



Pós-Graduação em Ciência da Computação

Integração de Requisitos Não-Funcionais a Processos de Negócio: Integrando BPMN e RNF

Por

Laís Xavier

DISSERTAÇÃO DE MESTRADO



Universidade Federal de Pernambuco
Centro de Informática

Recife – PE
31 de Agosto de 2009

Universidade Federal de Pernambuco
Centro de Informática

Laís Xavier

Integração de Requisitos Não-Funcionais a Processos de Negócio: Integrando BPMN e RNF

*Trabalho apresentado ao Programa de Pós-graduação em
Ciência da Computação do Centro de Informática da Univer-
sidade Federal de Pernambuco como requisito parcial para
obtenção do grau de Mestre em Ciência da Computação.*

*A Master Thesis presented to the Federal University of Per-
nambuco in partial fulfillment of the requirements for the
degree of M.Sc. in Computer Science.*

Orientador: *Jaelson Freire Brelaz de Castro*
Co-orientadora: *Fernanda Maria Ribeiro de Alencar*

Recife – PE
31 de Agosto de 2009

*Este trabalho é dedicado aqueles que
realmente fazem diferença na minha vida:
Gerino, Angélica
Isa
e Mano.
Obrigada.*

Agradecimentos

Meus sinceros agradecimentos a todos, que direta ou indiretamente, contribuíram para a conclusão deste trabalho e de uma importante fase da minha vida. Aqui começa uma nova etapa com muito entusiasmo, perspectivas, esperanças, projetos e sonhos.

Agradeço à minha família - em especial minha mãe, meu pai e minha irmã - pelo amor, paciência e apoio a mim dedicados. À Mano, pelo amor, companheirismo, força, pelas aventuras, sonhos e planos. E aos meus amigos, em especial à Dudu e Clarissa, que sempre estiveram ao meu lado. Com vocês, compartilho mais essa conquista tão importante.

Aos que fazem parte do grupo de pesquisa LER do Centro de Informática, tenho um agradecimento especial pelos seminários, trocas de experiência, momentos de descontração, e acima de tudo pelo suporte acadêmico, técnico e emocional. Em particular, agradeço ao meu orientador o Professor Jaelson Castro pelo apoio e confiança e a minha Co-orientadora Fernanda Alencar pela atenção e força.

Aos meus amigos e companheiros pra toda vida (Jordão, Menezes, Ricardo e Teoria) pelo apoio em uma fase decisiva desta jornada. E aos colegas da Mídias educativas (em especial à Cynthia), pois vocês são mais que colegas de trabalho e me acompanharam sempre dispostos nesta jornada.

*“And as we let our own light shine,
We unconsciously give other people permission to do the same.
As we’re liberated from our own fear,
Our presence automatically liberates others.”*

—MARIANNE WILLIAMSON

Resumo

A Engenharia de requisitos tem sido amplamente reconhecida como um fator crítico de sucesso de projetos de Software. Se não forem devidamente elicitados, os requisitos podem levar o projeto ao fracasso. A elicitação equivocada dos requisitos pode estar relacionada com a falta de compreensão do negócio pelo analista de sistemas, a falta entendimento dos objetivos do sistema, bem como a falta de comunicação entre analistas de negócio e analistas de sistema. A existência de uma lacuna entre os domínios do negócio e da computação podem causar desequilíbrios entre o que os usuários finais precisam e o que os analistas de sistema desenvolvem. Uma das razões para o problema de comunicação é que os modelos de requisitos utilizados para interagir com os usuários finais podem ser difíceis de serem compreendidos e validados em função da alta complexidade das notações. O problema tratado nesta dissertação é a integração dos requisitos não funcionais aos modelos de processos de negócio, através de notações intuitivas para todos os usuários envolvidos no processo. Portanto, uma pesquisa baseada em análise de requisitos não funcionais e modelagem de processos de negócio é apresentada. Portanto, propomos uma abordagem para inserir os requisitos não funcionais na notação Business Process Modeling Notation - Business Process Modeling Notation (**BPMN**) . A abordagem utiliza catálogos para requisitos não funcionais, descritos na notação Non-Functional Requirement (**NFR**), que orientam a descoberta das suas operacionalizações. Esta abordagem é validada com um estudo de caso real.

Palavras-chave: Engenharia de Requisitos, Requisitos não-funcionais, **BPMN**, **NFR**-Framework.

Abstract

Requirements engineering has been widely recognized as a critical factor for success of software projects. If not properly addressed, the requirements may cause the project to fail. The inappropriate elicitation of requirements can be related to a lack of understanding of the business by the systems analyst, the lack of understanding of the purpose of the system, and lack of communication between business analysts and system analysts. The existence of a gap between the fields of business and the computer can cause mistakes between what the end users need and what system analysts provide. One reason for the problem of communication is that the requirements models used to interact with end users may be difficult to be understood and validated due the inhibit. The problem addressed in this dissertation is the integration of non-functional requirements with business process models, through a more intuitive notation for all users involved in the process. Therefore, are search based the on analysis of non-functional requirements and modeling business processes is presented. We propose an approach to embed the non-functional requirements in the **BPMN** notation. The approach catalogs of non-functional requirements, decribed in the **NFR** notation, to guide the discovery of their operacionalizations. This approach is validated with a real case study.

Keywords: Requirements Engineering, Non-functional Requirements, **BPMN**, **NFR**-Framework

Sumário

1	Introdução	3
1.1	Motivação	3
1.2	Objetivo	5
1.3	Contribuições	5
1.4	Metodologia	6
1.5	Estrutura do trabalho	7
2	Modelagem de processos de negócio	9
2.1	Modelos de Negócio	9
2.2	Modelos de processos de negócio	10
2.3	Notações	11
2.4	BPMN	12
2.4.1	História	13
2.4.2	Elementos	13
	Objetos de Fluxo	14
	Objetos de Conexão	16
	<i>Swimlane</i>	17
	Artefatos	19
2.5	Considerações finais	20
3	Requisitos Não-Funcionais	25
3.1	Conceito	25
3.2	<i>NFR-Framework</i>	26
3.2.1	Aquisição ou acesso a conhecimentos	27
3.2.2	Identificação dos requisitos não-funcionais particulares do domínio	27
3.2.3	Decomposição dos requisitos não-funcionais	28
3.2.4	Identificação das operacionalizações	28
3.2.5	Lidar com Ambigüidades, Prioridades e Interdependências entre requisitos não-funcionais e operacionalizações	28
3.2.6	Escolher as operacionalizações	30
3.2.7	Apoiar decisões com concepção lógica	32
3.2.8	Avaliar os impactos das decisões	32
3.3	Operacionalização de requisitos não-funcionais	32
3.4	Catálogo de Usabilidade	35

3.4.1	A Pesquisa	36
3.4.2	O catálogo	38
3.4.3	Considerações	40
3.5	Considerações Finais	40
4	Abordagem BPMNFR	45
4.1	A abordagem BPMNFR	45
4.1.1	Construção do Business Process Diagram (BPD)	46
	Caso de exemplo	49
4.1.2	Rótulos de requisitos não-funcionais no BPD	49
4.1.3	Detalhamento do requisito não-funcional	51
4.1.4	Inserção das operacionalizações no BPD	53
4.2	Análise comparativa com outras abordagens	56
4.2.1	i*	56
4.2.2	Santander e Castro	57
4.2.3	Pavlovski e Zou	57
4.2.4	Abordagem BPMNFR	58
4.3	Considerações Finais	58
5	Estudo de caso	61
5.1	Contextualização	61
5.2	Objetivos do estudo de caso	62
5.3	Estudo de caso	63
5.3.1	Modelagem do processo de desenvolvimento do produto	63
	Elicitação de requisitos	63
	Validação de requisitos	65
	Desenvolvimento dos componentes do sistema	65
	Testes	67
	Considerações	67
5.3.2	Inserção dos requisitos não-funcionais através dos rótulos	69
5.3.3	Operacionalizações dos requisitos não-funcionais	71
	Refinamento do Catálogo de Usabilidade	71
	Identificação das operacionalizações	74
	Inserção das operacionalizações no BPD	74
5.4	Considerações finais	82

6 Conclusão	83
6.1 Conclusões	83
6.2 Trabalhos futuros	85

Lista de Figuras

2.1	Figura de Eventos	15
2.2	Figura de Atividades	15
2.3	Figura de <i>Gateways</i>	16
2.4	Figura de Fluxo de Sequência	16
2.5	Figura de Fluxo de Mensagem	17
2.6	Figura de Associação	17
2.7	Figura de <i>Pool</i>	18
2.8	Figura de <i>Lane</i>	18
2.9	Figura de Objetos de Dados	19
2.10	Figura de Grupo	20
2.11	Figura de Anotação	20
2.12	Figura de Exemplo do uso da notação BPMN	21
2.13	Figura 2 de Exemplo do uso da notação BPMN	22
3.1	Figura do Softgoal Interdependency Graphs (SIG) inicial	28
3.2	Figura da decomposição por tipos	29
3.3	Figura de Identificação de um <i>softgoal</i> como prioridade	30
3.4	Figura de seleção de operacionalizações	31
3.5	Figura de <i>Claim-Softgoal</i>	33
3.6	Figura de Identificação de possíveis operacionalizações para requisito não-funcional <i>Softgoal</i>	34
3.7	Figura de Operacionalizações	35
3.8	Figura do catálogo de usabilidade - Eficiência e Eficácia	39
3.9	Figura do catálogo de usabilidade - Utilidade	41
3.10	Figura do catálogo de usabilidade - Segurança no uso	42
3.11	Figura do catálogo de usabilidade - Fácil de aprender e usar	43
4.1	Figura da Abordagem BPMNFR inicial para inserção de requisito não-funcional em BPMN.	47
4.2	Figura de Abordagem BPMNFR Final	48
4.3	Figura do Processo geral de desenvolvimento um objeto de aprendizagem	50
4.4	Figura do BPD com rótulos	52
4.5	Figura do BPD com operacionalizações	55
5.1	Figura do macro processo do estudo de caso	64

5.2	Figura subprocesso de elicitação de requisitos	66
5.3	Figura subprocesso de validação de requisitos	67
5.4	Figura subprocesso de implementação da interface	68
5.5	Figura subprocesso de testes	69
5.6	Figura do diagrama do lampejo rotulado	70
5.7	Figura do catálogo de usabilidade refinado - Fácil de aprender e usar . .	72
5.8	Figura do catálogo de usabilidade novo <i>softgoal</i> - Entender domínio do usuário	73
5.9	Figura das operacionalizações dos requisitos não-funcionais no BPD do lampejo	75
5.10	Figura das operacionalizações dos requisitos não-funcionais no BPD de elicitação de requisitos	76
5.11	Figura do detalhamento da operacionalização “prototipação”	78
5.12	Figura das operacionalizações dos requisitos não-funcionais no BPD de validação de requisitos	79
5.13	Figura das operacionalizações dos requisitos não-funcionais no BPD de desenvolvimento da interface	80
5.14	Figura da operacionalizações do desenho dos ícones	81
5.15	Figura da operacionalização dos testes	81

Lista de Tabelas

4.1	Tabela de correspondências de operacionalizações	56
5.1	Tabela de correspondências de operacionalizações	77

Lista de Abreviaturas

BPD Business Process Diagram

BPEL4WS Business Process Execution Language for Web Service

BPMI Business Process Management Initiative

BPML Business Process Modeling Language

BPMN Business Process Modeling Notation

BPMN-WG Business Process Modelling Notation Working Group

CETE Centro de Experimentação em Tecnologia Educacional

MEC Ministério da Educação

NFR Non-Functional Requirement

OMG Object Management Group

RNF Requisito Não-Funcional

SD Strategic Dependency

SEAD Secretaria de Ensino a Distância da UFPE

SEED Secretaria de Educação a Distância

SIG Softgoal Interdependency Graphs

SR Strategic Rationale

UFPE Universidade Federal de Pernambuco

UML Unified Modeling Language

XML eXtensible Markup Language

*Programmers Drinking Song sung to the tune of “100
Bottles of Beer”:
“99 little bugs in the code,
99 bugs in the code,
fix one bug, compile it again,
101 little bugs in the code.
101 little bugs in the code....
(Repeat until BUGS = 0)”*

Anonymous



Introdução

Neste capítulo serão descritos a motivação para realização do trabalho, a caracterização do problema encontrado, o objetivo desta dissertação, as contribuições principais e secundárias, bem como a estrutura deste trabalho.

1.1 Motivação

A Engenharia de Requisitos tem sido amplamente reconhecida como um fator crítico de sucesso de projetos de *Software* (Group, 1995). Se não forem devidamente elicitados, os requisitos podem levar o projeto ao fracasso. No entanto, a experiência prática demonstra que os problemas podem facilmente ser identificados nesta fase. Por exemplo, eles estão relacionados à falta de compreensão do negócio pelo analista de sistemas, mal entendimento da finalidade do sistema, e a falhas de comunicação entre analistas de negócio e analistas de sistemas. Uma vez que esses problemas podem dificultar o alinhamento entre negócios e tecnologia da informação (Luftman *et al.*, 1999; Reich and Benbasat, 2000), os sistemas de informação podem não satisfazer às necessidades da organização.

Da forma como os requisitos têm sido definidos nas soluções orientada ao negócio eles não refletem o ambiente de negócios, ou consistem apenas de um modelo de dados na forma de um diagrama de entidade-relacionamento ou modelo de classes. Como solução, a comunidade de engenharia de requisitos tem reconhecido a importância do uso de preocupações relativas ao negócio para direcionar a elicitação de requisitos (Sommerville and Sawyer, 1997). Trata-se, mais especificamente, da importância da modelagem organizacional durante a análise de requisitos (Kirikova and Bubenko, 1994) fazendo com que os analistas de sistemas desempenhem o papel de analistas de negócio (of Business Analysis, 2009). Modelos organizacionais descrevem a estrutura e o comportamento de uma

organização e são muito úteis para ajudar analistas de sistema a compreenderem corretamente o ambiente de negócios e os requisitos do sistema. Dentre outras abordagens, a Modelagem de Processos de Negócio foi declarada como uma boa abordagem para a modelagem organizacional e tem sido muito importante para o desenvolvimento de sistemas de informação (Dumas *et al.*, 2005).

Um outro fator, a boa comunicação entre analistas de negócio e analistas sistema, é essencial na fase requisitos (Holtzblatt and Beyer, 1995). No entanto, pode ser difícil de ser alcançada devido à existência de uma lacuna entre os domínios de negócio e de computação (Taylor-Cummings, 1998). Essa lacuna pode causar desequilíbrios entre o que os usuários finais verbalizam e o que os analistas de sistema compreendem. Uma das razões para a falta de comunicação é que os modelos de requisitos utilizados para interagir com os usuários finais podem ser difíceis de serem compreendidos e validados. Isso se deve à falta de conhecimento das notações utilizadas para descrever os requisitos em computação. Portanto, conclui-se que modelos que facilitem a comunicação durante a análise de requisitos devam ser utilizados.

Observando o contexto descrito acima, foi escolhida neste trabalho uma notação para modelagem de processos de negócio, a BPMN (*Business Process Modelling Notation*) (BPML.org, 2008). BPMN foi a notação escolhida por ser uma notação intuitiva, que usa construções familiares e um conjunto de regras comuns em processos de negócio (Dubray, 2004).

Contudo, apesar da utilização de modelos de processos de negócio facilitar a interação entre os analistas de negócios, analistas de sistemas e usuário finais, esse tipo de modelo ainda não consegue abranger tudo o que os usuários finais desejam de um sistema de informação. Segundo (Recker *et al.*, 2006) a linguagem de notação utilizada nesse trabalho, BPMN, tem algumas limitações relacionadas à construção dos modelos, sugerindo assim, que os usuários finais não estão aptos a descreverem todos os fenômenos encontrados em situações reais. O autor, inclusive, notou nas pesquisas que os usuários que tinham um maior conhecimento na área de computação pediam a inclusão de mais símbolos à notação BPMN para que se tornasse mais expressiva. Em (Pavlovski and Zou, 2008) é identificado que uma dessas deficiências da notação está na não existência da descrição dos requisitos não-funcionais (Requisito Não-Funcional - Requisito Não-Funcional (RNF)). Este foi o fato motivador deste trabalho de pesquisa uma vez que os requisitos não-funcionais são peças chaves dentro da modelagem e elicitação de requisitos. Segundo (Cysneiros, 2001) erros provocados por não se lidar convenientemente, ou simplesmente não lidar com requisitos não-funcionais são apontados entre os mais caros e difíceis de corrigir.

1.2 Objetivo

Este trabalho irá apresentar uma pesquisa baseada na análise de requisitos não-funcionais e na modelagem de processos de negócio, onde é proposta uma abordagem, chamada BPMNFR, para integrar os requisitos não-funcionais à notação **BPMN**, como forma de solucionar sua deficiência de expressividade desses requisitos. Assim, é proposta a criação de novos símbolos para que esta notação possa contemplar os requisitos não-funcionais.

O analista de sistemas, no papel de analista de negócios, usa a abordagem BPMNFR para deixar de forma clara a todos os envolvidos no processo, os requisitos não funcionais que deverão ser utilizados para alcançar a qualidade esperada para aplicação.

Esses requisitos não-funcionais devem ser elicitados levando-se em consideração o domínio da aplicação e os atributos de qualidade que são de maior importância para a mesma.

A abordagem BPMNFR mostra como explorar, com elementos da notação de negócio, a qualidade esperada para a aplicação. Utiliza catálogos para requisitos não-funcionais que orientam a descoberta das operacionalizações destes requisitos.

O resultado desta pesquisa é um processo que viabiliza a inserção das operacionalizações dos requisitos não-funcionais nos diagramas de processos de negócio. Trata da natureza subjetiva dos requisitos não-funcionais e contribui com mais uma evidência do uso da modelagem de processos de negócio como um passo inicial do processo de engenharia de requisitos.

1.3 Contribuições

Uma extensão de **BPMN** para a modelagem de requisitos não-funcionais é o principal resultado dessa dissertação. A viabilidade de aplicação da abordagem BPMNFR proposta foi verificada através do estudo de caso, que demonstrou a possibilidade de utilização da mesma.

O estudo e a apresentação dos conceitos e aspectos básicos relacionados aos requisitos não-funcionais e à notação para modelagem de processos de negócio foi uma contribuição secundária deste trabalho.

A apresentação da importância da modelagem dos processos de negócios na fase de engenharia de requisitos para o desenvolvimento de uma aplicação de *software* também foi uma contribuição importante deste trabalho. Além da criação e detalhamento de um catálogo do requisito não-funcional usabilidade.

