOS

Regra de Negócio 03: ANALISAR PREÇOS DOS FORNECEDORES

ON ANALISAR PREÇOS

IF (listas de preços de todos fornecedores recebidas)

THEN (cria um relatório contendo os menores preços e os respectivos fornecedores para os itens necessários)

(realiza pedido de compra com base no relatório criado)

chama o evento REALIZAR PEDIDO

Regra de Negócio 04: REALIZAR PEDIDO DE COMPRA

ON REALIZAR PEDIDO

IF (fornecedores possuem os itens disponíveis)

THEN (envia pedido de compra aos fornecedores com base no relatório de análise de preços criado)
(aguarda recebimento dos itens adquiridos)

- *Maps*

O diagrama *Map* contendo a modelagem do estudo de caso pode ser visualizada na Figura 4.7. É importante destacar que, na abordagem utilizada, quando uma determinada estratégia para atingir certo objetivo é composta por um sistema computacional, um novo diagrama *Map* deverá ser desenvolvido contendo as estratégias e objetivos internos ao software que possibilitam o alcance da meta pretendida.

Na modelagem abaixo puderam ser identificadas diversas estratégias para a obtenção dos respectivos objetivos. Um exemplo são as estratégias “Consultando o sistema” ou “Consultando o estoque” para a realização do objetivo “Selecionar itens para compra”.

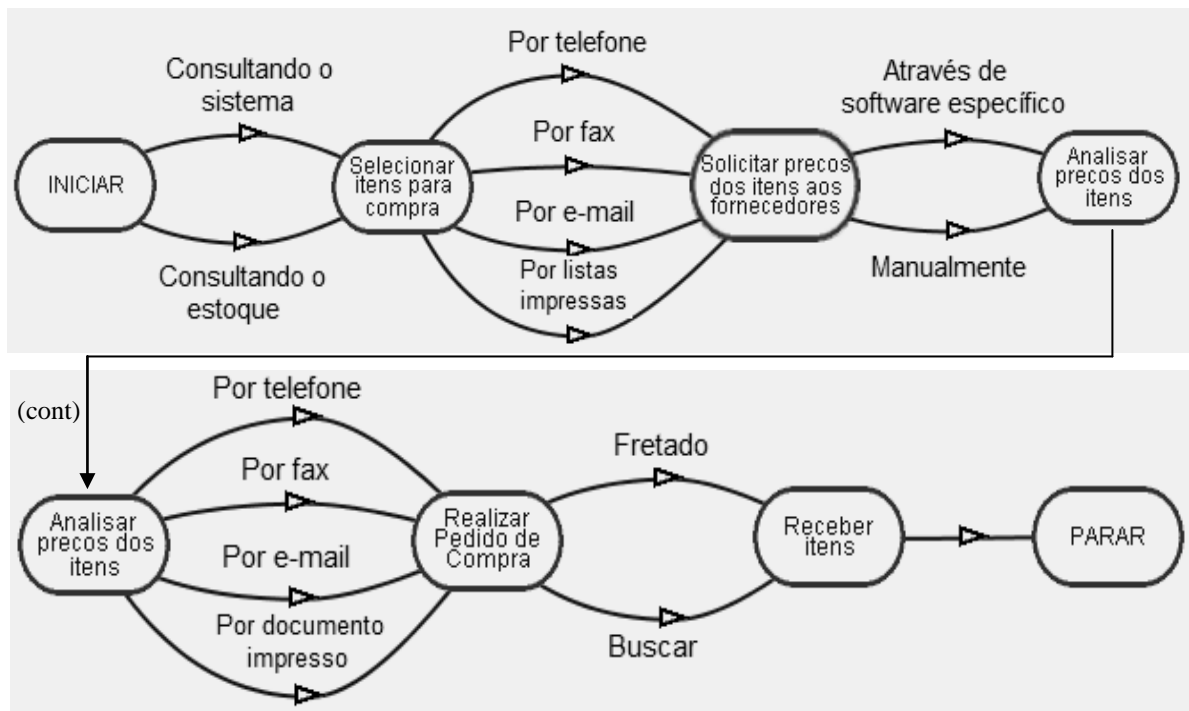


Figura 4.7: Diagrama *Map* para o módulo Cotação de Compras.

- **Maps, com a inclusão do Sistema**

Observando o diagrama *Map* inicialmente desenvolvido (Figura 4.7), pode-se destacar a utilização de um sistema computacional como estratégia para atingir o objetivo de analisar os preços dos itens. Dessa forma, um novo diagrama *Map* (Figura 4.8) pode ser criado para representar esse sistema computacional, exibindo as estratégias e objetivos internos ao software no processo de comparação dos preços informados pelos fornecedores.

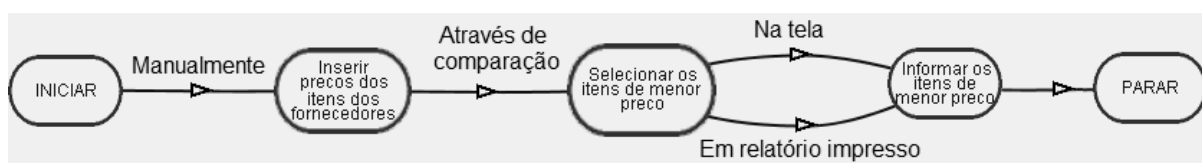


Figura 4.8: Diagrama *Map* para o objetivo Analisar Preços dos Itens.

- **Casos de Uso de Negócio**

O diagrama apresentado na Figura 4.9 representa a modelagem do sistema Cotação de Compras através de Casos de Uso de Negócio.

Nessa técnica também foram identificados dois atores principais, o Departamento de Compras e os Fornecedores. Entre os casos de uso de negócio “Selecionar itens necessários” e “Solicitar preços dos itens aos fornecedores” há um relacionamento de inclusão, significando que quando o primeiro caso de uso for realizado, o segundo também deverá ser. Já entre os casos de uso de negócio “Analisar preços dos itens” e “Realizar pedido” existe um relacionamento de extensão, o que significa que após a realização do primeiro caso de uso, opcionalmente o segundo também poderá ser realizado.

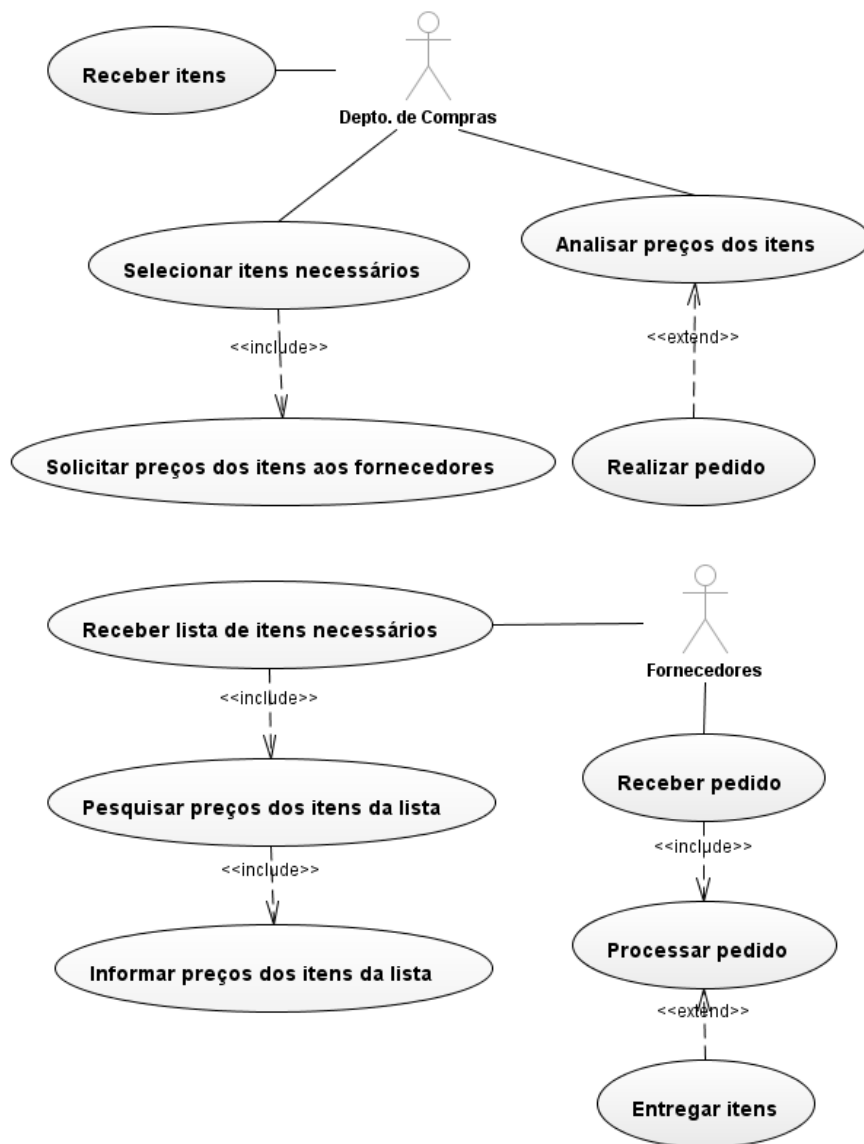


Figura 4.9: Diagrama de Casos de Uso de Negócio para o módulo Cotação de Compras.

- **Casos de Uso de Negócio, com a inclusão do Sistema**

Ao inserir um sistema computacional no diagrama de Casos de Uso de Negócio apresentado anteriormente (Figura 4.9), o mesmo sofrerá alterações apenas em seu ator Departamento de Compras, pois os processos utilizados pelos Fornecedores não serão implementados no novo sistema. Dessa forma, um diagrama de Casos de Uso de Sistema (Figura 4.10), mostrado a seguir, representa as responsabilidades do novo sistema no contexto do Departamento de Compras da organização.

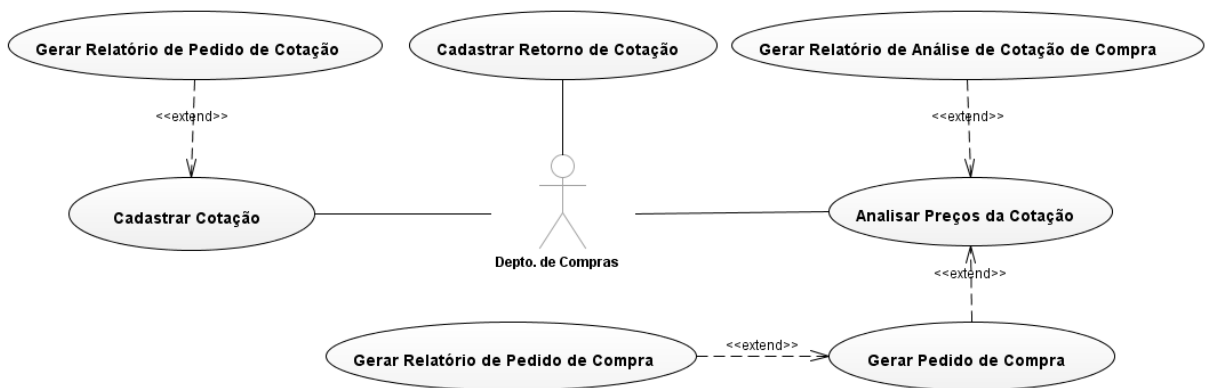


Figura 4.10: Diagrama de Casos de Uso de Sistema para o módulo Cotação de Compras.

- **EKD**

A seguir é apresentada uma série de modelos que compõem a abordagem utilizada na técnica EKD para a modelagem do estudo de caso da Cotação de Compras. O modelo de Objetivos e o modelo de Regras de Negócio foram integrados na Figura 4.11. Já o modelo de Conceitos pode ser visualizado na Figura 4.12, seguido pelo modelo de Processos de Negócio (Figura 4.13) e o modelo de Atores e Recursos (Figura 4.14). Por fim, o modelo de Requisitos e Componentes Técnicos é apresentado na Figura 4.15.

Pode-se visualizar no Modelo de Objetivos integrado com o modelo de Regras de Negócio que a regra “Só deve ser realizado pedido de compra se o item for o de menor preço entre os fornecedores” colabora positivamente para a obtenção do objetivo “Realizar pedido de compra”. Já o ponto fraco identificado “Difícil acesso às informações atualizadas dos fornecedores” dificulta a obtenção do objetivo “Analisar preços dos itens dos fornecedores”.

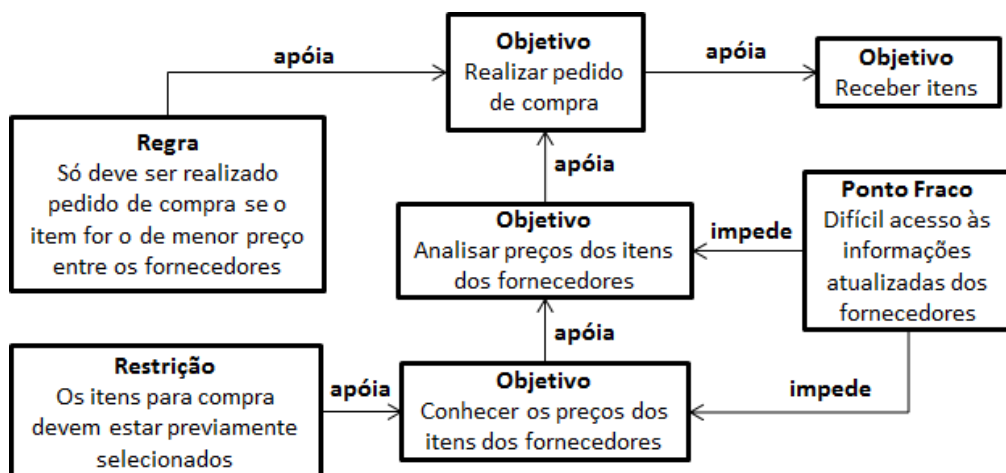


Figura 4.11: MO e MRN para o módulo Cotação de Compras.

No Modelo de Conceitos podem ser visualizados os significados do atributo “Item” e da entidade “Fornecedor”. O modelo também representa que o atributo “Item” faz parte da entidade “Fornecedor”.

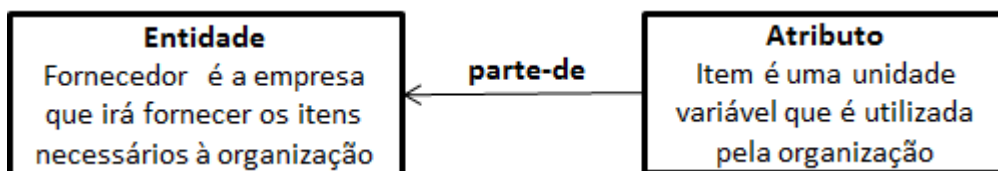


Figura 4.12: MC para o módulo Cotação de Compras

No Modelo de Processos de Negócio foram identificados três principais processos: “Solicitar preços dos itens necessários”, “Analisar preços dos itens” e “Realizar pedido de compra”. Para esses processos várias informações, tanto de entrada como de saída, podem ser visualizadas. Um exemplo, para o processo “Solicitar preços dos itens necessários”, é a entrada da informação “Lista de itens necessários” e a posterior saída da informação “Lista de preços dos itens”.