1.4 Metodologia

A pesquisa em Engenharia de *Software* pode ser conduzida através de quatro métodos: o científico, o de engenharia, o experimental e o analítico (Wohlin *et al.*, 2000).

Sousa, em seu trabalho (Sousa, 2004), descreve os métodos da seguinte maneira: o método científico foca na observação de um ambiente e na tentativa de extrair dele algum modelo ou teoria que possa explicar o fenômeno estudado. O método de engenharia, por sua vez, é baseado na observação de soluções existentes, na identificação de problemas nessas soluções e na sugestão de abordagens para melhorar as soluções analisadas. Já o método experimental pretende fornecer um modelo novo para solução de um problema e tenta estudar o impacto desse modelo. Por fim, o método analítico oferece uma base analítica (ou matemática) para o desenvolvimento de modelos.

Este trabalho de pesquisa seguiu o método de engenharia. Inicialmente foi feita uma revisão bibliográfica dos principais trabalhos sobre modelagens de processos de negócio e a utilização de modelagens de processos de negócio e outras modelagens juntas. Os principais trabalhos estudados foram (de La Vara *et al.*, 2008) que mostrava Utilização de BPMN e MAPS (Rolland, 2007), uso do BPMN com Casos de Controle e Condições de Operações, do inglês, *Control Cases* e *Operation Conditions*, visto em (Pavlovski and Zou, 2008). Esses trabalhos serviram para identificar pontos que precisavam ser melhorados na modelagem de processos de negócio.

O passo seguinte foi fazer um estudo aprofundado sobre o BPMN, a notação de modelagem de processos de negócio escolhida. O ponto identificado para melhoria da modelagem foi a inserção dos requisitos não-funcionais na notação. Na modelagem de processo de negócio, quando era necessário modelar requisitos não-funcionais, não existia a modelagem explícita desses requisitos, isto é a modelagem não oferecia informações suficientes para deixar claros esses requisitos.

Propomos a utilização do NFR-Framework (Mylopoulos *et al.*, 1992; Chung *et al.*, 2000; Mylopoulos *et al.*, 2001) para decompor preocupações não-funcionais em operacionalizações que indicariam como de fato elas deveriam ser concretizadas na modelagem. Sendo assim, na nossa abordagem BPMNFR, a inserção é feita com as operacionalizações e não com atributos abstratos e subjetivos.

Portanto, nesta dissertação são definidas algumas adaptações no BPMN para que possa suportar a modelagem dos requisitos não-funcionais. Por fim, como forma de avaliar o trabalho, realizamos um estudo de caso no qual aplicamos a nossa proposta numa aplicação real na área de educação.

1.5 Estrutura do trabalho

Além do capítulo de introdução, este trabalho apresenta mais cinco capítulos. São eles:

- Capítulo 2 - Modelagem de Processos de Negócio: neste capítulo é apresentado um resumo sobre modelos de negócios e sua aplicação em desenvolvimento de sistemas; apresenta-se também a modelagem de processos de negócio e algumas abordagens existentes; além de uma introdução sobre a notação **BPMN**, algumas definições e os elementos que fazem parte desta notação. Este capítulo também mostra sua utilização e aceitação no mercado e na academia. Finalmente, é apresentada a evolução da notação e o que motivou sua criação.
- Capítulo 3 - Requisitos Não-Funcionais: neste capítulo, inicialmente, introduz-se a noção de requisitos não-funcionais, e algumas definições básicas. Posteriormente, é apresentado o *NFR-Framework* e sua utilização. Depois é mostrada de forma mais detalhada a obtenção de operacionalizações no *NFR-Framework*.
- Capítulo 4 - Abordagem BPMNFR: neste capítulo é mostrado e detalhado o processo proposto para integrar os requisitos não-funcionais à modelagem de processos de negócio, bem como os passos para utilização do mesmo. Esses passos são demonstrados com um caso de exemplo.
- Capítulo 5 - Estudo de caso: neste capítulo, apresenta-se a execução de um estudo de caso. É feita a modelagem de uma aplicação real para testar a efetividade do uso da abordagem BPMNFR criada.
- Capítulo 6 - Conclusão: por fim, neste capítulo são apresentadas as conclusões obtidas no desenvolvimento desta dissertação, bem como descritos alguns possíveis trabalhos futuros.

*“Snowflakes are one of nature’s most fragile things,
but just look what they can do when they stick together.”*

Vesta M. Kelly

2

Modelagem de processos de negócio

O objetivo deste capítulo é apresentar um resumo sobre modelos de negócio e sua aplicação em desenvolvimento de sistema; apresentar a modelagem de processos de negócio e mostrar brevemente algumas abordagens; além de mostrar de forma mais detalhada a notação escolhida para este trabalho.

2.1 Modelos de Negócio

Modelos de Negócio são compostos por um conjunto de modelos que representam uma abstração da realidade de uma organização sob o ponto de vista do negócio, e representam o ambiente no qual a organização está inserida (Davenport and Stoddard, 1994; Davenport, 1994; Eriksson and Penker, 2000; Berio and Vernadat, 2001; Paim, 2007; Bittencourt, 2009).

Os modelos de negócio podem ser utilizados na distribuição do conhecimento dentro de uma organização; no planejamento de melhorias e mudanças nos processos; na automação de processos; no desenvolvimento de sistemas (Knight, 2004; de La Vara *et al.*, 2008; Weiß *et al.*, 2008; of Business Analysis, 2009; Sharp and McDermott, 2009); na implantação de sistemas integrados de gestão; na identificação de indicadores de desempenho; na análise organizacional; na implantação de sistemas de workflow; e na gerência de documentos (Paim, 2007).

Em relação à tecnologia da informação, os modelos de negócio permitem compreender as interfaces organizacionais e o fluxo de informações através das unidades de negócio; contribuir para evitar sistemas redundantes; contribuir para integração de base dados; e permiti descrever o processo de desenvolvimento de um produto. Para este trabalho é importante a aplicação dos modelos de negócio no desenvolvimento de sistemas, pois permitem unificar o conhecimento dos envolvidos e entender as necessidades que o

sistema (Knight, 2004). Particularmente, os modelos de processos de negócio têm sido vistos como uma forma de integrar as linguagens dos profissionais que têm visão de negócios e dos profissionais com visão tecnológica (Paim, 2007; de La Vara *et al.*, 2008).

2.2 Modelos de processos de negócio

No início da década de 1990, corporações e empresas em todo o mundo, começaram a adotar um novo conceito, o de reengenharia de processos de negócio - como o nome sugere, tem foco em processos de negócio -, em uma tentativa de voltar a atingir a competitividade que se tinha perdido durante a década anterior. Davenport (1993) define um processo de negócio da seguinte maneira: “(...) *um processo é simplesmente um conjunto de atividades estruturadas e medidas destinadas a resultar num produto específico para um determinado cliente ou mercado(...)* uma ordenação específica das atividades de trabalho no tempo e no espaço, com um começo, um fim, e entradas e saídas claramente identificados(...)”.

Seguindo a definição de Davenport (1993), pode-se concluir que um processo deve ter fronteiras claramente definidas, ter começo e fim, ter atividades sequenciais, que estão ordenadas no tempo e no espaço, que deve haver um receptor do resultado do processo - um cliente - e que a transformação em curso no âmbito do processo deve agregar valor cliente.

Outra definição de processo foi encontrada em (Rummler and Brache, 1995) que abrange claramente um foco em clientes externos da organização, ao afirmar que

“Um processo de negócio é uma série de medidas concebidas para produzir um produto ou serviço. (...) A maioria dos processos são transversais, abrangendo o 'espaço branco' entre as caixas sobre o organograma. Alguns processos resultam em um produto ou serviço que é recebido por uma organização externa do cliente(...) Outros processos produzem produtos que são invisíveis para o cliente externo, mas essencial para a gestão eficaz do negócio(...)”.

A definição acima distingue dois tipos de processos, dependendo se o processo está diretamente envolvido na criação de valor dos clientes, ou preocupado com as atividades internas da organização. A característica de processos de abranger o “espaço em branco” no organograma indica que os processos estão integrados em algum tipo de estrutura organizacional e que permeiam esta estrutura.

Finalmente, vamos considerar a definição de processo de Johansson and Pendlebury (1993). Eles definem um processo como:

“Um conjunto de atividades conexas que tenham uma entrada e que será transformada para gerar uma saída. Idealmente, a transformação que ocorre no processo deve agregar valor à entrada e criar uma saída que é mais útil e eficaz para o destinatário (...).”

Esta definição enfatiza também a constituição de vínculos entre as atividades e a transformação que ocorre dentro do processo.

Esta dissertação explora o conceito de processo constituído por um conjunto de atividades iniciadas em resposta a um evento, ordenadas no tempo e no espaço, com entradas e saídas claramente identificados, gerando produtos, serviços ou informação. Um processo cumpre um objetivo, seja da organização ou da aplicação que será gerada, e deve ser executado levando-se em consideração o escopo em que está inserido e suas particularidades.

De acordo com os conceitos para modelos de processos acima citados, destacam-se os elementos utilizados no modelo de processo de negócio: atividades, atores, eventos, regras, recursos, artefatos de entradas e saídas. Os elementos de negócio são conceitualmente as informações utilizadas para a especificação de um sistema de informação: as atividades se traduzem em funcionalidades; artefatos que são transformados pelas atividades e se transformam em entidades de informação manipuladas pelo sistema; e atores que representam os elementos (cargos, organizações, sistemas equipamentos) que interagem com o sistema (Bittencourt, 2009).

2.3 Notações

Algumas notações são utilizadas para representar a modelagem dos processos de negócios tais como: **BPMN** - *Business Process Modelling Notation* (BPMI.org, 2008), Unified Modeling Language (**UML**) - *Unified Modeling Language* (Booch et al., 1998), i* (Yu, 1995). Essas notações permitem representar processos oferecendo meios para se entender o contexto da organização.

BPMN é uma notação desenvolvida pelo Business Process Management Initiative (**BPMI**) (*Business Process Management Initiative*) que define um **BPD** (*Business Process Diagram*), modelo gráfico para processos de negócio, baseada na técnica de fluxograma (Harrington, 1991). O objetivo principal é oferecer uma notação padrão e compreensível aos usuários do negócio; aos analistas de negócio responsáveis pela diagramação dos processos; aos desenvolvedores, responsáveis pela implementação da tecnologia que executará os processos; às pessoas do negócio que gerenciam e monitoram os processos. Uma pesquisa (Harmon and Wolf, 2008) realizada no mercado abrangendo EUA, Europa

e América do Sul aponta que a preferência, com 41% dos votos dentre as empresas entrevistadas, está na adoção do **BPMN** como padrão para modelar processos enquanto **UML** ficou com 30% dos votos.

A notação **i*** é centrado na modelagem das dependências entre os atores organizacionais a fim de alcançar objetivos organizacionais (Yu, 1995). Tem sido utilizada em vários métodos de desenvolvimento, tais como Tropos (Castro *et al.*, 2002) ou RESCUE (Jones and Maiden, 2004). No entanto, várias deficiências foram identificadas pelos profissionais (Estrada *et al.*, 2006). Os diagramas que utilizam o **i*** podem se tornar muito complexos e deverão ser utilizados quando for necessária uma maior granularidade e refinamento dos processos na modelagem.

Algumas abordagens usam **UML** para modelagem organizacional e processo de negócio (Eriksson and Penker, 2000). Estas abordagens usam elementos que estão próximos aos elementos utilizados na área do desenvolvimento do *software*. Este fato é uma desvantagem, porque apesar dos modelos serem de fácil compreensão e utilização pelos analistas de sistemas, podem ser demasiado complexos para serem validados pelos usuários finais.

Neste trabalho, a notação **BPMN** foi escolhida por ter duas grandes vantagens: a notação ter a preferência do mercado e ser de fácil uso e entendimento. Como resultado, **BPMN** é considerada o padrão, de fato, para a modelagem de processos de negócio.

2.4 **BPMN**

Segundo (White, 2006), o **BPMN**, *Business Process Modeling Notation*, trata-se de uma notação padrão para o desenho de fluxogramas em processos de negócios e *web services*. Na prática trata-se de um conjunto de regras e convenções que determinam como os fluxogramas devem ser desenhados. **BPMN** foi desenvolvido pelo *Business Process Management Initiative* (**BPMI**) e agora é mantido pela *Object Management Group* (Object Management Group (**OMG**)) desde que as duas organizações foram fundidas em 2005. O **BPMN** está na versão 1.1 e está sendo proposta a versão 2.0.

O principal objetivo do **BPMN**, como dito em (Owen and Raj, 2003), é prover uma notação que seja entendida por todos os usuários envolvidos no negócio. Isto inclui desde os analistas de negócio que criam o primeiro esboço dos processos até os desenvolvedores responsáveis pela implementação da tecnologia que irá realizar esses processos. Outro e igualmente importante objetivo é assegurar que linguagens baseadas em eXtensible Markup Language (**XML**) para execução de processos como a Business Process Execution

Language for Web Service (**BPEL4WS**) (*Business Process Execution Language for Web Service*) e Business Process Modeling Language (**BPML**) (*Business Process Modeling Language*) possam ser visualmente expressadas com uma notação comum.

A notação do **BPMN** é usada para produzir um diagrama, chamado *Business Process Diagram* (**BPD**). O **BPD** é concebido a partir de um conjunto de elementos gráficos que compõem diagramas simples para desenvolver e de serem compreendidos. A especificação completa **BPMN** define mais de 50 atributos, agrupados em quatro categorias básicas de elementos. Um modelo de processo de negócios, portanto, é uma rede de objetos gráficos, com atividades (i.e. trabalho) e controladores de fluxo que definem a ordem que eles são executados.

Segundo (Dubray, 2004), o **BPMN** vem tentando achar a melhor maneira para a criação de uma notação intuitiva, usando construções familiares e um conjunto de regras comuns em processos de negócio. Além disso, **BPMN** define construções mais avançadas, como o *looping* de tarefas, múltiplas instâncias e *gateways* de decisão do tipo OR (ou seja, *OR-splits*), que pode ser substituído por uma combinação dos elementos principais de **BPMN**.

Nesta seção serão apresentados os principais conceitos e construtores relacionados à notação. Para mais informações sobre o **BPMN** consultar ([BPMI.org](http://bpmi.org), 2008).

2.4.1 História

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

Em agosto de 2001, o *Business Process Modeling Notation Working Group* (Business Process Modelling Notation Working Group (**BPMN-WG**)), da **BPMI.org**, foi formado por 35 empresas e iniciou os trabalhos para criar a **BPMN**. A versão 1.0 da especificação escrita por Stephen White da IBM saiu em maio de 2004. Em junho de 2005, a **BPMI** anunciou sua junção a **OMG** (*Object Management Group*). Segundo a **OMG**, até abril de 2009, existem cinquenta e três fornecedores que suportam a notação e mais quatro estão em fase de implementação. A última especificação da **BPMN** (versão 1.1) é de fevereiro de 2008, mas uma nova versão (versão 2.0) provavelmente surgirá até o final de 2009.

O **BPMN** vem recebendo não só atenção do mercado, mas também da academia, em (Wahl and Sindre, 2005) observa-se que praticamente todas as contribuições foram

feitas em um nível analítico e conceitual. Ainda assim existem também alguns estudos empíricos de como **BPMN** é utilizado na prática (Muehlen and Recker, 2008; de La Vara *et al.*, 2008; Recker *et al.*, 2006). Alguns grupos de pesquisa que estudam **BPMN** têm a preocupação de colher dados referentes a real utilização do **BPMN** no mercado, tal como dito em (Recker, 2008; Recker *et al.*, 2006; Muehlen and Ho, 2008).

2.4.2 Elementos

Um **BPD** é composto de um conjunto de elementos gráficos. Esses elementos permitem uma criação de um diagrama simples que é familiar para a maioria dos analistas de negócio. Os elementos foram escolhidos para ser distinguíveis entre eles e para utilizar formatos que fossem familiares para a maior parte dos usuários. Por exemplo, atividades são retangulares e as decisões são losangos. Como ressaltou (White, 2006) uma das premissas para o desenvolvimento do **BPMN** é criar um mecanismo simples para fazer modelos de processos de negócio, ao mesmo tempo ser capaz de lidar com a complexidade inerente aos processos de negócio.

A abordagem adotada para lidar com esses dois requisitos conflitantes se baseia na organização de aspectos gráficos da notação em categorias específicas. Isto fornece um pequeno conjunto de categorias de notação de modo a que o leitor de um **BPD** possa reconhecer facilmente os tipos básicos de elementos e compreender o diagrama. Dentro das categorias básicas de elementos, variações e informações adicionais podem ser inseridas para suportar os requisitos de complexidade sem, contudo, mudar muito o objetivo de olhar e entender o diagrama. As quatro categorias básicas de elementos são:

- Objetos de Fluxo (do inglês, *Flow Objects*)
- Objetos de conexão (do inglês, *Connecting Objects*)
- *Swimlanes*
- Artefatos

Também é permitido no **BPD** que o usuário faça o seu próprio Objeto de Fluxo ou um artefato para que seja mais fácil de compreender.

Objetos de Fluxo

Um **BPD** tem um pequeno conjunto de (três) elementos fundamentais, que são os Objetos de Fluxo, de forma que os analistas não têm de aprender e reconhecer um grande número de formas diferentes. Os três Objetos de Fluxo são:

- Eventos (Figura 2.1) - É representado por um círculo e é algo que “acontece” no decurso de um processo de negócio. Eventos afetam o fluxo do processo e, geralmente, têm uma causa (gatilho) ou um impacto (resultado). Eventos são círculos com centros abertos para permitir que sejam diferenciados. Existem três tipos de eventos, com base em quando eles afetam o fluxo: Início, Intermediário e Final.



Figura 2.1 Eventos.

- Atividades (Figura 2.2) - Termo genérico para o trabalho a ser realizado. Uma atividade é representada por um retângulo arredondado. Uma atividade pode ser atômica ou não atômica (composta). Atividades atômicas podem ser do tipo: serviço, enviar, receber, tarefa de usuário (*workflow*), tarefa de script ou tarefa manual. Uma tarefa de usuário deve ter um responsável.

BPMN permite especificar uma regra de negócio do tipo “para cada” (tradução livre - do inglês, *for-each*) em cada atividade. **BPMN** suporta também atividades em *looping* que é outra construção frequentemente utilizada em processos de negócio, mas, como dito em (Dubray, 2004), ausentes em linguagens de execução.