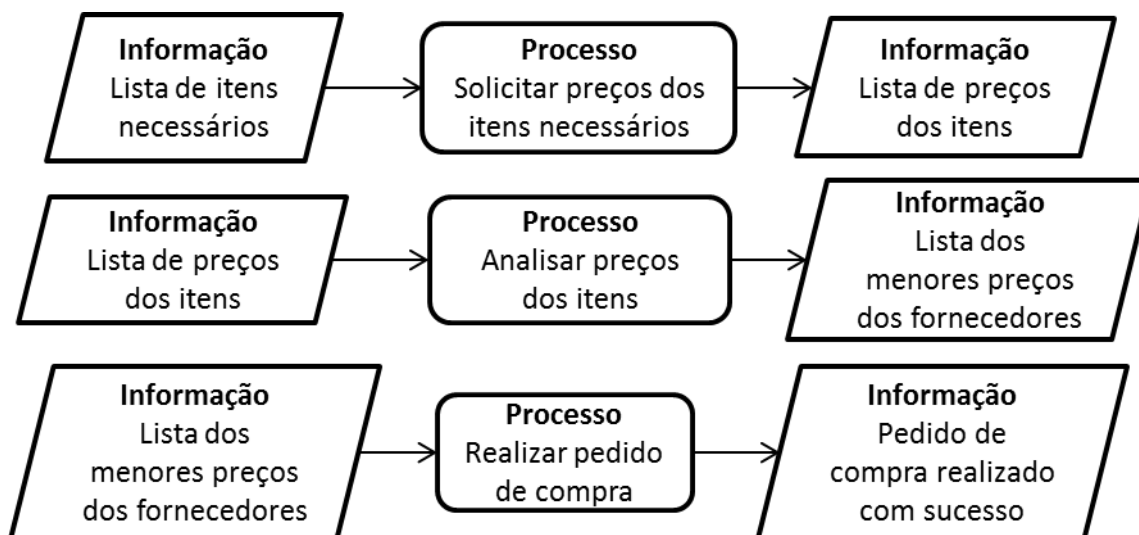


Figura 4.13: Modelo de Processos de Negócio para o módulo Cotação de Compras

No Modelo de Atores e Recursos pode-se visualizar que a unidade organizacional “Empresa” consiste de um “Departamento de Compras”, e que o “Funcionário” trabalha para a “Empresa”, no “Departamento de Compras”, e realiza compras com o “Fornecedor”.

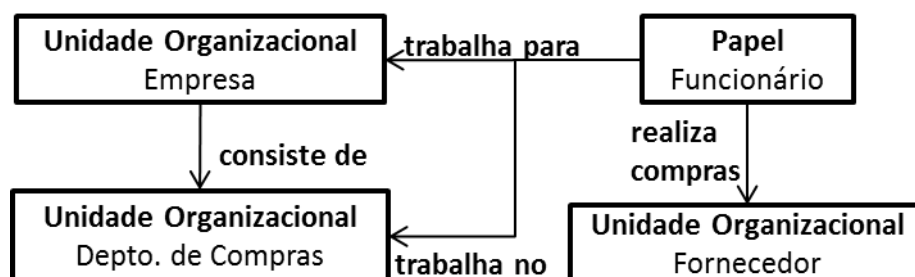


Figura 4.14: Modelo de Atores e Recursos para o módulo Cotação de Compras.

No Modelo de Requisitos e Componentes Técnicos vários objetivos para o sistema de informação foram identificados, os quais possuem diversos requisitos. O objetivo global que é “Manter todas as informações para a realização de compras” é apoiado pelo sub-objetivo “Manter uma lista de itens necessários”, que por sua vez é apoiado pelo requisito “Geração de relatório a partir da lista de itens necessários”.

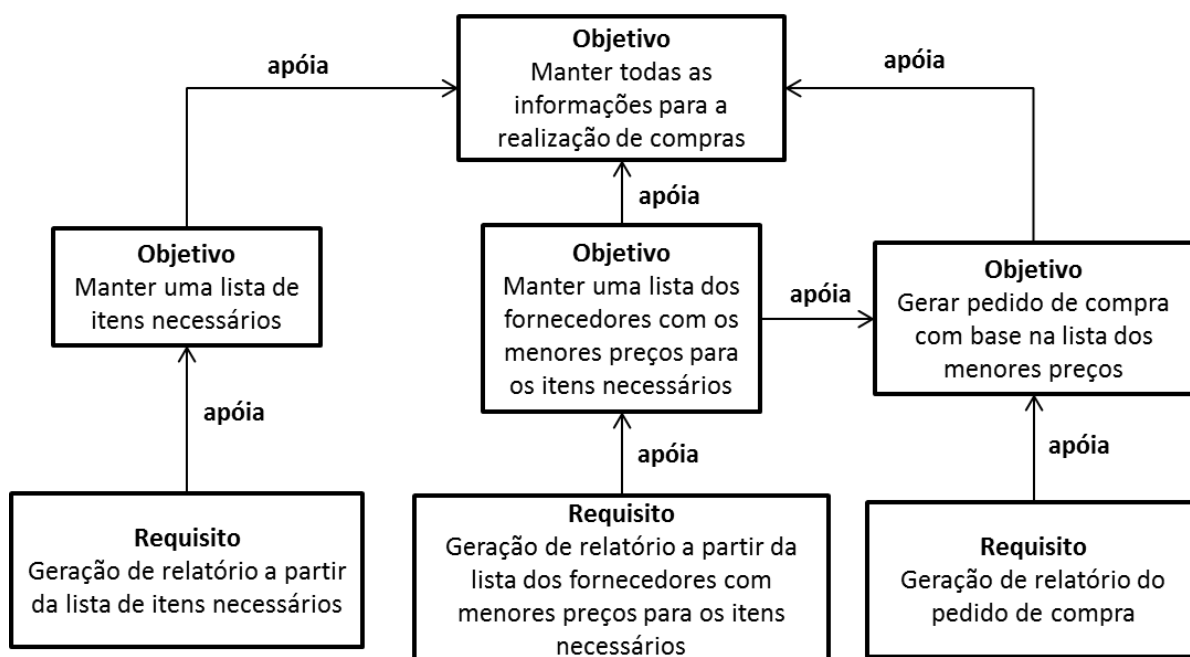


Figura 4.15: Modelo de Requisitos e Componentes Técnicos para o módulo Cotação de Compras.

4.1.4 Resultados Obtidos

Após a fase de planejamento e definição dos objetivos, questões e métricas, a coleta de dados da experimentação foi realizada. Adicionalmente, um analista de processos da empresa que forneceu o estudo de caso para a experimentação respondeu algumas das questões

formuladas, com o intuito de fornecer um *feedback* a respeito da representatividade dos requisitos computacionais a partir das técnicas estudadas.

Finalizada a coleta de dados, a próxima etapa foi a interpretação. Nela, os dados foram absorvidos, as questões definidas puderam ser respondidas e as conclusões extraídas. De posse das características de cada técnica, como mostra a Tabela 4.1, as mesmas foram analisadas e uma avaliação crítica foi elaborada (Seção 4.2).

Tabela 4.1: Características das técnicas.

CARACTERÍSTICA	EKD	BPMN	i*	Regras de Negócio	Maps	Casos de Uso de Negócio
M01 - Possui múltiplas visões	A	N	A	N	N	N
M02 - Representa diferentes aspectos da organização	A	P	A	N	N	N
M03 - Representa os objetivos da organização	A	N	A	N	A	N
M04 - Representa as regras de negócio	A	A	N	A	P	N
M05 - Representa os processos de negócio	A	A	P	N	P	A
M06 - Representa os atores da organização	A	A	A	N	N	A
M07 - Representa os recursos da organização	A	A	A	N	N	N
M08 - Representa as tarefas da organização	A	A	A	P	P	A
M09 - Notação gráfica facilmente compreensível	A	A	A	-	A	A
M10 - Permite decompor um processo em subprocessos	A	A	A	N	P	A
M11 - Possui descrição textual do(s) seu(s) modelo(s)	N	N	N	A	N	A
M12 - Possui apoio em ferramentas computacionais	N	A	A	N	N	A
M13 - Possui trabalhos já publicados onde a mesma é integrada a modelos funcionais	N	A	A	N	A	A
M14 - É de fácil aprendizado	P	A	A	A	A	A
M15 - É de fácil entendimento	P	A	P	P	A	A
M16 - Representa as relações de dependência entre os atores da organização	A	P	A	N	N	N
M17 - Permite inserir (graficamente ou textualmente) um sistema computacional na modelagem da organização	N	N	A	N	P	A
M18 - Permite representar (graficamente ou textualmente) as responsabilidades alocadas ao novo sistema computacional introduzido na organização	N	N	A	N	P	A
M19 - Possui um processo específico, através de diretrizes, para a inserção de um sistema computacional no modelo organizacional	N	N	P	N	N	N
M20 - Permite relacionar requisitos não funcionais ao sistema computacional sendo inserido	N	N	A	N	N	N

Legenda: A: Aplica-se N: Não Aplica-se P: Parcialmente aplicável - : Ausente

Com base nos resultados obtidos neste estudo comparativo, na próxima seção é realizada uma avaliação das técnicas de modelagem organizacional estudadas, a fim de expressar suas principais características e qualidades no contexto de desenvolvimento de sistemas computacionais, considerando a representatividade dos requisitos de software para pequenas e médias organizações.