Os requisitos funcionais de uma aplicação podem ser mapeados como atividades dentro de um **BPD**.

Os tipos de atividades são: Atividades e Sub-processos. O Sub-processo é distinguido por um pequeno sinal de mais na parte inferior central do desenho.

- *Gateways* (Figura 2.3) - Representado por um losango, é usado para controlar a divergência e convergência de seqüências. Assim, irá determinar decisões tradicionais, assim como *forking*, fusão e junção dos caminhos. Marcadores internos irão indicar o tipo de comportamento do controle. Como exemplificado em (Owen and Raj, 2003) um *gateway* pode ser pensado como uma questão num determinado



Figura 2.2 Atividades.

ponto do fluxo do processo. Esta questão tem um conjunto definido de respostas, que são na realidade opções.



Figura 2.3 Gateways.

Objetos de Conexão

Os Objetos de Fluxo são conectados em um diagrama para criar a estrutura básica de um processo de negócio através dos Objetos de Conexão. Existem três diferentes Objetos de Conexão, estes conectores são:

- Fluxo de Sequência (Figura 2.4) - Representado por uma linha sólida com uma seta sólida e é usado para mostrar a ordem (a sequência) que as atividades serão realizadas em um processo.
- Fluxo de Mensagem (Figura 2.5) - Representado por uma linha tracejada com uma seta aberta e é usada para mostrar o fluxo de mensagens entre dois processos distintos (entidades de negócio ou papéis de negócio) que envia e recebe essas mensagens.

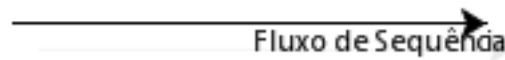


Figura 2.4 Fluxo de Sequência.



Figura 2.5 Fluxo de Mensagem.

- Associação (Figura 2.6) - Representada por uma linha pontilhada com uma seta e é usada para associar dados, texto e outros artefatos com fluxo de objetos. Associações são usadas para mostrar as entradas e saídas das atividades.

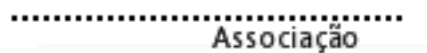


Figura 2.6 Associação.

Swimlane

Muitas metodologias de modelagem de processos utilizam o conceito de *swimlanes* como um mecanismo para organizar atividades em categorias visuais distintas, a fim de ilustrar diferentes responsabilidades ou capacidades funcionais (White, 2006). O BPMN suporta *swimlanes* com duas principais construções, elas são:

- *Pool* (Figura 2.7) - Representa um participante em um processo. Como exemplificou (Owen and Raj, 2003), um *pool* pode representar outras coisas além de uma organização, tais como uma função (algo que a organização atua, como marketing ou vendas), uma aplicação (ou *software* de computador), um local (um local físico na empresa), uma classe (um módulo de *software* orientado a objeto), ou uma entidade (que representa um quadro lógico em um banco de dados).



Figura 2.7 *Pool*.

- *Lane* (Figura 2.8) - É uma sub-partição dentro de um *Pool* e se estenderá por todo o comprimento do *Pool*, tanto horizontal como verticalmente. *Lanes* são usadas para organizar e categorizar as atividades. Permite agrupar atividades que são logicamente relacionadas entre si (por exemplo, quando são realizados pelo mesmo departamento). As *Lanes* organizam Objetos de Fluxo, Objeto de Conexão e Artefatos mais precisamente.

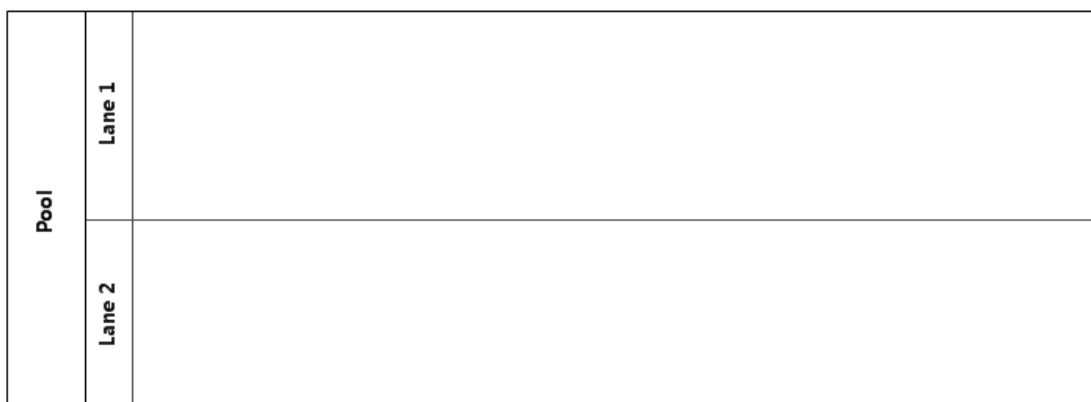


Figura 2.8 *Lane*.

Tipicamente, um *Pool* representa uma organização, e uma *Lane* representa um departamento dentro dessa organização. Ao tomar os processos e colocar em *Pools* ou *Lanes*, especificam-se quem faz o quê, para os eventos deve-se especificar onde ocorrem, e *gateways* para especificar onde as decisões são tomadas, ou quem as toma.

Como mostrado em (Owen and Raj, 2003) a analogia entre essa representação e piscinas é pertinente. Imagine um processo natação nadando numa raia, e alterando raias quando necessário para realizar uma atividade, dentro de uma piscina. Um *Pool* pode ser

considerado uma “piscina” de recursos. Há ocasiões em que o processo tem de saltar para outro grupo, porque ele tem diversos recursos necessários para completar a atividade.

Artefatos

BPMN foi projetada para permitir modelos e ferramentas de modelagem com flexibilidade para estender a notação básica. Qualquer número de Artefatos pode ser adicionado a um esquema conforme apropriado para o contexto dos processos de negócio a ser modelado.

Artefatos permitem aos usuários (tais como analistas de negócios e desenvolvedores) colocarem mais algumas informações no modelo/esquema. Desta forma, o modelo/esquema torna-se mais legível. Há três pré-definidos e Artefatos, são os seguintes:

- Objetos de Dados (Figura 2.9) - Artefatos são principalmente objetos de dados. Os dados são digitados e objetos representam a entrada e saída de atividades. Objetos de Dados é um mecanismo para mostrar como os dados são exigidos ou produzidos por atividades. Eles estão ligados a atividades através de associações. Objetos de Dados são artefatos que podem representar muitos tipos diferentes de itens físicos ou eletrônicos.



Figura 2.9 Objetos de Dados.

- Grupo (Figura 2.10) - Representado por um retângulo arredondado com uma linha tracejada. O agrupamento pode ser utilizado para a documentação ou análises finais do diagrama pelos analistas, pois eles não alteram o **BPD**.
- Anotação (Figura 2.11) - Utilizada para dar ao leitor do modelo/diagrama um entendimento maior. **BPMN** tem uma anotação textual que pode ser colocada em qualquer modelo, descrevendo detalhes sobre o elemento em linguagem natural.

A figura 2.12 exemplifica o uso de alguns elementos apresentados juntos. Nela é possível observar eventos de início e fim, atividades e artefatos que são representados

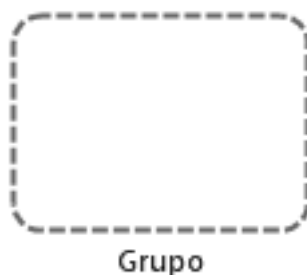


Figura 2.10 Grupo.

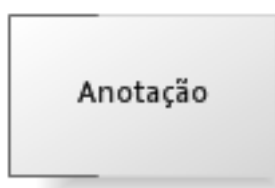


Figura 2.11 Anotação.

por objetos de dados. Um exemplo de atividade mostrado na figura é “Construção de um **BPD**” e de um objeto de dado, “**BPD**”.

Já a figura 2.13 exemplifica o uso de atividades, sub-processos, *gateway* e eventos juntos. Uma das atividades da figura 2.13 é “Integração” e um dos sub-processos é “Testes”.

2.5 Considerações finais

O levantamento bibliográfico sobre Modelos de Negócio e modelagem de processos de negócio mostra a importância do conhecimento da organização e sua estrutura para o desenvolvimento de algum sistema de tecnologia da informação. Algumas notações foram consideradas, i*, **UML** e **BPMN**. Dentre elas a **BPMN** é a notação mais bem aceita no mercado e por isso escolhida para o desenvolvimento deste trabalho.

A notação **BPMN** tem como principal vantagem a fácil compreensão de seus modelos por todos os atores envolvidos no processo de desenvolvimento de um sistema. Os diagramas gerados podem ser entendidos desde o analista de sistemas e o pessoal do desenvolvimento até o analista de negócios e usuários finais.

Para modelar os requisitos funcionais do sistema, a notação provê um elemento que pode ser diretamente associado a esse tipo de requisito, a Atividade. Portanto,

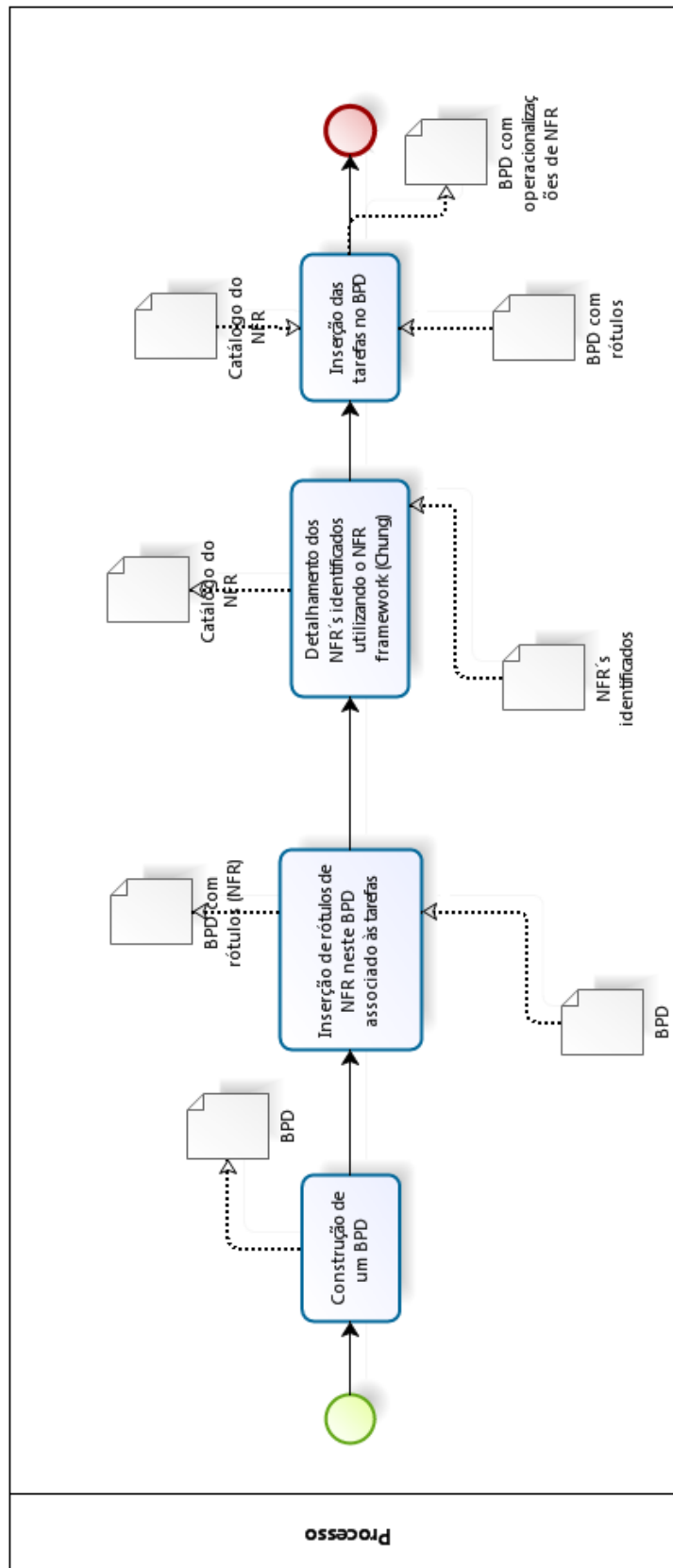
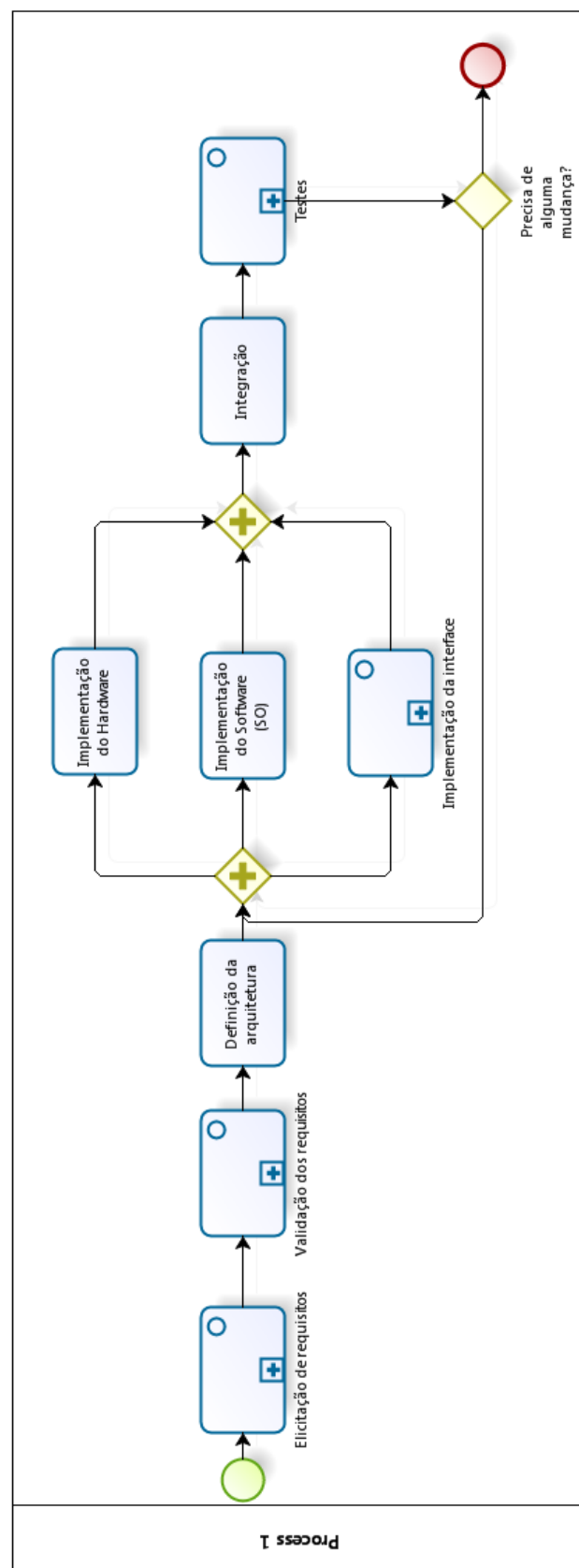


Figura 2.12 Exemplo do uso da notação **BPMN**.



os requisitos funcionais ficam explícitos durante a fase de requisitos. Porém, não são demonstrados de forma explícita os requisitos de qualidade (não-funcionais) através do diagrama gerado por esta notação.

Apesar dos requisitos não-funcionais não serem explicitamente tratados pela **BPMN**, a abordagem direta seria associar anotações textuais como artefatos para atividades em que tais requisitos se aplicam. Isto pode ser considerada uma abordagem desestruturada para iniciar a captura desses requisitos. Além disso, anotações textuais não são naturalmente interpretadas por máquinas.

Por isso, será proposta no capítulo 4 uma abordagem, chamada BPMNFR, para que os requisitos não-funcionais sejam mostrados explicitamente no diagrama gerado.

“Vision without action is a daydream.
Action without vision is a nightmare.”

Japanese Proverb

3

Requisitos Não-Funcionais

Neste capítulo serão apresentados alguns conceitos de requisitos não-funcionais. O *NFR-Framework* será mostrado, assim como o passo a passo para sua utilização. Posteriormente será apresentado um catálogo de usabilidade elaborado neste trabalho. Por fim, as considerações finais serão feitas.

3.1 Conceito

Os requisitos não-funcionais (RNFs) de *software* em geral se relacionam com padrões de qualidade como confiabilidade, performance, robustez, etc. Estes requisitos são importantes, pois definem se o *software* será eficiente para a tarefa que se propõe a fazer ou não. Segundo (Kirner and Davis, 1996), os requisitos não-funcionais representam requisitos adicionais que definem as qualidades globais ou atributos a serem exibidos pelo sistema resultante.

Segundo (Cysneiros and do Prado Leite, 1998), os requisitos não-funcionais, ao contrário dos funcionais, não expressam nenhuma função a serem realizados pelo *software*, e sim comportamentos e restrições que este *software* deve satisfazer.

Acessibilidade, segurança, confidencialidade, performance, portabilidade, consistência, manutenibilidade, eficiência, robustez, são alguns exemplos de requisitos não-funcionais. (Chung et al., 2000) também considera a usabilidade como um requisito não-funcional, apesar de não ser catalogada no seu *framework*.

Será dada uma ênfase em usabilidade numa seção deste capítulo, pois este é o requisito não-funcional que será utilizado para instanciar o processo desta dissertação.

Não há como desconsiderar a importância dos requisitos para o processo de desenvolvimento de *software*. Além disso, Engenheiros de Requisitos e a maioria das linguagens existentes incidem sobre os requisitos funcionais, negligenciado ou esque-

cendo os requisitos não-funcionais (RNFs) durante a concepção do *software*. Os atributos de qualidade de *software* (requisitos não-funcionais), por exemplo, são vistos como consequência destas decisões e não como algo que foi pensado. Para mudar este quadro e tratá-los coerentemente, é necessário não apenas métodos e representações, mas também como promover a sua reutilização.

Os requisitos não-funcionais desempenham um papel crítico durante o desenvolvimento de sistemas. Erros devido à falta de elicitação ou a elicitação incorreta destes requisitos estão entre os mais caros e difíceis de corrigir, uma vez que um sistema tenha sido implementado (Brooks, 1987; Davis, 1993). De acordo com (Cysneiros and do Prado Leite, 1998), para satisfazer um requisito não-funcional é possível que sejam criados conflitos, tanto com outros requisitos não-funcionais, como com requisitos funcionais. Consequentemente, o não tratamento dos requisitos não-funcionais durante o desenvolvimento do *software* pode ocasionar que esses conflitos só apareçam na implementação do *software*.

Segundo (Chung *et al.*, 2000), não existe uma definição formal ou lista completa de requisitos não-funcionais. Também não existe algum esquema de classificação único e universal que acomode todas as necessidades de aplicações de diferentes domínios. Por conta disso para cada aplicação e domínio diferentes classificações são feitas. Vários trabalhos foram desenvolvidos com classificações diferentes como visto em (Boehm, 1988; Sommerville and Sawyer, 1997; Davis, 1993).

Sendo assim, pesquisas indicam que técnicas baseadas em metas são mais adequadas para requisitos não-funcionais. Pois possibilitam (Chung *et al.*, 2000; Kavakli and Loucopoulos, 2003):

- Examinar a utilidade dos requisitos;
- Uma visão vertical desde a estratégia de alto nível até o detalhe;
- Uso de operadores lógicos E e OU que ajudam a uma melhor tomada de decisão;
- As formalizações, que permitem provar a correção e completude dos requisitos não-funcionais.

O *NFR-Framework* proposto por Chung e Mylopoulos é uma abordagem muito utilizada, pois se ajustam a todos os tipos de requisitos não-funcionais, desde as primeiras etapas do processo de desenvolvimento de *software* (Chung *et al.*, 2000; Chung and Nixon, 1995; Mylopoulos *et al.*, 1992). Por este motivo será utilizado nesta dissertação.

3.2 *NFR-Framework*

Muitos trabalhos sobre requisitos não-funcionais foram desenvolvidos tais como (Mylopoulos *et al.*, 1992; Chung and Nixon, 1995; Chung *et al.*, 2000) resultando em um *framework*, onde os requisitos não-funcionais são tratados como metas possivelmente conflitantes, devendo ser identificados em sua forma mais geral e refinados até que se chegue a um conjunto de requisitos que satisfaçam ao requisito geral.

A abordagem *NFR-Framework* (Chung *et al.*, 2000) orientada a metas foi concebida para ser intuitiva e suportar a lógica dos desenvolvedores. Foi baseada nas árvores E/OU usadas na resolução de problemas. Entretanto, essas metas, que representam requisitos não-funcionais podem raramente ser consideradas como totalmente “satisfeitas”. Diferentes decisões de projeto podem contribuir positiva ou negativamente em direção a uma meta particular. Essas metas podem ser avaliadas. Essa avaliação determina se um conjunto de requisitos não-funcionais está sendo alcançado por um projeto específico.

Este *framework* oferece uma estrutura para representar e registrar a concepção e raciocínio dos processos em grafos, chamados *Softgoal Interdependency Graph* (SIG). Além disso, ele também oferece um catálogo de conhecimento, incluindo técnicas de desenvolvimento.

O *NFR-Framework* pode ser usado para definir os requisitos não-funcionais tais como segurança, exatidão, performance e custo. De acordo com (Chung *et al.*, 2000), existem vários passos importantes neste processo de concepção que serão visto nas seções seguintes.

3.2.1 Aquisição ou acesso a conhecimentos

Neste ponto é importante que o desenvolvedor, ou analista adquira conhecimento sobre:

- O domínio em questão e o sistema que será desenvolvido;
- Requisitos funcionais do sistema em questão;
- Tipos específicos de requisitos não funcionais e associado com técnicas de desenvolvimento;

Estes conhecimentos devem ser adquiridos através de documentos (de natureza variada) da organização, informação sobre a própria organização e suas operações.

3.2.2 Identificação dos requisitos não-funcionais particulares do domínio

O desenvolvedor ou analista de sistemas estima inicialmente quais as prioridades para a organização e o sistema. Um **SIG** inicial é construído apenas com os principais requisitos não funcionais estimados. Esses requisitos são os *softgoals*. *Softgoals* são todos os objetivos não-funcionais que são difíceis de ser avaliados (Goldsby *et al.*, 2007). Esses *softgoals* posteriormente serão compreendidos, priorizados e tratados. Na figura 3.1 existe um **SIG** inicial e dois *NFR-softgoals*, representados através de nuvens. Esta figura adaptada do livro de (Chung *et al.*, 2000). Nela se vê os requisitos não-funcionais *good performance for accounts* e *secure accounts*, onde *good performance* e *secure* são os requisitos não-funcionais e *accounts* é o domínio da aplicação (tema, tópico).

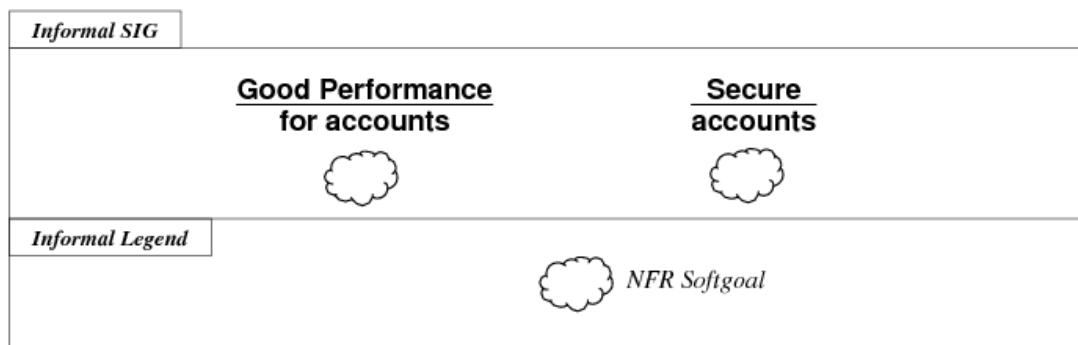


Figura 3.1 **SIG** Inicial (Chung *et al.*, 2000).

3.2.3 Decomposição dos requisitos não-funcionais

Segundo (Chung *et al.*, 2000) a decomposição dos requisitos não-funcionais se divide em tipo e tópico. Na decomposição por tipo são usados os métodos de decomposição existentes neste *framework*, que indicam quais são as decomposições possíveis para um determinado requisito não-funcional. A decomposição por tópicos trata da decomposição estrutural do problema. A Figura 3.2 mostra um exemplo de decomposição por tipo apresentada em (Chung *et al.*, 2000).

3.2.4 Identificação das operacionalizações

Em seguida se deve conceber possíveis alternativas para satisfazer os requisitos não-funcionais no sistema em questão. Este tópico será detalhado numa seção posterior.

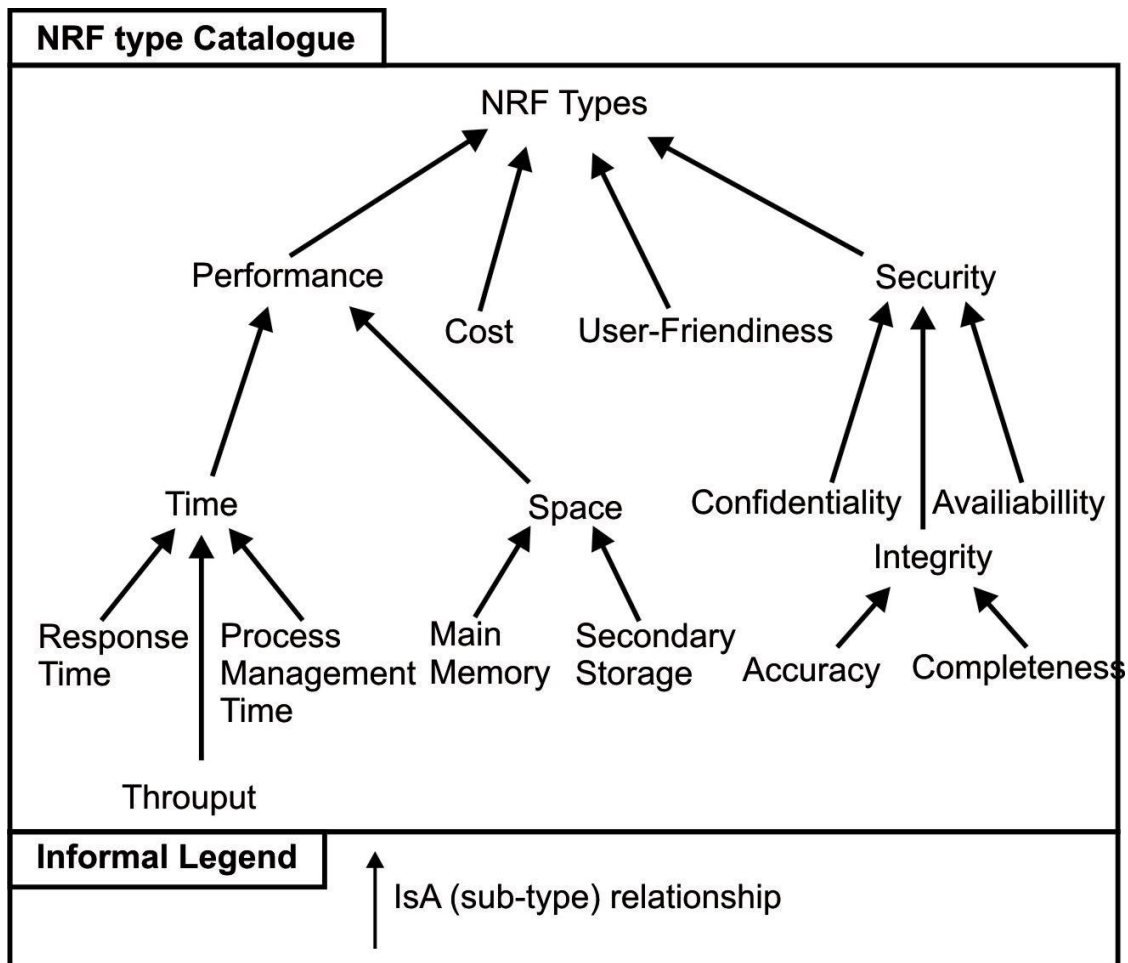


Figura 3.2 Decomposição por tipos (Chung *et al.*, 2000).

3.2.5 Lidar com Ambigüidades, Prioridades e Interdependências entre requisitos não-funcionais e operacionalizações

O **SIG** pode crescer e ficar muito grande e complexo. Por isso é necessário lidar com ambigüidades, prioridades e interdependência de operacionalizações. As prioridades organizacionais ou carga de trabalho que fazem parte do domínio da aplicação devem ser analisados. Além disso, requisitos podem ser identificados como prioridades durante várias fases do desenvolvimento. Os *softgoals* prioritários são identificados como críticos e vitais para o sucesso do *software* ou organização. Esses requisitos prioritários são identificados por um sinal de exclamação, como na figura 3.3.

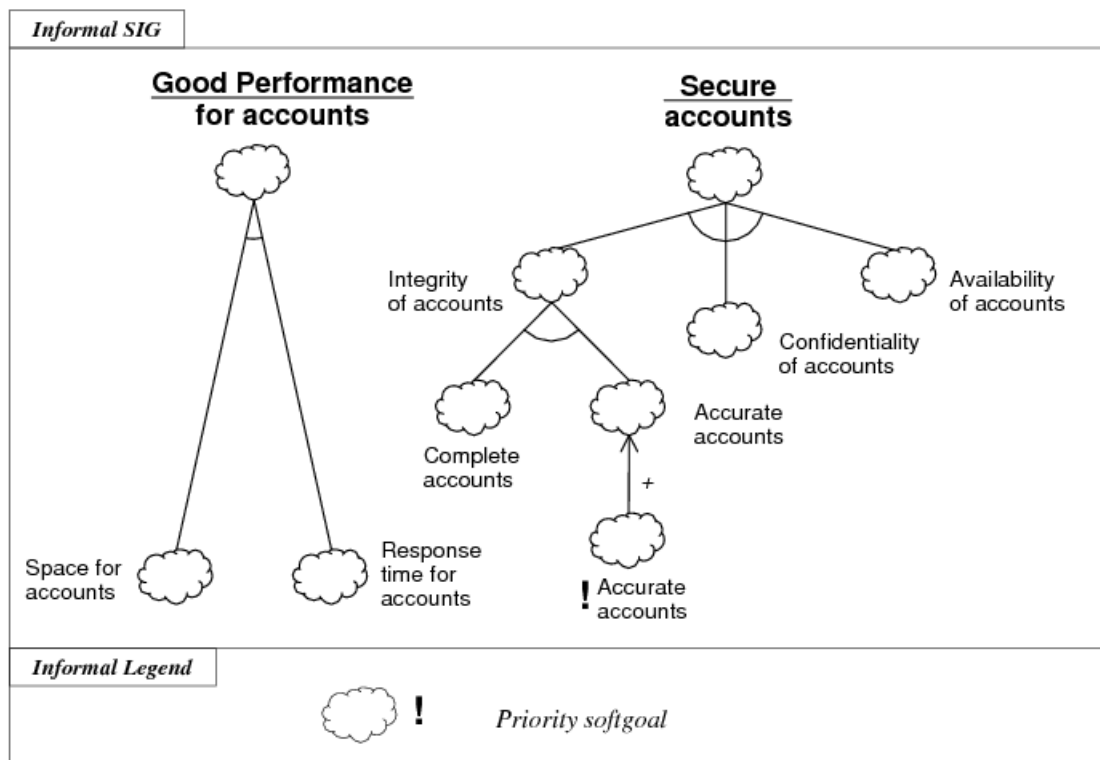


Figura 3.3 Identificação de um *softgoal* como prioridade (Chung *et al.*, 2000).

Contudo também é necessário observar de que forma as operacionalizações de um requisito não-funcional pode influenciar positivamente ou negativamente em outro. Essas interações são importantes, pois têm um impacto nas decisões de projeto. As interdependências implícitas só são identificadas à medida que o gráfico é construído, as explícitas aparecem de forma direta através da decomposição. Na figura 3.4 as interdependências explícitas são as setas sólidas e as implícitas são as tracejadas.

3.2.6 Escolher as operacionalizações

O processo de refinamento continua até o desenvolvedor ou analista de sistemas considerar que as soluções possíveis para aquele domínio específico foi suficientemente detalhada e não há mais nenhuma alternativa a ser considerada.

Como os *softgoals* foram refinados e o desenvolvedor ou analista já os detalharam suficientemente, agora chega o momento de aceitar ou rejeitar essas operacionalizações para fazerem parte da aplicação.

Existem opções de operacionalizações de *softgoals* no grafo, ver figura 3.4. Algumas serão aceitas (indicadas por um “v”) e outras rejeitadas (indicadas por um “x”).

Como mostrado na figura 3.4 foram identificadas três possíveis formas de autenticar o acesso dos usuários: “*Use Pin*”, “*Compare Signature*” e “*Require additional ID*”. A operacionalização aceita foi “*Compare Signature*” e por outro lado “*Require additional ID*” foi rejeitada, quanto nenhum ação foi tomada em relação a “*Use Pin*”.

3.2.7 Apoiar decisões com concepção lógica

Uma premissa desse *framework* é que as decisões de concepção devem ser suportadas com argumentos bem justificados, ou, como (Chung *et al.*, 2000) chama, deve registrar o *design rationale*. As razões podem ser indicadas para que haja refinamentos, para selecionar alguma alternativa de operacionalização para a aplicação específica.

O *rationale* é representado através do *claim-softgoal*. Quando uma declaração é utilizada para argumentar a favor ou contra alguma coisa, que constitui um argumento, isto é um *claim-softgoal*. O *claim-softgoal* é representado por uma nuvem tracejada, como visto na figura 3.5.

As argumentações devem ser expressas para fazer o refinamento, selecionar alternativas de operacionalizações, definirem as priorizações em qualquer fase do processo.

3.2.8 Avaliar os impactos das decisões

Para determinar o impacto das decisões, tanto as decisões atuais quanto as anteriores são consideradas. O processo de avaliação pode ser visto como *bottom-up*. Portanto pode ser considerado como trabalho de baixo para cima, começando com as decisões, que muitas vezes são as folhas do grafo, e, muitas vezes, estão na sua parte inferior. Em seguida, o processo de avaliação trabalha para a parte superior do gráfico, determinando o impacto sobre o nível mais alto, os principais *softgoals*. Estes *softgoals* globais refletem os

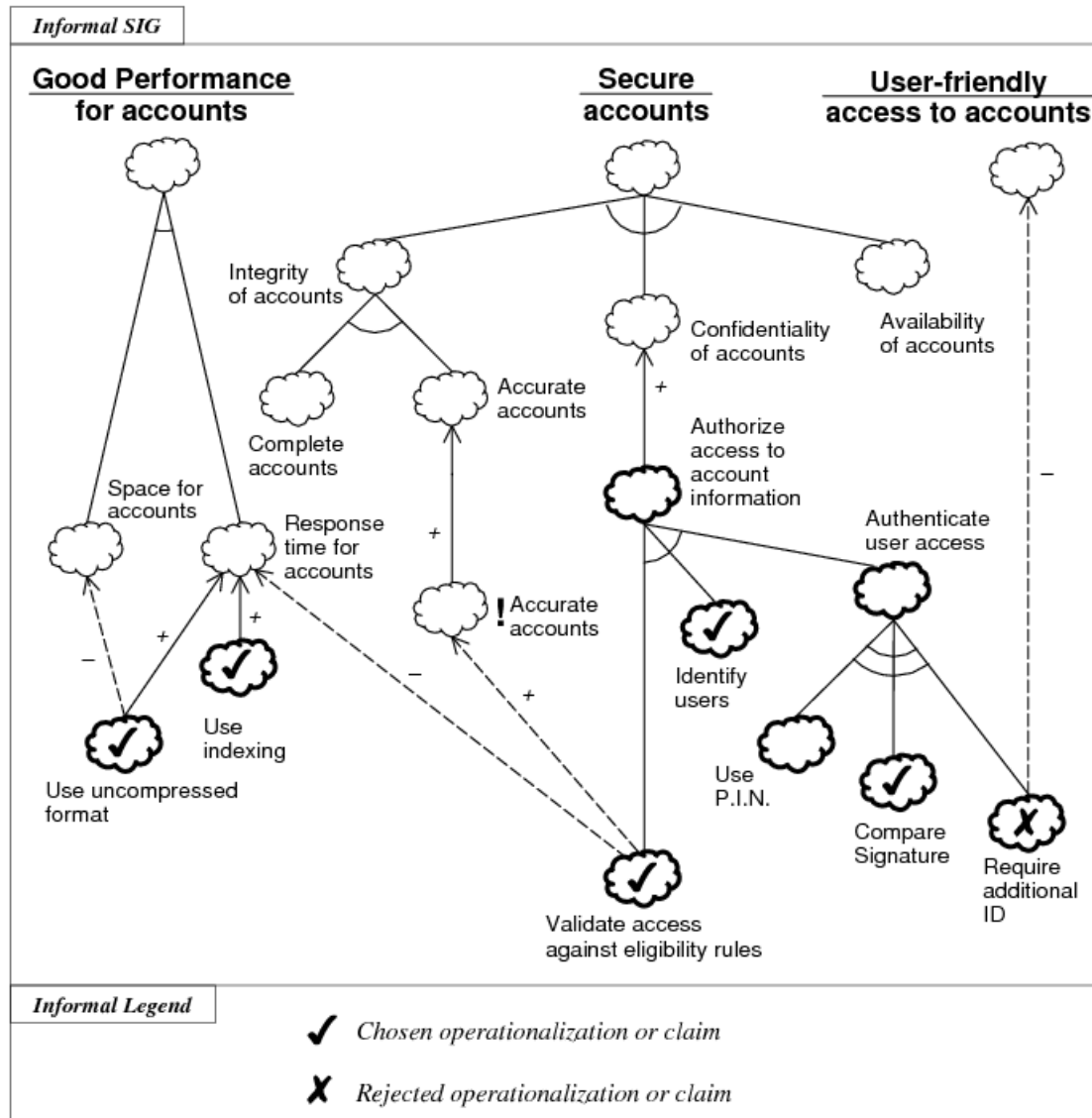


Figura 3.4 Seleção de operacionalizações (Chung *et al.*, 2000).