4.2 Avaliação

Observando os resultados alcançados no estudo comparativo realizado na Seção 4.1, conclusões acerca das técnicas de modelagem organizacional introduzidas na experimentação puderam ser obtidas.

A técnica EDK é orientada a objetivos, dispondo de múltiplas visões através de diferentes submodelos que a compõe, o que permite representar diferentes aspectos da organização, como seus objetivos, as regras de negócio, os processos de negócio (permitindo decompor processos em subprocessos), os atores envolvidos (e suas dependências), os recursos e tarefas da organização.

Segundo o analista entrevistado, os elementos gráficos utilizados em EKD podem ser considerados bastante simples, porém, quando se tratando de grandes organizações, onde muitos fatores e aspectos estão envolvidos, a construção e entendimento dos seus submodelos pode não ser uma tarefa trivial, devido à complexidade das inúmeras relações entre eles.

O aprendizado da técnica pode ser facilitado utilizando-se diretrizes que orientam todo o processo de modelagem, incluindo um conjunto de questões que apoiam a verificação das ligações entre componentes de todos os submodelos.

Infelizmente a técnica EKD não possui apoio em ferramentas computacionais específicas, dificultando a automação do processo de construção dos seus submodelos, nem possui uma descrição textual para melhor compreensão pelos *stakeholders*.

Apesar de ser uma técnica considerada bastante completa, EKD não permite inserir um sistema computacional na modelagem da organização, o que dificulta a detecção e alocação das responsabilidades do sistema introduzido na mesma. Também não foram encontrados na literatura trabalhos onde a técnica EKD é integrada a modelos funcionais.

A técnica BPMN é conceitualmente orientada a modelagem de processos de negócio, permitindo representar as unidades organizacionais envolvidas (e implicitamente suas dependências), os recursos, tarefas e regras de negócio de um determinado processo,

possibilitando também a divisão desse processo em subprocessos. Porém, a técnica não é capaz de representar os objetivos da organização, nem fornecer diferentes visões da mesma.

Do ponto de vista do analista entrevistado, as categorias de elementos básicos da técnica BPMN possuem uma notação gráfica simples, sendo considerada uma técnica de fácil aprendizado e entendimento por sustentar a noção de fluxogramas, mesmo sem possuir uma descrição textual dos seus diagramas. Além do mais, existem várias ferramentas computacionais específicas que auxiliam no processo de construção dos seus diagramas.

Alguns trabalhos, como em Vara *et al.* (2009), propõem integrar a técnica BPMN a modelos funcionais. Nesse contexto, os requisitos funcionais são elicitados e especificados através de diagramas de processos de negócio.

No entanto, a técnica BPMN também não permite a inserção de um sistema computacional na modelagem da organização, resultando em uma maior dificuldade na identificação das funcionalidades que vão ser alocadas ao novo software sendo inserido.

A técnica i^* é orientada a atores, tratando o relacionamento de dependência entre os mesmos. É composta por dois modelos, onde podem ser representados, além dos atores, os objetivos, as tarefas e os recursos da organização. A técnica não considera as regras de negócio envolvidas e representa apenas parcialmente os processos de negócio, faltando-lhe uma característica que atribua uma determinada sequencialidade das atividades na execução das suas tarefas e subtarefas.

O analista entrevistado destacou que a notação gráfica empregada em i^* é bastante simples e de fácil aprendizado. No entanto, o entendimento dos seus modelos pode não ser uma tarefa tão simples assim, devido ao fato de os mesmos possuírem diversas ligações e não fornecerem a ideia de fluxo de tarefas.

Os modelos em i^* não possuem descrição textual, porém existem diversas ferramentas computacionais específicas que apoiam a construção dos mesmos. Além do mais, existem vários trabalhos onde a técnica i^* é integrada a modelos funcionais. Yu *et al.* (2011) apresentam a proposta original da técnica i^* , bem como trabalhos que aplicam, adaptam, estendem ou avaliam a sua abordagem.

O grande diferencial da técnica i^* perante as outras é a possibilidade de inserção de um sistema computacional na sua modelagem. Dessa forma, o sistema passa a ser modelado como um novo ator dentro da organização, podendo ser alocadas a ele responsabilidades específicas nesse contexto, facilitando a identificação tanto de requisitos funcionais como não funcionais.

A modelagem através de Regras de Negócio é bem restrita. A técnica não possui diversas características que são desejáveis no contexto de desenvolvimento de software, conseguindo representar através de uma descrição textual (não possui representação gráfica) apenas as regras de negócio propriamente ditas e parcialmente as tarefas da organização.

Na visão do analista entrevistado, a técnica baseada em Regras de Negócio pode ser considerada de fácil aprendizado. No entanto, o seu entendimento é prejudicado por não possuir uma notação gráfica que permita a visualização das regras e as suas interações. Dessa forma, o analista sugeriu que a técnica fosse complementada através de um diagrama que consiga expressar de maneira clara e concisa as regras de negócio envolvidas na organização.

Objetivos, processos, atores e recursos da organização não são cobertos por essa abordagem, tão pouco existe a possibilidade de a técnica representar um sistema computacional sendo inserido na organização, o que dificulta muito a identificação e captura dos requisitos de software para esse sistema. Ferramentas computacionais específicas que auxiliem no processo de construção das regras de negócio não foram encontradas, nem trabalhos onde regras de negócio são integradas a modelos funcionais.

A técnica *Maps* é dirigida a objetivos e permite, de forma simples, porém expressiva, representar os processos da organização em termos de objetivos e estratégias. Apesar disso, a técnica não possui múltiplas visões, e dessa forma não consegue representar diferentes aspectos da organização, como por exemplo, os atores (e suas dependências) e recursos.

Segundo o analista entrevistado, diagramas *Maps* possuem uma notação gráfica extremamente simples, e, devido ao fato de o escopo da técnica ser bem limitado, a mesma é de fácil aprendizado e entendimento, dispensando qualquer descrição textual dos seus modelos. Apesar da simplicidade, não foram encontradas ferramentas computacionais específicas que guiassem o processo de construção de diagramas *Maps*.

Na abordagem utilizada, a inserção de um sistema computacional na modelagem da organização só será realizada quando, em um diagrama *Map*, existir uma estratégia que utilize o sistema para atingir determinado objetivo. Dessa maneira, um novo diagrama *Map* é construído para representar esse sistema computacional, exibindo as estratégias e objetivos internos ao software para atingir o objetivo inicial.

No entanto, não existem diretrizes específicas que apoiam essa inserção do sistema computacional, tão pouco a técnica permite relacionar requisitos não funcionais ao software

sendo inserido. Kaabi, Souveyet e Rolland (2004) propõem derivar funcionalidades do sistema computacional pretendido pela organização a partir da técnica *Maps*.

Casos de Uso de Negócio é uma maneira natural de identificar os processos de negócio, os quais podem ser divididos em subprocessos. A técnica possui uma notação gráfica simples, onde é possível representar os atores e as tarefas da organização, porém não é capaz de expressar as dependências entre esses atores e os diferentes aspectos da organização, como objetivos, regras de negócio e os recursos.

Através de uma descrição textual detalhada dos seus diagramas, onde os fluxos de atividades do cenário são apresentados, Casos de Uso de Negócio proporciona um bom entendimento por parte dos *stakeholders*.