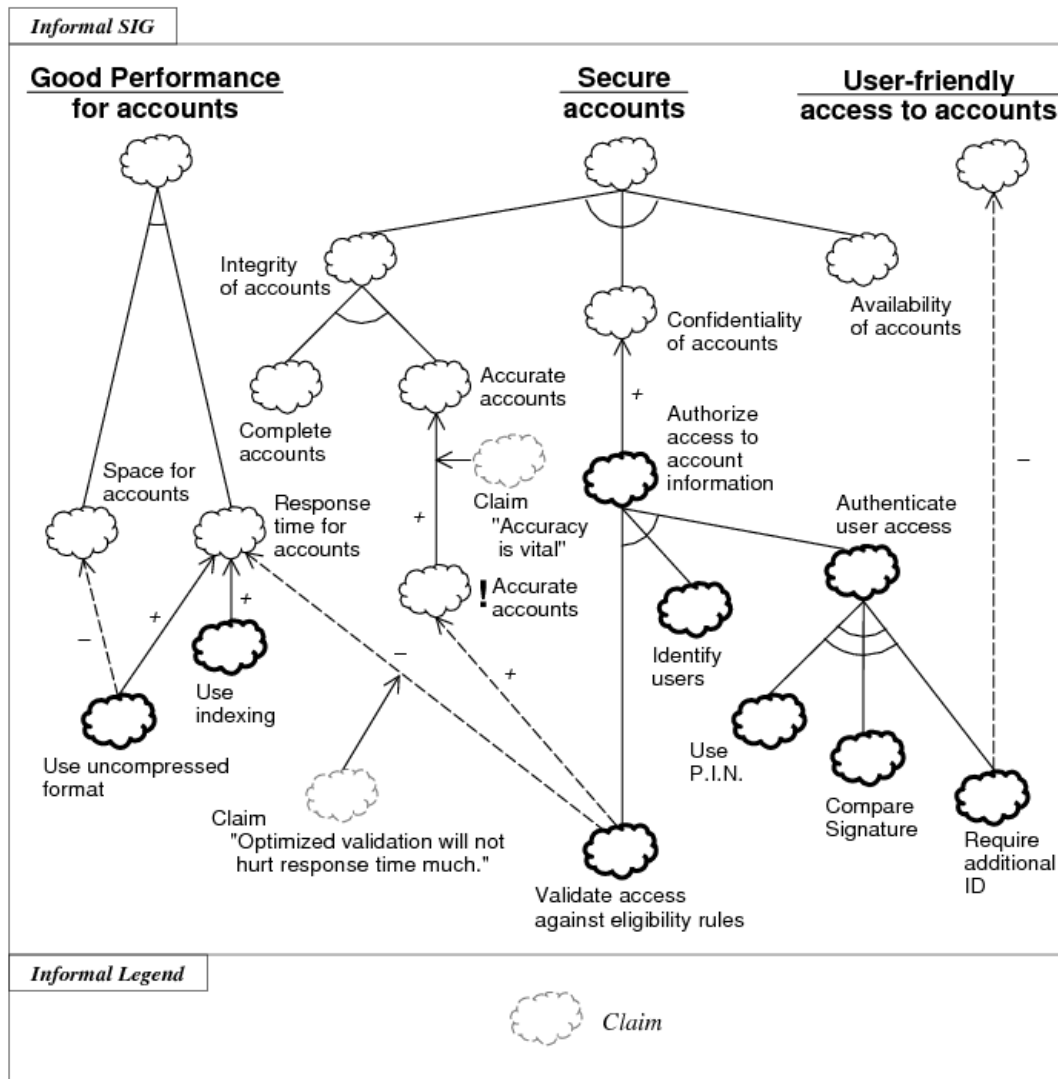


Figura 3.5 *Claim-Softgoal* (Chung *et al.*, 2000).

requisitos não-funcionais previstos pelo desenvolvedor ou analista para a qual a aplicação está sendo desenvolvida.

Não é necessário que esses passos sejam sequenciais, além disso, alguns podem iterar mais vezes durante o processo de concepção.

3.3 Operacionalização de requisitos não-funcionais

O processo de refinamento de um requisito não-funcional provê uma melhor interpretação do significado do mesmo, porém ainda não fornece um meio de alcançar um requisito não-funcional refinado. Conforme Chung em (Chung *et al.*, 2000), quando um requisito não-funcional é suficientemente refinado será possível identificar várias técnicas de desenvolvimento que possam ser utilizadas para executar este requisito não-funcional, depois será feita a escolha da solução específica para o sistema em questão.

Todavia, é importante frisar que existe uma lacuna entre um *softgoal* e as técnicas de desenvolvimento. Para romper esta lacuna é necessário observar algumas questões tais como análise de performance, ambigüidades, prioridades, informações do domínio da aplicação. Esses fatores deverão ser observados em vários passos no processo da operacionalização.

Segundo Chung em (Chung *et al.*, 2000), as operacionalizações não são limitadas a operações e funções. Como alternativa, operacionalizações podem ser correspondentes a dados e restrições da aplicação. Assim como os *softgoals*, as operacionalizações podem também ter contribuições positivas ou negativas.

De acordo com (Chung *et al.*, 2000) as operacionalizações podem ser definidas a partir de catálogos de técnicas de desenvolvimento. Os catálogos podem ajudar a procurar por possíveis operacionalizações.

A transição de requisito não-funcional para operacionalização é um passo crucial no processo, pois o requisitos não-funcionais precisam ser convertidos em algo que seja “implementável”. Contudo, inicialmente pode não ser possível converter requisitos em uma operacionalização concreta em apenas um passo, serão necessários refinamentos. No *NFR-Framework* as operacionalizações são tratadas como *softgoals* que podem novamente ser refinados em outros *softgoals* e decompostos em operacionalizações mais específicas.

Os catálogos também mostram possibilidades de decomposição de operacionalizações em operacionalizações mais especializadas. Eles também mostram as contribuições dessas operacionalizações em outras.

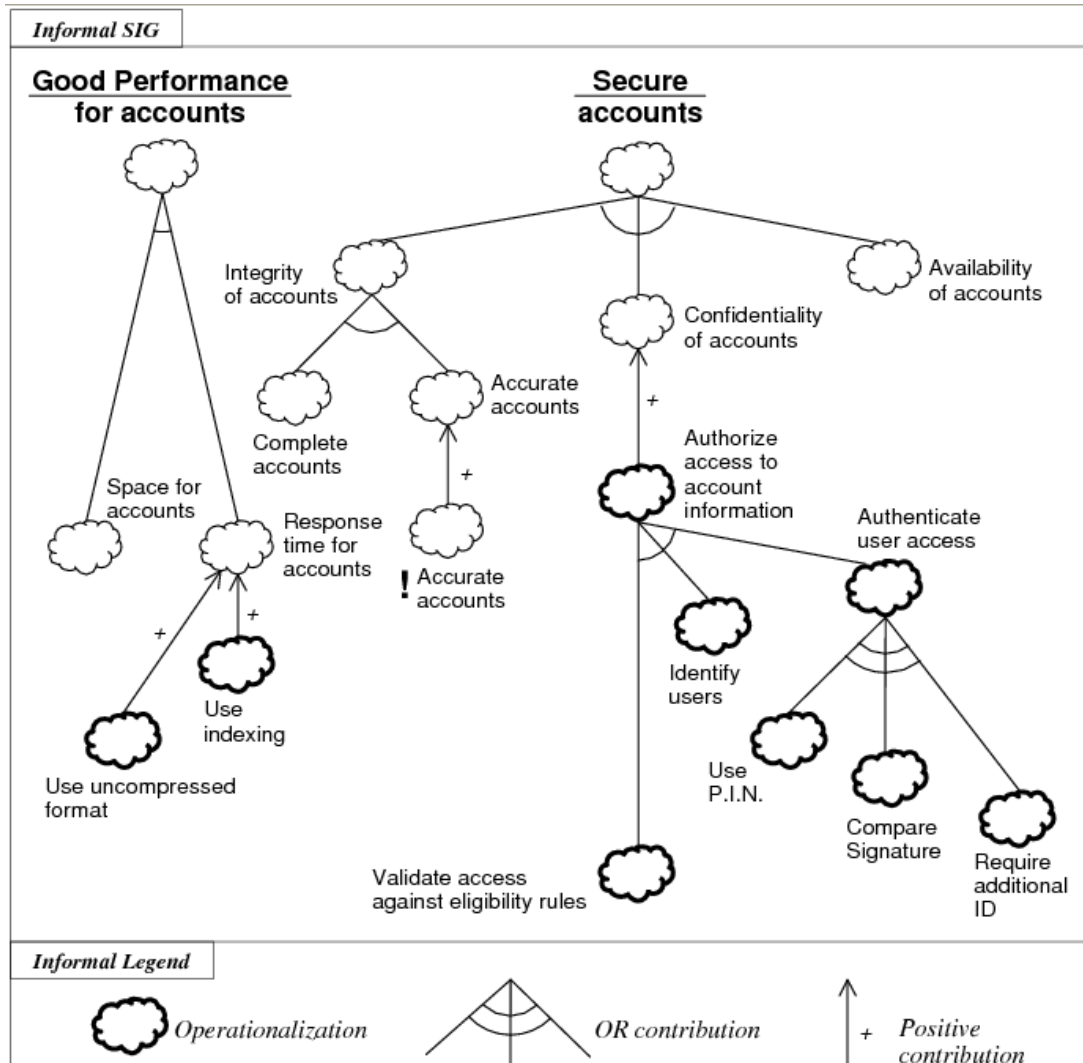


Figura 3.6 Identificação de possíveis operacionalizações para requisito não-funcional *Softgoal* (Chung *et al.*, 2000).

As contribuições AND e OR são arcos, as direções das contribuições são setas nas pontas dos traços, as contribuições são sinais de mais (+) e menos (-). As operacionalizações são destacadas com um contorno mais escuro.

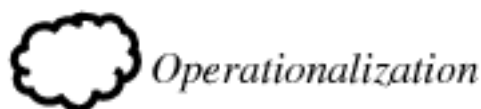


Figura 3.7 Operacionalizações (Chung *et al.*, 2000).

3.4 Catálogo de Usabilidade

O requisito não-funcional utilizado no estudo de caso e no caso de exemplo desta dissertação é usabilidade, por este motivo, se faz necessária a utilização de um catálogo deste requisito. Este catálogo refere-se à usabilidade para interfaces.

O trabalho de (Chung *et al.*, 2000) não prevê nenhum catálogo para este requisito, nem foi encontrado nenhum catálogo que utilizasse o *NFR-Framework* como base. Em (Cysneiros, 2007) havia um catálogo para usabilidade, porém esse catálogo utiliza o i* para modelá-lo e como este estudo baseia-se no *NFR-Framework*, não iremos utilizá-lo.

3.4.1 A Pesquisa

Para o desenvolvimento do catálogo usamos como base três conceitos de usabilidade, metas, princípios de *design* e heurísticas. As metas estão num nível de abstração mais alto, são fatores de satisfação e estabelecem critérios de usabilidade para aceitabilidade de um sistema (Preece *et al.*, 2005). Os princípios de *design* são lembretes do que fornecer e do que evitar durante o *design* da interface (Norman, 1988). Heurísticas são princípios da usabilidade utilizados na prática, servem para avaliar a aceitabilidade das interfaces (Nielsen, 1994).

Preece em seu livro (Preece *et al.*, 2005) descreveu seis metas de usabilidade baseando-se na idéia de que usabilidade é geralmente considerada como o fato que assegura que os produtos são fáceis de usar, eficientes e agradáveis. São elas:

- Ser eficaz no uso - refere-se a quanto um sistema é bom em fazer o que se espera dele;

- Ser eficiente no uso - refere-se à maneira como o sistema auxilia os usuários na realização das tarefas;
- Ser seguro no uso - implica proteger o usuário de condições perigosas e situações indesejáveis;
- Ser de boa utilidade - refere-se à medida na qual o sistema propicia o tipo certo de funcionalidade, de maneira que os usuários possam realizar aquilo de que precisam ou que desejam;
- Ser fácil de aprender - refere-se à quão fácil é aprender a usar o sistema;
- Ser fácil de lembrar como se usa - refere-se à facilidade de lembrar como utilizar o sistema.

Já os princípios de design descritos por Norman em (Norman, 1988) tratam de abstrações generalizáveis, destinadas a orientar *designers* a pensar sobre aspectos diferentes. Os princípios mais conhecidos referem-se a como determinar o que os usuários devem ver e fazer quando realizam tarefas. Os princípios de Norman são:

- Visibilidade - refere-se à maneira que as funções estão dispostas numa determinada interface ou produto. Quanto mais visíveis forem as funções, mais os usuários saberão como proceder;
- *Feedback* - refere-se ao retorno de informações a respeito da ação que foi feita e do que foi realizado. Está relacionado ao conceito de visibilidade;
- Restrições - refere-se à determinação das formas de delimitar o tipo de interação que pode ocorrer em um determinado momento;
- Mapeamento - refere-se à relação entre os controles e os seus efeitos no mundo;
- Consistência - refere-se a projetar interfaces de modo que tenha operações semelhantes e que utilizem elementos semelhantes para a realização de tarefas similares;
- *Affordance* - Termo utilizado para se referir ao atributo de um objeto que permite às pessoas saber como utilizá-lo.

Dentre os princípios de *design* descritos por (Norman, 1988), o último não será utilizado para a criação do catálogo, pois este se trata de um catálogo para interfaces.

O próprio autor argumenta que não faz sentido tentar projetar *affordances* reais para interfaces.

Seguindo o detalhamento dos trabalhos citados para a elaboração do catálogo, Nielsen em seu trabalho com usabilidade (Nielsen, 1994), criou dez heurísticas para este requisito não-funcional. São elas:

- Visibilidade do *status* do sistema - o sistema mantém os usuários sempre informados sobre o que está acontecendo, fornecendo um *feedback* adequado, dentro de um tempo razoável;
- Compatibilidade do sistema com o mundo real - o sistema fala a linguagem do usuário utilizando palavras, frases e conceitos familiares a ele, em vez de termos orientados ao sistema;
- Controle do usuário e liberdade - fornece maneiras de permitir que os usuários saiam facilmente dos lugares inesperados em que se encontram, utilizando “saídas de emergência” claramente identificadas;
- e padrões - evita fazer com que os usuários tenham que pensar palavras, situações ou ações diferentes que significam a mesma coisa;
- Prevenção de erros - onde possível, impede a ocorrência de erros;
- Reconhecimento em vez de memorização - tornar objetos, ações e opções visíveis;
- Flexibilidade e eficiência de uso - fornece aceleradores invisíveis aos usuários inexperientes, os quais, no entanto, permitem aos mais experientes realizar tarefas com mais rapidez;
- Estética e *design* minimalista - evita o uso de informações irrelevantes ou raramente necessárias;
- Ajuda e documentação - fornece informações que podem ser facilmente encontradas e ajuda mediante uma série de passos concretos que podem ser facilmente seguidos;
- Ajuda os usuários a reconhecer diagnosticar e recuperar-se de erros - utiliza linguagem simples para descrever a natureza do problema e sugere uma maneira de resolvê-lo.

Além de pesquisar as heurísticas que Nielsen criou, as metas de Preece e os princípios de Norman, foi consultada também a existência de algum outro catálogo para o requisito não-funcional usabilidade. No trabalho de (Cysneiros, 2007), o autor elabora um catálogo utilizando a notação i*.

3.4.2 O catálogo

Todos os trabalhos citados anteriormente foram observados para a criação do catálogo que utiliza o *NFR-Framework*. Este catálogo será mostrado por partes, pois a visualização dele em apenas uma figura não é apropriada por ser demasiado grande.

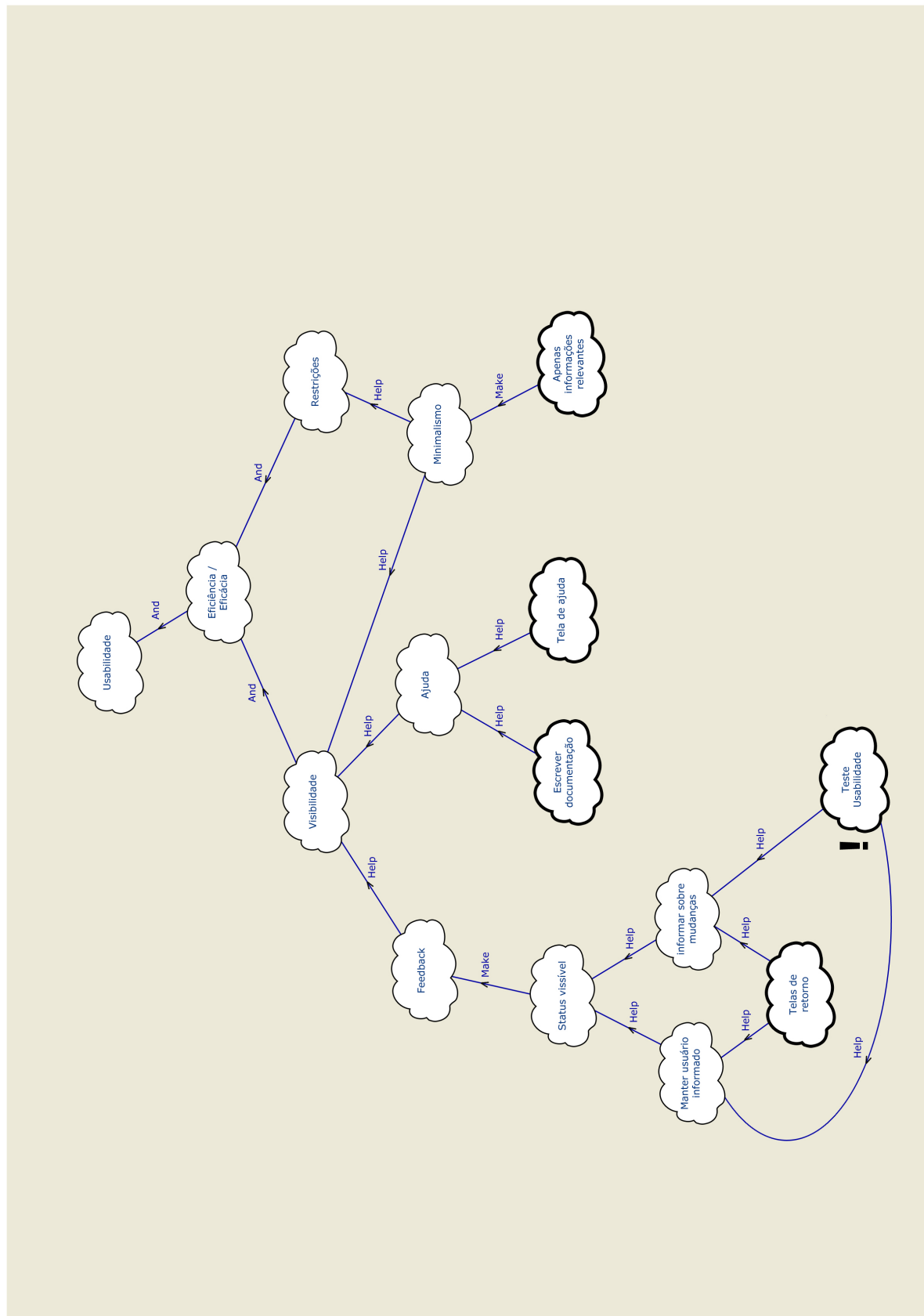
No catálogo, as metas do trabalho de (Norman, 1988) se transformaram nos quatro *softgoals* que estão imediatamente abaixo do requisito Usabilidade. Esse *softgoals* são: eficiência / eficácia, utilidade, segurança no uso e fácil de aprender e usar. Esses são os *softgoals* que estão na parte mais alta do SIG, pois tem um nível de abstração mais alto e ainda não são operacionalizáveis.

A figura 3.8 está focada na dimensão de “eficiência / eficácia”. Visibilidade e restrições (do trabalho de (Preece et al., 2005)) são os *softgoals* imediatamente abaixo que contribuem para atingir a meta eficiência / eficácia. Visibilidade deriva para *Feedback*, ajuda e design minimalista, enquanto Restrições deriva para o *design* minimalista. As operacionalizações destes *softgoals* foram criadas baseadas nas heurísticas de (Nielsen, 1994) e seus detalhamentos.

Relacionadas à “utilidade” (vide figura 3.9) a flexibilização e o livre controle do usuário são *softgoals* que contribuem para atingir esta meta principal e estão relacionadas ao trabalho de Nielsen. Restrições é um *softgoals* que foi associado à “eficiência e eficácia” e que também está associado à meta de utilidade, porém neste caso ele contribui negativamente. As operacionalizações destes *softgoals* também foram criadas baseadas nas heurísticas de (Nielsen, 1994) e seus detalhamentos.

Já o *softgoals* “segurança no uso” engloba a heurística de Nielsen “Tratamento de erros”, vide figura 3.10. Esta heurística é subdividida em duas partes, prevenção de erros e restabelecer de erros. Essas partes são detalhadas e operacionalizadas seguindo orientações do trabalho de (Nielsen, 1994).

E finalmente, na figura 3.11 o *softgoals* “fácil de aprender e usar” que une duas metas de (Preece et al., 2005) em um *softgoal* só. Esse *softgoal* é expandido para outros *softgoals*, consistência e mapeamento, que estão relacionados ao trabalho de (Norman, 1988). Como aconteceu em outras partes, as operacionalizações destes *softgoals* foram criadas baseadas nas heurísticas de (Nielsen, 1994) e seus detalhamentos.



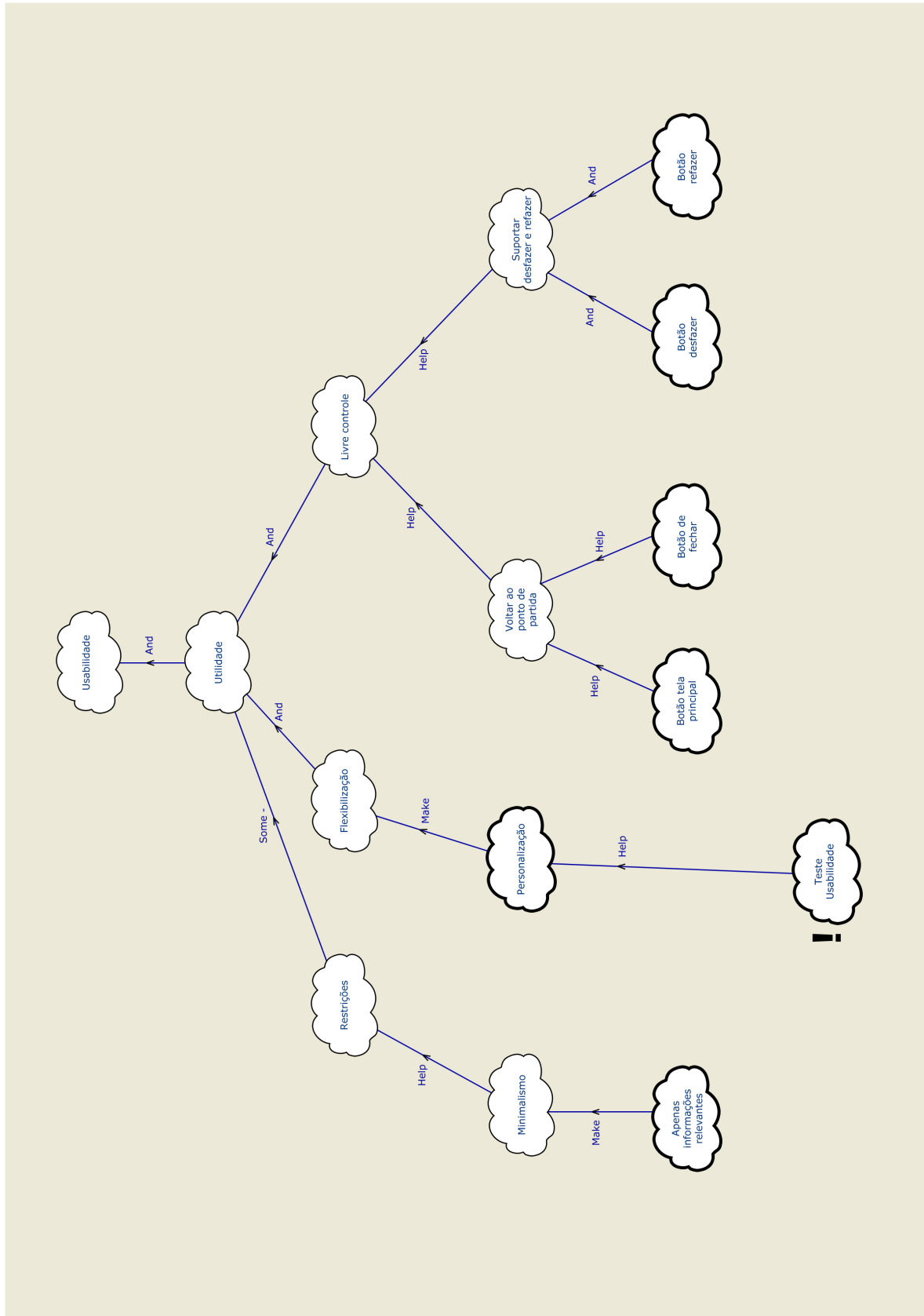
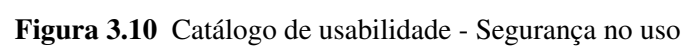


Figura 3.9 Catálogo de usabilidade - Utilidade



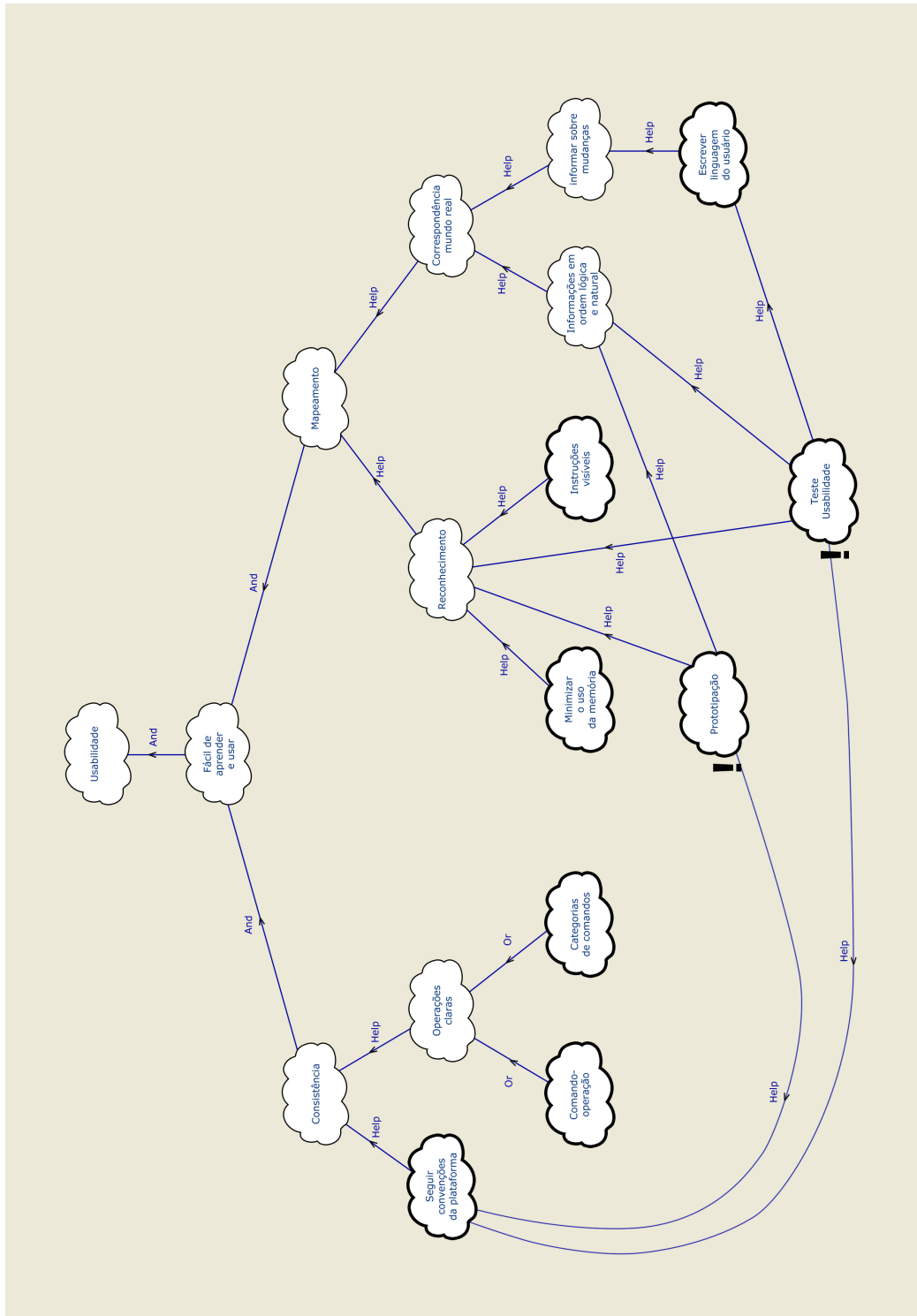


Figura 3.11 Catálogo de usabilidade - Fácil de aprender e usar

3.4.3 Considerações

Este catálogo não tem a pretensão de esgotar todas as possibilidades de operacionalizações que satisfaçam o requisito não-funcional usabilidade, nem muito menos dita de forma inflexível os grupamentos que foram feitos. Este catálogo tem objetivo de dar um passo inicial na criação de um catálogo genérico e mais completo para este requisito utilizando como base o *NFR-Framework*.

Conforme dito neste capítulo, os catálogos gerados pelo *NFR-Framework* podem e devem ter refinamentos até que se chegue num ponto em que satisfaça quem irá utilizá-los.

3.5 Considerações Finais

Acredita-se que para este trabalho a utilização do *NFR-Framework* dentro do contexto da modelagem de processos de negócio irá contribuir para uma melhoria na qualidade da especificação, modelagem, análise e decisões tomadas ao longo do processo de desenvolvimento do sistema de informação. O *NFR-Framework* torna explícitas as ligações e as soluções entre as *softgoals* e as operacionalizações coerentes em cada projeto. Isto permite uma visualização clara dos objetivos e, por conseguinte, analisar sistematicamente, o quanto o projeto está alinhado com as especificações do usuário. A sua utilização combinada, juntamente com a modelagem de processos de negócio, irá ajudar a melhorar a comunicação com entre os analistas de negócios e analistas de sistema, alinhando assim suas expectativas de maneira completa.

*“Making the simple complicated is commonplace;
making the complicated simple, awesomely simple,
that’s creativity.”*

Charles Mingus

4

Abordagem BPMNFR

As notações de modelagem de processos de negócio podem capturar as funcionalidades de um *software*, porém não capturam ou identificam requisitos não-funcionais (vide capítulo 2). A notação **BPMN**, portanto não suporta requisitos não-funcionais explicitamente. Todavia, os requisitos não-funcionais são peças chaves dentro da modelagem e elicitação de requisitos (vide capítulo 3). Erros provocados por não se lidar convenientemente, ou simplesmente não lidar com requisitos não-funcionais são apontados como estando entre os mais caros e difíceis de corrigir.

Neste capítulo é proposta uma abordagem, chamada BPMNFR, para integrar requisitos não-funcionais à modelagem dos processos de negócio. Esta abordagem tem uma notação mais intuitiva para todos os usuários envolvidos no processo além de tratar os requisitos não-funcionais.

Assim, propõe-se estender a notação **BPMN** e prover em nível de negócio a modelagem dos requisitos não-funcionais, através da inserção das operacionalizações de requisitos não-funcionais nos diagramas de processos de negócio.

Para explicar a abordagem BPMNFR será utilizado um pequeno caso de exemplo relacionado ao processo de desenvolvimento de objetos de aprendizagem.

4.1 A abordagem BPMNFR

A abordagem BPMNFR foi elaborada para que mantivesse a legibilidade da notação utilizada, **BPMN**, e, bem como, considerar os requisitos não-funcionais ausentes na notação original.

A primeira tentativa de solucionar o problema(vide figura 4.1) tinha os seguintes passos: primeiramente, a criação de um **BPD** com a modelagem da aplicação; depois a identificação dos requisitos não-funcionais; a descoberta das operacionalizações através

dos catálogos criados a partir do *NFR-Framework*; e finalmente, a inserção de maneira direta dessas operacionalizações dentro do primeiro **BPD** gerado. Desta forma cobriria a deficiência da notação, porém não deixaria assinalado a que requisito não-funcional àquela operacionalização se referia. Como numa modelagem podem existir vários requisitos não-funcionais que precisam ser tratados, não haveria como saber a qual requisito não-funcional a operacionalização se tratava. Desta forma ainda existia uma passo que precisava ser dado para que houvesse essa correspondência.

Depois da primeira tentativa, chegou-se na evolução da abordagem BPMNFR. A nova abordagem elaborada consiste em quatro fases, são elas:

- Construção do **BPD**;
- Inserção de rótulos nas atividades do **BPD** que representam os requisitos não-funcionais identificados para aquela atividade;
- Utilização, extensão ou criação de catálogos dos requisitos não-funcionais;
- Inserção das operacionalizações no **BPD**, agrupadas no grupo lógico rotulado.

A figura 4.2 representa graficamente o processo que envolve a abordagem BPMNFR, o fluxo de atividades além dos artefatos que são gerados a cada atividade.

Durante o processo, ao todo, são gerados 4 artefatos como resultado de cada atividade. Esses artefatos serão detalhados nas seções seguintes. As atividades deste processo podem iterar mais vezes durante o processo de modelagem, portanto não necessariamente os primeiros artefatos produzidos serão os artefatos definitivos na modelagem.

4.1.1 Construção do **BPD**

Na primeira fase, é feita a modelagem dos processos de negócio. Esses processos de negócio podem estar relacionados a um processo de desenvolvimento de uma aplicação, ou estarem de fato relacionados diretamente com a modelagem da própria aplicação, onde as funcionalidade da mesma são tratadas como atividades dentro da notação e os atores como *lanes* dentro de um *pool*.

Independentemente do tipo de modelagem que for feita, o diagrama de processos gerado deverá ser validado pelo analista de negócios e analista de sistemas, a fim de garantir que o processo foi adequadamente modelado e entendido. Esta primeira fase pode ter mais de uma interação com a finalidade de refinar o diagrama de processos. O artefato gerado desta atividade dentro do processo da abordagem BPMNFR é um **BPD**.