Apesar disso, uma ressalva feita por parte do analista entrevistado cita que diversos clientes possuem dificuldade na compreensão dos significados dos estereótipos utilizados na técnica. Como exemplo, pode-se citar os estereótipos <<*include*>> e <<*extend*>>, os quais possuem significado de utilização para incluir um comportamento comum de um caso de uso incluído para um caso de uso base, e utilização para incluir um comportamento opcional de um caso de uso extensor para um caso de uso estendido, respectivamente.

O processo de aprendizado da técnica não necessita de muito esforço, além de existirem diversas ferramentas computacionais específicas que auxiliam o processo de construção dos diagramas.

A inserção de um sistema computacional na modelagem da organização utilizando Casos de Uso de Negócio se dá através da mudança do escopo dos Casos de Uso, partindo do modelo de negócios para o modelo de sistema. Agora, diversas responsabilidades poderão ser alocadas ao ator que irá interagir com o software, as quais representarão os requisitos funcionais no contexto de desenvolvimento desse sistema.

Os requisitos não funcionais não são representados pela técnica, e infelizmente, não há um processo específico, através de diretrizes, para guiar o processo de inserção de um sistema computacional no modelo organizacional.

A abordagem para a modelagem de negócios apresentada na metodologia de desenvolvimento de software *Rational Unified Process* (RUP) inclui uma maneira concisa e simples de gerar requisitos de software a partir dos Casos de Uso de Negócio.

De maneira geral, verifica-se na Tabela 4.1 que as técnicas de modelagem organizacional que mais se destacam, segundo as características analisadas, são: EKD, BPMN, i* e Casos de

Uso de Negócio. Já os modelos utilizando Regras de Negócio e *Maps* se mostraram mais restritos, não conseguindo representar diversas características importantes no contexto organizacional, como os atores da organização e os diferentes aspectos da mesma.

O principal fator negativo das técnicas EKD e BPMN, no contexto da avaliação, é o fato de ambas não fornecerem apoio na inclusão de um sistema computacional em seus modelos. Apesar disso, EKD pode ser considerada uma técnica de modelagem organizacional bastante completa, conseguindo representar a maioria das características avaliadas. Já BPMN se mostrou uma ótima ferramenta para modelar os processos de negócio da organização.

As técnicas *i** e Casos de Uso de Negócio foram os modelos que mais apresentaram as diferentes características desejadas considerando o desenvolvimento de sistemas computacionais e que foram avaliadas no estudo experimental. Ambas as técnicas conseguem representar os atores e as tarefas da organização, suas notações gráficas são simples, diversas ferramentas computacionais auxiliam a criação de seus modelos, possuem trabalhos onde as mesmas são integradas a modelos funcionais, e, principalmente, fornecem apoio na inserção do sistema computacional em suas modelagens.

Os pontos negativos da técnica *i** são: a impossibilidade de a mesma representar as regras de negócio do ambiente organizacional e a falta de uma característica que contribua com uma determinada sequencialidade na execução das suas tarefas e subtarefas. Já Casos de Uso de Negócio não consegue representar os objetivos e recursos da organização, e as dependências entre seus atores.

Sendo assim, *i** pode ser considerada a técnica que mais se destacou na avaliação realizada, pois permite, de forma abrangente, representar vários aspectos do contexto organizacional, como os atores envolvidos e suas dependências, objetivos, recursos e tarefas, além de possibilitar a alocação de responsabilidades ao sistema computacional inserido como um novo ator na modelagem da organização, permitindo representar tanto requisitos funcionais como não funcionais desse novo sistema. Além disso, inúmeros trabalhos auxiliam a utilização da técnica *i** fornecendo diretrizes para a integração da mesma a modelos funcionais (Yu *et al.*, 2011).

4.3 Considerações Finais

Este capítulo apresentou um estudo comparativo entre as técnicas de modelagem organizacional estudadas no Capítulo 3, e uma posterior avaliação desses modelos.

Para o estudo comparativo foi utilizado um estudo de caso real fornecido por uma empresa de desenvolvimento de software na área de concessionários de veículos e implementos agrícolas da região oeste do Paraná. Esse estudo de caso se contextualiza através de um dos principais produtos da organização, um sistema de gerenciamento comercial e de concessionárias, o qual necessitou uma nova funcionalidade no seu sistema de compras, visando auxiliar na determinação do menor preço para efetuar a compra de itens.

Com o intuito de que o estudo comparativo fornecesse resultados válidos, o mesmo foi organizado, controlado e acompanhado por princípios introduzidos pela Engenharia de Software Experimental, destacados no Capítulo 2. Dessa forma, a estrutura da experimentação através da abordagem GQM foi composta da seguinte maneira: o objetivo do estudo é analisar as técnicas de modelagem organizacional, com a finalidade de comparar as suas características, com respeito a representatividade dos requisitos organizacionais, do ponto de vista dos *stakeholders*, no contexto de pequenas e médias organizações. Ainda através da metodologia GQM, diversas questões e métricas foram estipuladas, visando avaliar as diferentes características desejáveis para um modelo organizacional no contexto de desenvolvimento de sistemas computacionais.

A partir do estudo de caso fornecido e da estrutura de experimentação criada, o experimento então pôde ser realizado. Para isso, o estudo de caso foi modelado conforme cada uma das diferentes técnicas de modelagem organizacional, exceto *Workflows*, pois foi optado pela sua ocultação devido ao fato de que a técnica BPMN fornece uma notação gráfica padronizada para desenhar processos de *workflows*.

Após a utilização das técnicas, a coleta de dados da experimentação foi realizada, o que incluiu uma entrevista ao analista de processos da empresa que forneceu o estudo de caso. Após a obtenção dos resultados, esses foram disponibilizados em uma tabela informando as características encontradas em cada técnica.

Concluído o estudo comparativo, uma avaliação crítica dos modelos organizacionais foi abordada. Nela, pôde-se concluir que as técnicas de modelagem organizacional que mais se destacaram, segundo as características analisadas, foram: EKD, BPMN, i* e Casos de Uso de Negócio. Já os modelos utilizando Regras de Negócio e *Maps* se mostraram mais restritos.

O fato de não fornecer apoio na inclusão de um sistema computacional na sua modelagem acabou prejudicando a avaliação das técnicas EKD e BPMN. Mesmo assim, EKD foi

considerada uma técnica de modelagem organizacional bastante completa e BPMN uma ótima ferramenta para modelar processos de negócio da organização.

Em relação às técnicas *i** e Casos de Uso de Negócio, as mesmas foram as que mais apresentaram as diferentes características desejadas considerando o desenvolvimento de sistemas computacionais e que foram avaliadas no estudo experimental.

Os pontos negativos da técnica *i** é a impossibilidade da mesma representar as regras de negócio do ambiente organizacional e falta de sequencialidade na execução das suas tarefas. Já Casos de Uso de Negócio não consegue representar os objetivos e recursos da organização, e as dependências entre seus atores.

Em resumo, *i** foi considerada a técnica que mais se destacou na avaliação realizada, pois permite representar diferentes aspectos da organização e também possibilita a alocação de responsabilidades ao sistema computacional inserido como um novo ator na modelagem da organização. Outro fator positivo são os diversos trabalhos que auxiliam a utilização da técnica *i** através de diretrizes que permitem a integração da mesma a modelos funcionais (Yu *et al.*, 2011).

Capítulo 5

Considerações Finais

Apresentou-se inicialmente neste trabalho uma breve retrospectiva dos principais conceitos da Engenharia de Requisitos, subárea da Engenharia de Software, introduzindo também algumas definições a respeito da Engenharia de Software Experimental, disciplina muito importante para a ES, onde há uma compreensão cada vez maior na comunidade de que os estudos empíricos são necessários para avaliar, desenvolver ou melhorar processos, métodos e ferramentas para desenvolvimento de software e manutenção dos mesmos.