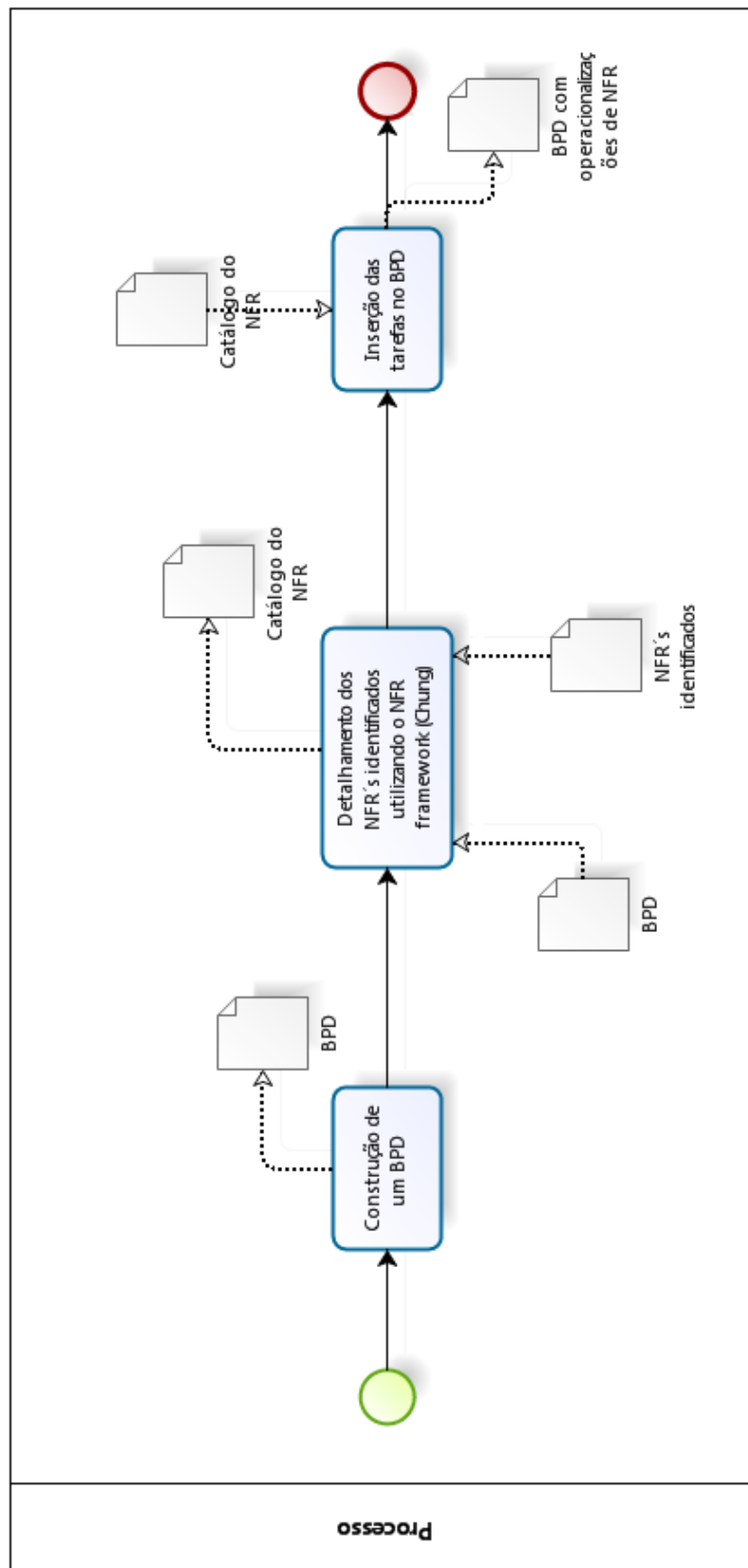
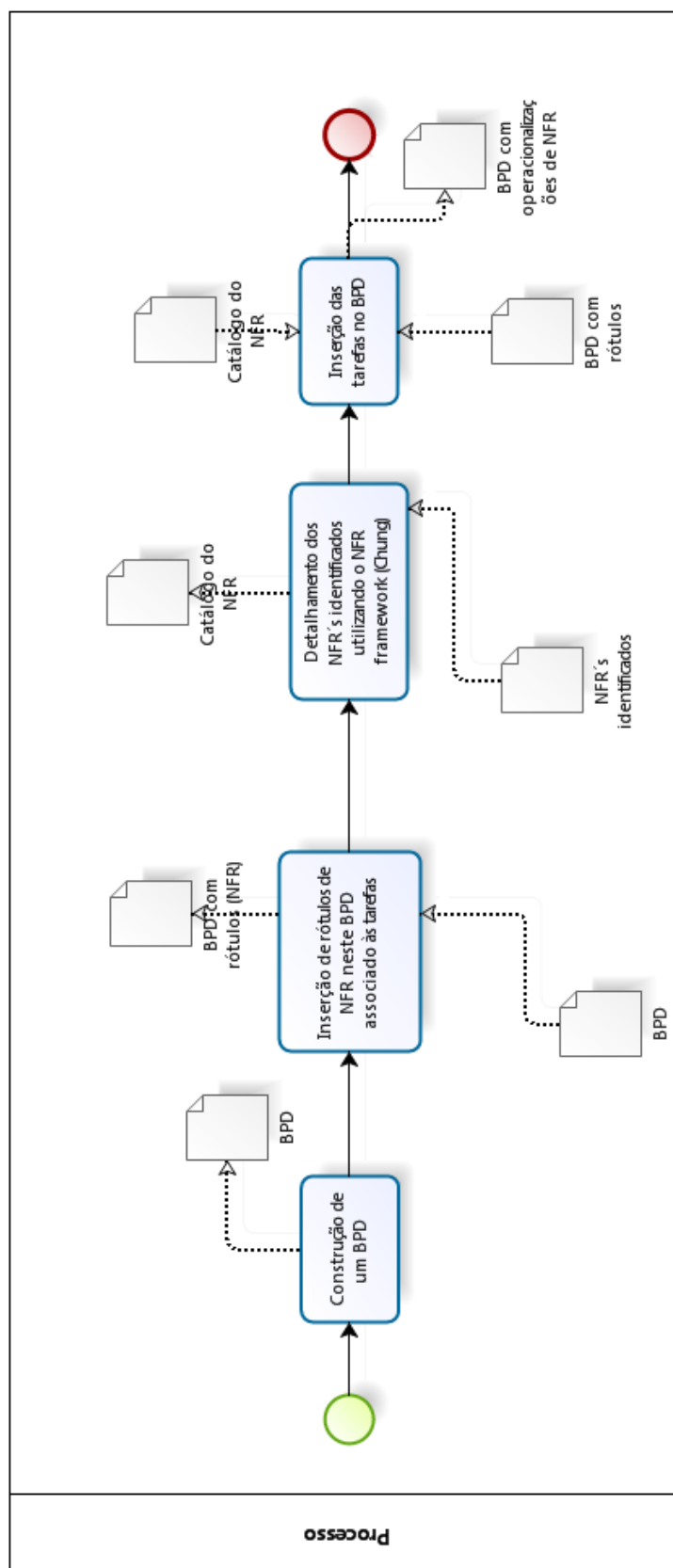


Figura 4.1 Figura da Abordagem BPMNFR inicial para inserção de requisito não-funcional em BPMN.



Caso de exemplo

Um caso de exemplo foi utilizado para exemplificar o uso da abordagem BPMNFR. Esse caso de exemplo trata do processo de desenvolvimento de Objetos de Aprendizagem. Objetos de Aprendizagem são quaisquer recursos digitais que possa ser reutilizado e ajude na aprendizagem (Wiley, 2002).

Para os objetos de aprendizagem algumas características são essenciais, tais como, ser adaptáveis às necessidades, habilidades e estilos cognitivos do aprendiz, devem ser acessíveis de qualquer lugar a qualquer hora. Para que se obtenha esse resultado, é importante que haja uma equipe multidisciplinar no desenvolvimento dos objetos de aprendizagem. A integração de equipes multidisciplinares é fator necessário para a produção de componentes de aprendizagem. Faz-se necessária a integração de diversos profissionais especialistas em várias áreas do conhecimento: como pedagogos, jornalistas, designers, por exemplo. Estes precisam comunicar-se e trabalhar colaborativamente (Gomes *et al.*, 2006).

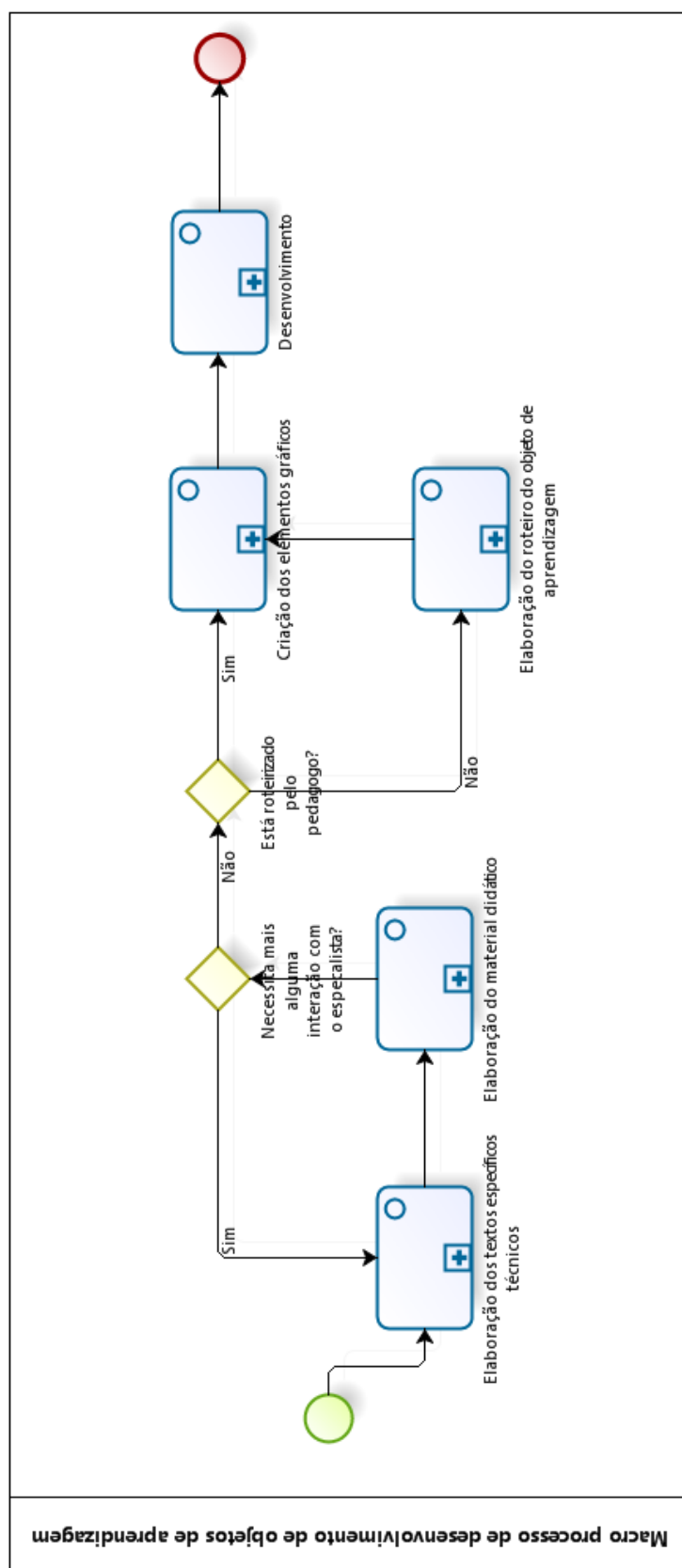
No caso de exemplo, foi utilizado o processo de desenvolvimento de Objetos de Aprendizagem. Este processo é modelado utilizando a notação de BPMN, sendo gerado desta forma o BPD da figura 4.3.

Neste BPD, foram levados em consideração os macro-processos de desenvolvimento de Objetos de aprendizagem utilizado por uma empresa especializada neste tipo de desenvolvimento. Este processo é um refinamento do processo criado por (Gomes *et al.*, 2006). Nele é possível observar os macro-processos de Elaboração dos textos Técnicos, Elaboração do material didático, Elaboração do roteiro, Criação dos elementos gráficos e Desenvolvimento. Existem também dois *gateways* que verificam a necessidade ou não de mais uma iteração com o sub-processo anterior.

Não iremos detalhar cada subprocesso já que este caso é apenas um exemplo para mostrar de forma mais clara as atividades do processo da abordagem BPMNFR. As próximas fases serão descritas e detalhadas nas subseções seguintes.

4.1.2 Rótulos de requisitos não-funcionais no BPD

Nesta fase, o artefato de entrada é o digrama (BPD) gerado na fase anterior. Os analistas de negócios irão identificar os requisitos não-funcionais pertinentes à aplicação, pois o que determina a existência ou não de um requisito não-funcional na aplicação é o seu domínio. O trabalho de (Bittencourt, 2009) mostra uma sistematização para identificação dos requisitos não-funcionais.



Após a identificação dos requisitos não-funcionais relevantes para o determinado domínio é necessário relacionar as atividades do **BPD** com os requisitos não-funcionais. As atividades que forem relacionadas com determinado requisito não-funcional deverão ser rotuladas com a inicial do requisitos não-funcionais o que ela está associada. Se existir mais de um requisito não-funcional que deve ser associado a uma atividade dentro do **BPD**, devem-se colocar tantos rótulos quanto a quantidade de requisitos não-funcionais associados.

No nosso caso de exemplo, usaremos o requisito não-funcional mais crítico para a aplicação, já que a finalidade neste momento é apenas apresentar os processos da abordagem BPMNFR, como dito anteriormente. O requisito não-funcional identificado como crítico e mais relevante para a aplicação foi usabilidade. Desta maneira, o requisito não-funcional de usabilidade foi associado a algumas atividades dentro do BPD utilizando os rótulos como mostra a figura 4.4.

As atividades que tiveram a usabilidade como requisito não-funcional associado foram elaboração do roteiro, criação dos elementos gráficos e desenvolvimento. Nestas três atividades a preocupação com usabilidade é crítica e por isso o requisito não-funcional foi associado.

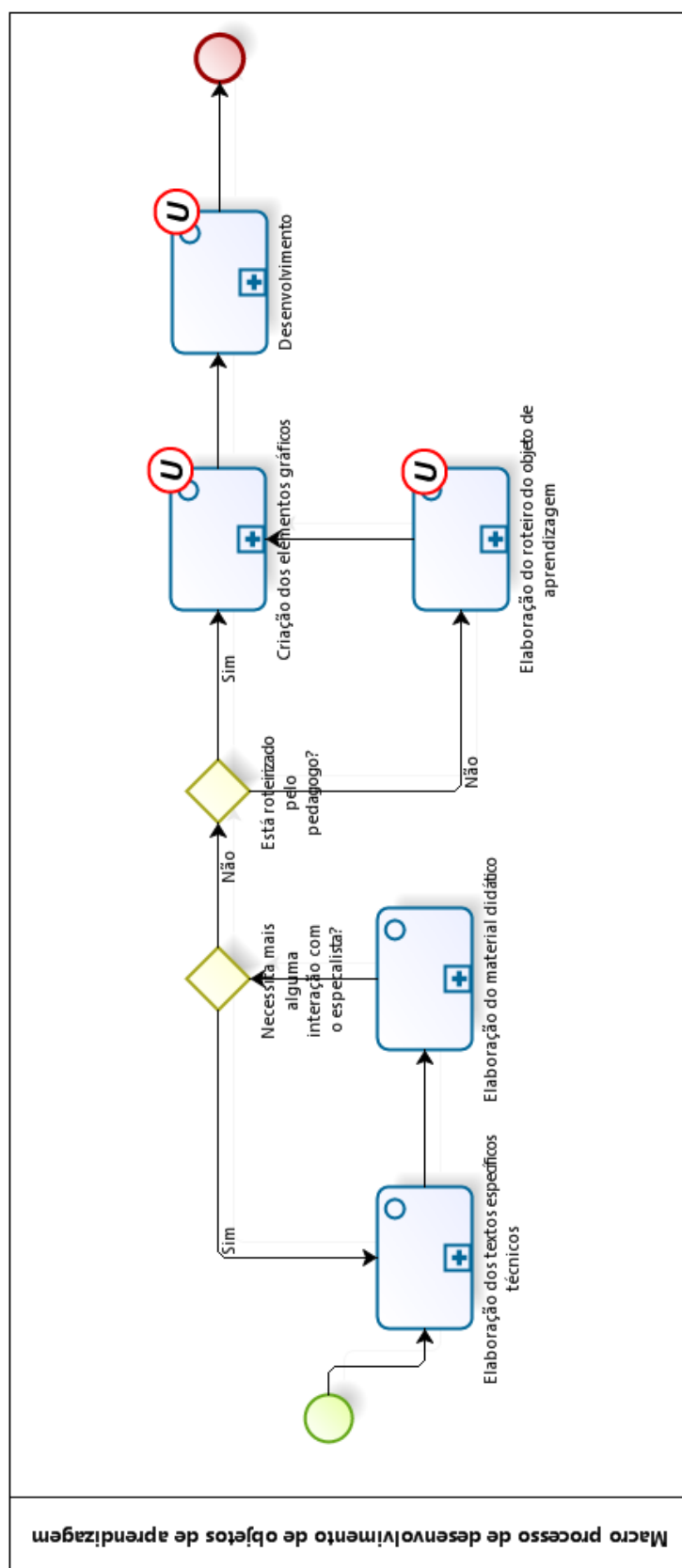
O artefato gerado nesta atividade é um **BPD** rotulado com os requisitos não-funcionais associados a atividades. Identificados os requisitos não-funcionais e associadas as atividades com seus devidos rótulos, o próximo passo é descobrir quais as operacionalizações desse(s) requisito(s) não-funcional(ais). Isso é mostrado na próxima seção.

4.1.3 Detalhamento do requisito não-funcional

Os artefatos de entrada para esta atividade são os requisitos não-funcionais identificados na etapa anterior. Com os requisitos não-funcionais identificados e relacionados às atividades do **BPD** é necessário que sejam descobertas as operacionalizações destes requisitos não-funcionais.

Para detalhamento dessas operacionalizações usa-se o *NFR-Framework* que já foi apresentado no segundo capítulo. Se já existir algum catálogo em (Chung et al., 2000) para o requisito não-funcional em questão, ele poderá ser utilizado. Porém deve ser adaptado para levar em consideração as particularidades do domínio da aplicação. Se o catálogo não existir, deve-se criá-lo a partir do *NFR-Framework*, também observando o domínio da aplicação.

Quando houver mais de um requisito não-funcional envolvido, é necessário que se tenha uma maior atenção na criação do catálogo. Essa atenção maior é importante,



pois podem existir interdependências entre os requisitos, ocasionando desta forma uma interferência, ou não, positiva ou negativa numa operacionalização ou *softgoal* de outro requisito diretamente. Esta interferência pode ajudar ou prejudicar a satisfazer este outro requisito.