Com relação à Engenharia de Software Experimental, foi estudada a metodologia de experimentação GQM (*Goal/Question/Metric*), uma abordagem orientada a metas e utilizada em Engenharia de Software para a medição de produtos e processos de software. Essa abordagem apresenta um mecanismo eficaz para planejamento, definição de objetivos da medição e avaliação.

Posteriormente, foi apresentada uma visão a respeito da fase de modelagem organizacional no contexto de desenvolvimento de sistemas computacionais. A representação do conhecimento sobre processos organizacionais e relacionamentos entre os membros de uma organização são muito úteis para uma adequada compreensão da mesma, resultando na elicitação mais efetiva e consistente dos requisitos de sistemas computacionais pretendidos.

Dessa forma, a partir da compreensão da estrutura e ambiente organizacional, os engenheiros de requisitos são capazes de definir como o sistema computacional pretendido conseguirá satisfazer os objetivos da empresa, por que ele é necessário, quais as alternativas existentes, quais as implicações das alternativas para as várias partes interessadas, entre outros aspectos.

Para tanto, diversas técnicas de modelagem organizacional foram abordadas, como: EKD, BPMN, i*, *Workflows*, Regras de Negócio, *Maps* e Casos de Uso de Negócio. Destaca-se que

vários elementos nesses modelos podem ser utilizados e mapeados para requisitos funcionais e não funcionais do sistema pretendido.

Sendo assim, um experimento comparativo entre essas diferentes técnicas de modelagem organizacional foi realizado, visando avaliar as características e qualidades existentes nos modelos e no processo de desenvolvimento de software. Para este fim, uma empresa desenvolvedora de software da região oeste do Paraná, na área de concessionárias de veículos e implementos agrícolas, forneceu um estudo de caso real que envolveu o desenvolvimento de um novo módulo para o sistema de compras de um software de gerenciamento comercial e de concessionárias, o qual foi modelado conforme cada uma das diferentes técnicas estudadas, com exceção de *Workflows*, pois foi optado pela sua ocultação devido ao fato de que a técnica BPMN fornece uma notação gráfica padronizada para desenhar processos de *workflows*.

Com o intuito de que o estudo comparativo fornecesse resultados válidos, o mesmo foi organizado, controlado e acompanhado por princípios introduzidos pela Engenharia de Software Experimental, incluindo a abordagem GQM.

Após a utilização das técnicas, a coleta de dados da experimentação foi realizada, o que incluiu uma entrevista ao analista de processos da empresa que forneceu o estudo de caso. Após a obtenção dos resultados, esses foram disponibilizados em uma tabela informando as características encontradas em cada técnica.

Com base nesse experimento, foi realizada uma avaliação crítica a respeito das técnicas de modelagem organizacional estudadas. Conseqüentemente, pode-se concluir que as técnicas que mais se destacaram, segundo as características analisadas, foram: EKD, BPMN, *i** e Casos de Uso de Negócio. Já os modelos utilizando Regras de Negócio e *Maps* foram os mais restritos.

Uma característica muito importante no contexto da avaliação das técnicas foi o fato de o modelo permitir ou não a inclusão de um sistema computacional na sua modelagem. Isso implicou pontos positivos para as técnicas *i** e Casos de Uso de Negócio, por permitirem, e negativos para EKD e BPMN, por não permitirem. Mesmo assim, EKD foi considerada uma técnica de modelagem organizacional bastante completa e BPMN uma grande ferramenta para modelar os processos de negócio da organização.

Já em relação às técnicas *i** e Casos de Uso de Negócio, as mesmas foram as que mais apresentaram as diferentes características desejadas considerando o desenvolvimento de sistemas computacionais e que foram avaliadas no estudo experimental. Infelizmente, Casos

de Uso de Negócio não consegue representar os objetivos e recursos da organização, e as dependências entre seus atores, características essenciais para uma melhor compreensão do ambiente organizacional.

Por fim, i^* foi considerada a técnica que mais se destacou na avaliação realizada, pois permitiu representar diferentes aspectos da organização e também possibilitou a alocação de responsabilidades ao sistema computacional inserido como um novo ator na modelagem do ambiente organizacional. Outro fator positivo são os diversos trabalhos que auxiliam a utilização da técnica i^* através de diretrizes que permitem a integração da mesma a modelos funcionais, apresentados em (Yu *et al.*, 2011).

Pode-se citar como deficiência da técnica i^* o fato de não conseguir representar as regras de negócio envolvidas e representar apenas parcialmente os processos de negócio, faltando-lhe uma característica que atribua determinada sequencialidade das atividades na execução das suas tarefas e subtarefas.

Como trabalhos futuros, pretende-se estender a avaliação comparativa das técnicas a outros estudos de caso, em empresas de maior porte. Também deseja-se desenvolver um estudo que possibilite a integração de duas ou mais técnicas aqui estudadas, bem como formalizar a abordagem utilizada na técnica *Maps*, onde um novo diagrama *Map* deverá ser desenvolvido quando uma determinada estratégia para atingir certo objetivo é composta por um sistema computacional. Esse novo *Map* deverá conter as estratégias e objetivos internos ao software que possibilitam o alcance da meta pretendida. Por fim, pensa-se na realização de um estudo mais aprofundado a respeito da técnica EKD, com o objetivo de definir diretrizes que permitam o mapeamento de um sistema computacional na sua modelagem.

Referências Bibliográficas

- ALENCAR, F. M. R. *Mapeando a Modelagem Organizacional em Especificações Precisas*. Tese (Tese de Doutorado) – Engenharia de Software – Centro de Informática – Universidade Federal de Pernambuco, Recife, PE, Dezembro, 1999.
- ANTON, A. Goal Based Requirements Analysis. In: *SECOND INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING – ICRE'96*, 1996. Proceedings..., Colorado Springs, Colorado, USA: [s.n.], 1996, p. 136–144.
- BASILI, V. R. The Experimental Paradigm in Software Engineering. In: *INTERNATIONAL WORKSHOP ON EXPERIMENTAL SOFTWARE ENGINEERING ISSUES: CRITICAL ASSESSMENT AND FUTURE DIRECTIONS*, 1992. Proceedings..., Dagstuhl Castle, Germany: Kindle Edition, 1992, p. 3-12.
- BASILI, V. R. The Role of Experimentation in Software Engineering: Past, Current, and Future. In: *18th INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING – ICSE 1996*, 1996. Proceedings..., Berlin, Germany: IEEE Computer Society, 1996, p. 442-449.
- BASILI, V., CALDEIRA, G., ROMBACH, H. Encyclopedia of Software Engineering, capítulo: Experience Factory, p. 469-479, John Wiley & Sons, v. 1, 1994.
- BASILI, V. R., ROMBACH, D., The TAME Project: Towards Improvement-Oriented Software Environments. *IEEE Transactions on Software Engineering*, v. 14, n. 6, p. 758- 773, June, 1988.
- BASILI, V. R., SELBY, R. W., HUTCHENS, D. H. Experimentation in Software Engineering. *IEEE Transactions on Software Engineering*, v. 12, n. 7, p. 733-743, July, 1986.
- BASILI, V. R., ROMBACH, D., SCHNEIDER, K., KITCHENHAM, B., PFAHL, D., SELBY, R. W. *Empirical Software Engineering Issues: Critical Assessment and Future Directions*. 1 ed., Germany: Springer, 2006.

- BIDER, I., JOHANNESSON, P. Tutorial on: Modeling Dynamics of Business Processes – Key for Building Next Generation of Business Information Systems. In: *THE 21st INTERNATIONAL CONFERENCE ON CONCEPTUAL MODELING - ER2002*, 2002. Proceedings..., Tampere, FL: [s.n.], 2002, p. 7-11.
- BOEHM, B. W. *Software Risk Management*. 1 ed., Washington: IEEE Computer Society Press, 1989.
- BOOCH, G., JACOBSON, I., RUMBAUGH, J. *The Unified Modeling Language Reference Manual*. 1 ed., Addison-Wesley, 1999.
- BPMI. *Business Process Management Initiative*. <http://www.bpmi.org>, Consultado na Internet em: 20/05/2010.
- BPMN. *Business Process Modeling Notation*. <http://www.bpmn.org>, Consultado na Internet em: 20/05/2010a.
- BPMN. *Business Process Modeling Notation*. <http://www.omg.org/spec/BPMN/1.2>, Consultado na Internet em: 20/05/2010b.
- BRG. *GUIDE Business Rules Project*. Final Report, Revisão 1.2, Oct, 1997, <http://www.businessrulesgroup.org>. Consultado na Internet em: 20/05/2010.
- BUBENKO JR., J. A., KIRIKOVA, M. Enterprise modelling: improving the quality of requirements specification. In: *IRIS-17 INFORMATION SYSTEMS RESEARCH SEMINAR IN SCANDINAVA*, 1994. Proceedings..., Oulu: [s.n.], 1994.
- BUBENKO JR, J. A., STIRNA, J., BRASH, D. *EKD user guide*. Royal Institute of Technology, Dpt of computer and systems sciences, Stockholm, 1998.
- BUBENKO JR, J. A., WANGLER, B. Objectives driven capture of business rules and information systems requirements. In: *INTERNATIONAL CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS*, 1993. Proceedings..., Le Touquet: [s.n.], 1993, p. 670-677.
- CANÓS, J. H., PENADÉS, M. C., CARSÍ, J. A. From Software Processes to Workflow Processes: the Workflow Lifecycle, In: *INTERNATIONAL SOFTWARE TECHNOLOGY WORKSHOP*, 1999. Proceedings..., Grenoble: [s.n.], 1999.
- CASTRO, J. F. B. Introdução a Engenharia de Requisitos. In: *JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA DO XIV CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO*, 1995. Proceedings..., Canela: SBC, 1995.

- CASTRO, J. F. B., ALENCAR, F. M. R., SANTANDER, V. F. A., LOURENÇO, C. T. L. Integration of i* and Object Oriented Models. In: YU, E., MYLOPOULOS, J., MAIDEN, N., GIORGINI, P., *Social Modelling for Requirements Engineering*. Cambridge, MA: MIT Press, 2010.
- CASTRO, J. F. B., KOLP, M., MYLOPOULOS, J. Towards Requirements-Driven Information Systems Engineering: The Tropos Project, *Information Systems*, Elsevier, Amsterdam, The Netherlands, 2002.
- CHICHINELLI, M. *Contribuição da Técnica de Modelagem Organizacional I* ao Processo de Engenharia de Requisitos, com destaque aos Requisitos Não Funcionais*. Dissertação (Dissertação de Mestrado) – Engenharia de Software – Universidade de São Paulo, São Carlos, SP, 2002. Dissertação.
- CHUNG, L., NIXON, B. A., YU, E., MYLOPOULOS, J. *Non-Functional Requirements in Software Engineering*. Software Engineering – Kluwer Academic Publishers. 2000. Monografia.
- CONRADI, R., BASILI, V., CARVER, J., SHULL, F., TRAVASSOS, G. *A Pragmatic Document Standards for and Experience Library: Roles, Documents, Contents, and Structure*. University of Maryland, April 2001, Technical Report CS-TR-4235.
- CRUZ NETO, G. G. *Estudos qualitativos para elicitação de requisitos: uma abordagem que integra análise sócio-cultural e modelagem organizacional*. Tese (Tese de Doutorado) – Engenharia de Software – Universidade Federal de Pernambuco, Recife, PE, 2008.
- DAVIS, A M. *Software Requirements: Objects, Functions, and States*. New Jersey: P T R Prentice Hall, Englewood Cliffs, 1993.
- DUARTE, M. A. *Workflow – Fluxo de Trabalho*, http://www.catalao.ufg.br/cc/disc/ti/wfl_aula.pdf, Consultado na Internet em: 15/06/2010.
- DUMAS, M., AALST, W. M. P., HOFSTEDE, A. H. Process-Aware Information Systems: Bridging People and Software Through Process Technology. *Wiley-Interscience*, 2005.
- ERIKSSON, H. E., PENKER, M. *Business Modeling With UML*. New York: John Wiley & Sons, 2000.

- ESTRADA, H., MARTÍNEZ, A., PASTOR, O., ORTIZ, J., RÍOS, O. A. Generación Automática de un Esquema Conceptual OO a Partir de un Modelo Conceptual de Flujo de Trabajo. In: *ANAIS DO WER01 – WORKSHOP EM ENGENHARIA DE REQUISITOS*, 2001. Proceedings..., Buenos Aires: [s.n.], 2001, p. 223-245.
- ESTRADA, H., MARTÍNEZ, A., PASTOR, O., SÁNCHEZ, J. Generación de Especificaciones de Requisitos de Software a partir de Modelos de Negocios: un enfoque basado en metas, In: *ANAIS DO WER02 – WORKSHOP EM ENGENHARIA DE REQUISITOS*, 2002. Proceedings..., Valencia: [s.n.], 2002, p. 11-12.
- FENTON, N. How Effective Are Software Engineering Methods?. *Journal Systems and Software*, v. 22, n. 2, p. 141-146, 1993.
- FISCHER, L. *Workflow Handbook*. WFMC, 2005.
- GEORGAKOPOULOS, D., HORNICK, M., SHETH, A. An Overview of Workflow Management: From Process Modeling to Workflow Automation. *Distributed and Parallel Databases*, n. 3, p. 119-153, Mar. 1995.
- GOMES, A., OLIVEIRA, K. M., ROCHA, A. R. *Métricas para Medição e Melhoria de Processo de Software*. COPPE/UFRJ - Programa de Engenharia de Sistemas e Computação, Rio de Janeiro, 2001.
- GOTTESDIENER, E. Business rules show Power, promise. *Application Development Trends*, v. 4, n. 3, p. 36-43, March, 1997.
- GRESSE, C. B., HOISL, J. W. *A Process Model for GQM - Based Measurement*. Kaiserslautern, Germany: University of Kaiserslautern, Department of Computer Science, 1995. Technical Report STTI-95-04-E.
- HAMMER, M., CHAMPY, J. *Reengineering the Corporation: A Manifesto for Business Revolution*. HarperBusiness, 1993.
- HERBST, H. Business Rules in system analysis: a meta-model and repository system. *Information Systems*, v. 21, n. 2, p. 147-166, 1996.
- IEEE. *IEEE Software Engineering Standards Collection*. Computer Society Press, 1997.
- IEEE. Std. 830 *IEEE Guide to Software Requirements Specification*. The Institute of Electrical and Electronics Engineers, New York, EUA, 1984.
- ISO. *IEC 9126 Standard for information technology: software products evaluation: quality characteristics and guidelines for theirs use*. Geneva, 1988.

- JACOBSON, I. *Object Oriented Software Engineering: A Use Case Driven Approach*. Addison-Wesley, 1992.
- JACOBSON, I., BOOCH, G., RUMBAUGH, J. *The Unified Software Development Process*. Addison-Wesley, 1999.
- JACOBSON, I., ERICSON, M., JACOBSON, A. *The Object Technology*. Addison-Wesley, 1994.
- KAABI, R. M., SOUVEYET, C., ROLLAND, C. Eliciting Service Composition in a Goal Driven Manner. In: *2nd INTERNATIONAL CONFERENCE ON SERVICE ORIENTED COMPUTING*, 2004. Proceedings..., USA: [s.n.], 2004, p. 308-315.
- KATZENSTEIN, G., LERCH, F. J. Beneath the surface of organizational processes: a social representation framework for business process redesign. *ACM Transactions On Information*, v. 18, n. 4, p. 383-422, October, 2000.
- KAVAKLI, V., LOUCOPOULOS, P. Goal-Driven Business Process Analysis - Application in Electricity Deregulation. *Information Systems*, v. 24, n. 3, p. 187-207, 1999.
- KOLIADIS, G., VRANESEVIC, A., BHUIYAN, M., KRISHNA, A., GHOSE, A. *Combining i* and BPMN for Business Process Model Lifecycle Management School of Information Technology and Computer Science (SITACS)*. University of Wollongong (UOW), Gwynneville, Australia, 2007.
- KOTONYA, G., SOMMERVILLE, I. *Requirements Engineering: Processes and Techniques*. 1 ed., John Wiley & Sons, 1998.
- LEITE, J. C. S. P., LEONARDI, M. C. Business rules as organizational policies. In: *INTERNATIONAL WORKSHOP ON SOFTWARE SPECIFICATION & DESIGN*, 1998. Proceedings..., Los Alamitos: IEEE CPS, 1998, p. 68-76.
- LOUCOPOULOS, P., KARAKOSTAS, V. *System Requirements Engineering*. 1 ed., McGraw-Hill, 1995.
- MACAULAY, L. *Requirements Engineering*, 1 ed., London: Springer, 1996.
- OUYANG, C., AALST, W. M. P., DUMAS, M., HOFSTEDDE, A. H. M. Translating BPMN to BPEL. *BPM Center Report*, 2006.
- PÁDUA, S. I. D., CAZARINI, E. W., INAMASU, R. Y. Modelagem Organizacional: Captura dos Requisitos Organizacionais no Desenvolvimento de Sistemas de Informação. *Gestão e Produção*, v. 11, n. 2, p. 197-209, mai-ago, 2004.

- PENADÉS, M. C., CANÓS, J. H., SÁNCHEZ, J. Automatic Derivation of Workflow Specifications from Organizational Structures and Use Cases. In: *ANAIS DO WER01 – WORKSHOP EM ENGENHARIA DE REQUISITOS*, 2001. Proceedings..., Buenos Aires: [s.n.], 2001, p. 166-180.
- ROLLAND, C. Conceptual Modelling in Information Systems Engineering, Capítulo: Capturing System Intentionality with Maps, p. 141-158, Springer, 2007.
- ROLLAND, C., NURCAN, S., GROSZ, G. A Decision making pattern for guiding the enterprise knowledge development process. *Journal of Information and Software Technology*, v. 42, p. 313-331, 2000.
- ROLLAND, C., PRAKASH, N., BENJAMEN, A. A Multi-Model View of Process Modelling. *Requirements Engineering*, v. 4, n. 4, p. 169-187, 1999.
- ROSCA, D., GREENSPAN, S., FEBLOWITZ, M., WILD, C. A Decision Making methodology in Support of the Business Rules Lifecycle. In: *IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING - RE97*, 1997. Proceedings..., Annapolis, MD, U.S.A.: [s.n.], 1997, p. 236-246.
- ROSCA, D., WILD, C. Business rules in the real world: a decision support approach. In: *EIGHTH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING – SEKE'96*, 1996. Proceedings..., Lake Tahoe, California, USA: [s.n.], 1996, p. 121-128.
- SANTANDER, V. F. A. *Integrando Modelagem Organizacional com Modelagem Funcional*. Tese (Tese de Doutorado) – Engenharia de Software – Universidade Federal de Pernambuco, Recife, PE, 2002.
- SANTANDER, V. F. A., CASTRO, J. F. Deriving Use Cases from Organizational Modeling. In: *IEEE JOINT INTERNATIONAL REQUIREMENTS ENGINEERING CONFERENCE – RE02*, 2002. Proceedings..., University of Essen, Germany: [s.n.], 2002, p. 32–39.
- SANTANDER, V. F. A., CASTRO, J. F. B. Desenvolvendo Use Cases a partir de Modelagem Organizacional. In: *WORKSHOP DE ENGENHARIA DE REQUISITOS*, 2000. Proceedings..., Rio de Janeiro: [s.n.], 2000, p. 158–179.
- SANTANDER, V. F. A., SILVA, D. R. Requirements Engineering Contributions on the Development of Educational Software for the Blind or People with Impaired Vision – An Experience Account. *CLEI Electronic Journal*, v. 9, n. 1, 2006.

- SANTOS, R. F. *Mapeamento e Modelagem de Processos de Negócios com BPMN*, <http://www.scribd.com/doc/18310726/Mapeamento-e-Modelagem-de-Processos-de-Negocio-com-BPMN>, Consultado na Internet em: 15/05/2010.
- SMITH, H., FINGAR, P. *Business Process Management – The Third Wave*. Florida: Meghan-Kiffer Press, 2003.
- SOLINGEN, R., BERGHOUT, E. *The Goal/Question/Metric Method: a Practical Guide for Quality Improvement of Software Development*. UK: McGraw-Hill, 1999.
- SOMMERVILLE, I. *Software engineering*. 6 ed., Addison-Wesley, 2001.
- SOMMERVILLE, I., SAWYER, P. *Requirements Engineering: A Good Practice Guide*. John Wiley & Sons, 1997.
- TEIXEIRA, E. P. *Integrando a Teoria da Atividade e a Técnica i* na fase de Requisitos Detalhados*. Monografia (Monografia de Curso) – Engenharia de Software – Universidade Estadual do Oeste do Paraná, Cascavel, PR, 2009. Monografia.
- TRAVASSOS, G. H., GUROV, D., AMARAL, E. A. G. *Introdução à Engenharia de Software Experimental*. Rio de Janeiro: COPPE/UFRJ, Programa de Engenharia de Sistemas e Computação, 2002. Relatório Técnico RT-ES-590/02.
- TROVA, E. C. V. *A importância da modelagem dos processos de negócios para o desenvolvimento de sistema de informação: uma aplicação em gestão e controle acadêmico*. Dissertação (Dissertação de Mestrado) – Engenharia de Software – Universidade Metodista de Piracicaba, Santa Bárbara D’Oeste, 2004. Dissertação.
- VARA, J. L., FORTUNA, M. H., SÁNCHEZ, J., WERNER, C. M. L., BORGES, M. R. S. A Requirements Engineering Approach for Data Modelling of Process-Aware Information Systems. In: *12TH INTERNATIONAL CONFERENCE ON BUSINESS INFORMATION SYSTEMS – BIS 2009*, 2009. Proceedings..., Poznan, Poland: [s.n.], 2009, p. 133-144.
- VICENTE, A. A., SANTANDER, V. F. A., CASTRO, J. F. B., SILVA, I. F., MATUS, F. G. R. JGOOSE: A Requirements Engineering Tool to Integrate i* Organizational Modeling with User cases in Uml. *Ingeniare. Revista chilena de ingeniería*, v. 17, n. 1, p. 6-20, 2009.
- WFMC. *The Workflow Management Coalition*. <http://www.WFMC.org>, Consultado na Internet em: 20/05/2010.

- WHITE, S. *Business Process Modeling Notation (BPMN) Version 1.0*. Business Process Management Initiative, May, 2004.
- WOHLIN, C., RUNESON, P., HOST, M., OHLSSON, M. C., REGNELL, B., WESSLÉN, A. *Experimentation in Software Engineering: an Introduction*. 1 ed., Springer, 2000.
- YU, E. *Modelling Strategic Relationships for Processes Reengineering*. Thesis (PhD Thesis) – Software Engineering – University of Toronto, Toronto, Canadá, 1995a.
- YU, E. Models for Supporting the Redesign of Organizational Work. In: *CONFERENCE ON ORGANIZATIONAL COMPUTING SYSTEMS - COOCS'95*, 1995. Proceedings..., Milpitas, California, USA: [s.n.], 1995b, p. 225-236.
- YU, E. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. In: *3rd IEEE INTERNATIONAL SYMPOSIUM ON REQUIREMENTS ENGINEERING - RE97*, 1997. Proceedings..., Annapolis, MD, USA: [s.n.], 1995, p. 226-235.
- YU, E., BOIS, P., MYLOPOULOS, J. From Organization Models to System Requirements - A Cooperating Agents Approach. In: *3rd INTERNATIONAL CONFERENCE ON COOPERATIVE INFORMATION SYSTEMS - COOPIS95*, 1995. Proceedings..., Vienna, Austria: [s.n.], 1995, p. 2-9.
- YU, E., GIORGINI, P., MAIDEN, N., MYLOPOULOS, J. *Social Modeling for Requirements Engineering*. (to appear), 1 ed., The MIT Press, 2011.