Para o caso de exemplo deste capítulo e o estudo de caso utilizado no capítulo 5 desta dissertação, foi necessária a criação de um catálogo do RNF utilizado, usabilidade (vide capítulo 3), já que (Chung *et al.*, 2000) não prevê nenhum catálogo para este requisito, nem foi encontrado nenhum catálogo que utilizasse o *NFR-Framework* como base.

Esta atividade gera um artefato de saída que é o catálogo dos requisitos não-funcionais. Com o catálogo do requisito não-funcional criado e/ou refinado o próximo passo é identificar quais operacionalizações desse catálogo serão relevantes para o domínio da aplicação, pois não é necessário o uso de todas as operacionalizações sempre. Para exemplificar, utilizaremos apenas duas operacionalizações do catálogo de usabilidade genérico visto na figura ??, a prototipação e o teste de usabilidade.

4.1.4 Inserção das operacionalizações no BPD

Os artefatos de entrada para esta atividade do processo da abordagem BPMNFR são o BPD modelado na primeira atividade e os catálogos dos requisitos nãofuncionais gerados a partir do passo anterior. Finalmente o BPD, previamente modelado, deverá ser modificado, isto é, alguns elementos devem ser removidos ou introduzidos de acordo com as operacionalizações identificadas.

Nem todas as operacionalizações deverão estar neste último passo, cabe ao analista de negócios juntamente com o analista de sistemas decidir quais as operacionalizações serão pertinentes aos domínio. Como dito antes, já existem trabalhos, como por exemplo o de (Bittencourt, 2009), que mostram como identificar essas operacionalizações com o envolvimento dos *stakeholders*.

As operacionalizações escolhidas se transformarão em atividades ou macro-processos dentro do BPD. Essas atividades/macro-processos devem estar agrupadas num grupo lógico que será rotulado com a letra inicial do requisito não-funcional a que ele se refere. Esse grupo lógico pode estar no macro-processo definido ou dentro dos subprocessos. O grupo lógico deve estar associado à atividade que estava antes rotulada. Essa associação pode ser feita através do elemento de associação, ou pode estar dentro de um subprocesso da atividade antes rotulada.

Aplicando ao exemplo, a figura 4.5 mostra o novo processo de negócio. As operacionalizações escolhidas para integrarem este BPD foram prototipação e teste de

usabilidade. Estas operacionalizações deverão estender esse **BPD** a partir da dos rótulos criado no segundo passo da abordagem BPMNFR. Estes rótulos estão no macro-processo de “Elaboração do roteiro do objeto de aprendizagem”, “Criação dos elementos gráficos” e “Desenvolvimento”.

A primeira operacionalização está associada ao macro-processo de “Elaboração do roteiro do objeto de aprendizagem” e será o teste de usabilidade. Ele está associado ao macro-processo através de um objeto de um fluxo de seqüência e está envolto no grupo lógico rotulado.

A próxima operacionalização adicionada foi a “Prototipação”. Esta operacionalização está associada ao macro-processo de “Criação dos elementos gráficos” através de um fluxo de seqüência e também está envolto no grupo lógico.

A última operacionalização inserida foi o “Teste de usabilidade” novamente e está associada ao macro-processo de desenvolvimento através de um fluxo de seqüência e novamente envolto num grupo lógico.

O diagrama foi modificado como o resultado da introdução do grupo lógico. Novas tarefas foram definidas (como resultado das operacionalizações dos requisitos não-funcionais) e nenhuma tarefa foi retirada. Os macro-processos novos como teste de usabilidade e prototipação estão envoltas pelo grupo lógico.

Note que o grupo lógico está associado à atividade que o gerou através do elemento associação.

Para facilitar o entendimento, foi criada a tabela 4.1 com quatro colunas. Na primeira é colocada a tarefa que foi rotulada na segunda fase da abordagem BPMNFR, na segunda é colocado o requisito não-funcional que se refere à tarefa rotulado, na terceira coluna se ver as operacionalizações que foram encontradas para o requisito e que estão diretamente ligadas a tarefa que foi rotulada, na quarta coluna se ver as classificações das operacionalizações.

Através desta tabela é possível fazer a correspondência entre as tarefas rotuladas do **BPD** e as operacionalizações que foram criadas a partir do requisito não-funcional. Nesta tabela as operacionalizações têm duas classificações.

A primeira classificação está relacionada ao lugar que as operacionalizações ocupam dentro do **BPD** obtido através da execução do último passo da abordagem BPMNFR. Essas operacionalizações podem ser de dois tipos:

- A - são as operacionalizações que estão associadas através de um fluxo de seqüência à tarefa anteriormente rotulada;

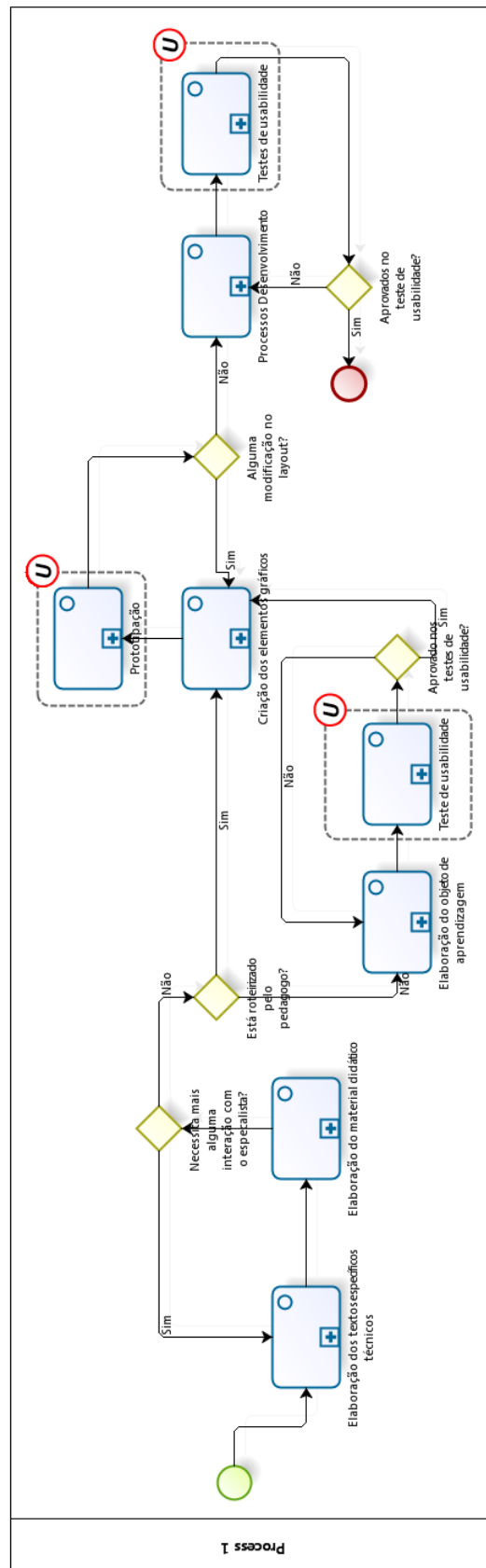


Figura 4.5 BPD com operacionalizações

- S - são as operacionalizações fazem parte do sub-processo da tarefa anteriormente rotulada, estando, portanto dentro do sub-processo desta tarefa.

A segunda classificação está relacionada à natureza da operacionalização, isto é, se a operacionalização se tornou uma tarefa ou um macro-processo dentro do **BPD**.

- T - Se a operacionalização se tornar uma tarefa única;
- MP - Se a operacionalização se tornar um macro-processo com tarefas internas.

Tarefa rotulada	RNF associado	Operacionalizações	Tipo
Criação dos elementos gráficos	Usabilidade	Prototipação	A / MP
Elaboração do objeto de aprendizagem	Usabilidade	Teste de usabilidade	A / MP
Processo de desenvolvimento	Usabilidade	Teste de usabilidade	A / MP

Tabela 4.1 Correspondências de operacionalizações

Depois da inserção das operacionalizações em forma de tarefas dentro de um **BPD** e agrupadas através de um grupo lógico a abordagem BPMNFR sugere uma maneira explícita de prover em nível de negócio a modelagem dos requisitos não-funcionais.

4.2 Análise comparativa com outras abordagens

Nesta seção são apresentados alguns trabalhos que também fazem a modelagem dos processos de negócio e de requisitos não-funcionais.

4.2.1 i*

A modelagem i* - denominação associada a “intencionalmente distribuído” - é uma técnica de modelagem faz uso dos conceitos do *NFR-Framework* (Chung *et al.*, 2000), especialmente dos *softgoals*, que são utilizados nos dois modelos: Strategic Dependency (**SD**) (*Strategic Dependency*) e Strategic Rationale (**SR**) (*Strategic Rationale*) para elaborar modelos de negócio e auxiliar o redesenho de processos (Chung *et al.*, 2000). O **SD** descreve um processo através da dependência de relacionamentos entre atores para alcançar objetivos, desempenhar tarefas e fornecer recursos, portanto, é um modelo direcionado a capturar motivações e intenções.

No *framework* i* (Yu, 1995), as organizações constituem-se de atores sociais que dependem uns dos outros para satisfazer um *softgoals*, alcançar objetivos, executar tarefas

e fornecer recursos. A técnica i^* faz uma análise a partir dos objetivos relativos ao estado do negócio que um ator deseja alcançar, mas também dos *softgoals* relativos às condições, que o ator deseja obter, delineadas pelas suas intenções e razões.

Os diagramas que utilizam o i^* podem se tornar muito complexos e deverão ser utilizados quando for necessária uma maior granularidade e refinamento dos processos na modelagem.

4.2.2 Santander e Castro

Os autores apresentam, em seu trabalho (Santander and Castro, 2000), diretrizes de como obter diagramas de casos de uso que satisfaçam aos objetivos organizacionais, a partir de processos modelados utilizando a abordagem i^* . O ator é o elemento comum e fundamental entre i^* e o diagrama de casos de uso, portanto o processo de construção do diagrama inicia-se com a identificação dos atores no modelo SD, ressaltando que nem todo ator no SD será ator no diagrama. As dependências do modelo SD, que ocorrem entre atores, objetivos, tarefas e *softgoals*, traduzem-se parcialmente em atividades e funções nos casos de uso.

Os autores visualizam a possibilidade de identificar aspectos não-funcionais através das dependências de *softgoals* entre os atores e afirmam que os requisitos não-funcionais estão associados direta ou indiretamente aos diagramas de casos de uso. Os aspectos não-funcionais capturados são anexados através de uma descrição textual aos diagramas, mas os autores não indicam como transformar as intenções entre os atores em requisitos não-funcionais de baixo nível e como associá-los aos diagramas. Os requisitos não-funcionais são tratados de forma secundária e não sistemática.

4.2.3 Pavlovski e Zou

O objetivo é identificar requisitos não-funcionais a partir de restrições do negócio identificadas através da modelagem de processo de negócio. (Pavlovski and Zou, 2008) utiliza BPMN para modelar processos e propõem uma extensão à notação para representar requisitos não-funcionais através da condição de operação que é descrita através do *control case*. A condição de operação representa uma restrição associada a uma atividade no modelo de processo de negócio. O *control case* é uma descrição textual da condição de operação onde são descritos os riscos ao negócio e os controles que são aplicados para mitigar os riscos.

De acordo com os autores é possível identificar, durante o processo de modelagem,

os seguintes requisitos não-funcionais em alto nível: desempenho de tarefas, políticas de segurança, disponibilidade das atividades ou processos, tempo de resposta das atividades ou processos, restrições organizacionais e regulatórias e a qualidade da interação do usuário com a atividade. O foco durante a análise do negócio é identificar os riscos que possam ameaçar os objetivos do negócio e que têm origem nas restrições do negócio.

O método proposto oferece um meio de considerar as restrições durante a modelagem de processos como prováveis fontes de requisitos não-funcionais e propõe que o *control case* seja um canal que facilite a comunicação sobre requisitos não-funcionais entre o analista e os *stakeholders*. A identificação destes requisitos permanece em alto nível, não há representação do *rationale* ou a decomposição dos mesmos.

4.2.4 Abordagem BPMNFR

A abordagem BPMNFR apresentada nesta dissertação utiliza o **BPMN** para a descrição e modelagem dos processos de negócio e o **NFR-Framework** como base para obtenção das operacionalizações dos requisitos não-funcionais. O objetivo é identificar requisitos não-funcionais a partir de restrições do negócio identificadas através da modelagem de processo de negócio.

Esta abordagem também deixa modelados de forma explícita os requisitos não-funcionais durante todo o seu processo, deste quando esses requisitos são pensados em alto nível, até quando eles são operacionalizados. Existe inclusive um mapeamento para identificar qual requisito não-funcional de alto nível se tornou numa operacionalização.

4.3 Considerações Finais

A abordagem BPMNFR proposta visa a inserção de requisitos não-funcionais durante a fase de modelagem de negócio, porém com a facilidade de uso de uma notação que seja de fácil entendimento entre TI e clientes/usuários, conseguindo, desta forma, que os objetivos de negócio sejam melhor alcançados.

Para descobrir as operacionalizações dos requisitos não-funcionais usa-se um *framework* baseado em metas que, como visto no capítulo 3, é mais adequado para este tipo de requisito.

A desvantagem de uso desta abordagem BPMNFR é a utilização de dois modelos, um que utiliza o **BPMN** e outro que utiliza o **NFR-Framework**, durante seu processo. Portanto o analista de sistemas deverá conhecer e utilizar duas ferramentas durante o processo de modelagem.

A limitação desta abordagem se dá em relação aos catálogos, pois se não existirem catálogos genéricos que utilizem o *NFR-Framework* para um determinado requisito não-funcional, o analista de sistemas terá de criar um.

O capítulo seguinte apresentará um estudo de caso aplicado em uma situação real para avaliar a eficácia da abordagem BPMNFR.

*“Life is about not knowing, having to change,
taking the moment and make the best of it.”*

Gilda Radner

5

Estudo de caso

Este capítulo apresenta uma aplicação prática da abordagem BPMNFR proposta em um ambiente de desenvolvimento real. Serão descritos o contexto do ambiente no qual o experimento ocorreu, a sua realização e as considerações finais. É importante observar, porém, que este estudo aconteceu após a aplicação ficar pronta, portanto foi feita a modelagem do estudo de caso sem os requisitos não-funcionais e depois eles foram inseridos.

5.1 Contextualização

O Ministério da Educação (Ministério da Educação (**MEC**)) tem iniciado diversos projetos na área de tecnologia na educação com o intuito de incentivar a modernização do ensino e das técnicas pedagógicas. O **MEC** tem percebido que a instrumentação da sala de aula com métodos e ferramentas mais modernas é uma necessidade e prioridade para o sistema de ensino brasileiro. Esta modernização está sendo pautada em diversas iniciativas como os programas de laboratório de informática nas escolas - ProInfo; o Portal do Professor que possibilita o compartilhamento de conteúdo digital entre educadores; dentre outras. Neste contexto, partindo de testes já iniciados pelo Centro de Experimentação em Tecnologia Educacional - Centro de Experimentação em Tecnologia Educacional (**CETE**), da Secretaria de Educação a Distância - Secretaria de Educação a Distância (**SEED**) do **MEC**, foi desenvolvida uma prova de conceito, um protótipo tecnológico de uma solução portátil multimídia que agrega funcionalidades de "projektor, ou display inteligente", de baixo custo e uso simplificado, focado na dinâmica e uso de sala de aula. A combinação de funcionalidades de sistema de projeção e computador permite a conexão com a Internet, portabilidade, manipulação de conteúdo multimídia, desenvolvimento de tarefas colaborativas, apresentação de conteúdos educacionais, entre outras funções.

Assim, o **MEC** convidou a Universidade Federal de Pernambuco (Universidade Federal de Pernambuco (**UFPE**)), através do Secretaria de Ensino a Distância da UFPE (**SEAD**) (Secretaria de Ensino a Distância da **UFPE**), que conjuntamente com o Centro de Informática da **UFPE**, aceitaram o desafio de propor e desenvolver um protótipo que atendesse aos anseios do **MEC**. Dentro deste contexto, representantes do **MEC** apresentaram para a coordenação do projeto na **UFPE** um protótipo inicial que serviu como referência para o desenvolvimento de tal tecnologia. O protótipo final deveria ser validado pelo **MEC** e industrializado. O estudo de caso deste trabalho ocorreu no contexto desta parceria entre o **MEC** e a **UFPE**. Para realização do projeto seria necessário desenvolver um protótipo de um equipamento que reunisse as características especificadas pelo **MEC** bem como um *software* robusto que pudesse suportar estas características e uma interface que suportasse as necessidades específicas do usuário. Portanto, configura o desenvolvimento de um sistema de *hardware*, *software* e interface, ambiente perfeito para aplicação da abordagem BPMNFR, proposta no capítulo 4 deste trabalho. Durante um ano uma equipe técnica formada por 20 pessoas trabalhou dedicada ao projeto, que foi denominado Lampejo. O grupo era totalmente heterogêneo e multidisciplinar e foi organizado de acordo com suas especialidades nos diversos subsistemas do projeto, como será descrito nas seções posteriores.

5.2 Objetivos do estudo de caso

Como dito no capítulo 1, a análise de requisitos tem sido amplamente reconhecida com um fator crítico de sucesso de projetos de *software* (Group, 1995). Se não forem devidamente elicitados, os requisitos podem levar o projeto ao fracasso. No entanto, a experiência prática demonstra que os problemas podem facilmente ser identificados nesta fase.

Há dificuldade em elicitar e avaliar os requisitos não-funcionais de uma forma estruturada e compreensiva, principalmente quando envolve vários *stakeholders* (Wagner *et al.*, 2008). Adicionalmente, os modelos de requisitos propostos pelos analistas de sistemas (representando a tecnologia da informação) e utilizados para interagir com os usuários podem gerar dificuldades de compreensão, já que os usuários finais não dominam a linguagem de representação de tecnologia da informação (de La Vara *et al.*, 2008).

Para tentar resolver este problema, os analistas de sistemas têm utilizado notações que são de fácil entendimento pelo usuário, pois elas são intuitivas, tais como **BPMN**. Porém mesmo utilizando esta notação ainda não é possível expressar todas as necessidades dos

usuários, pois a notação não suporta a modelagem de requisitos não-funcionais.

Portanto, tendo em vista o contexto e o problema apresentado este estudo de caso tem o objetivo de validar se a abordagem apresentada no capítulo 4 obtém sucesso numa aplicação real, que foi descrita na seção anterior, sob o ponto de vista da modelagem de requisitos não-funcionais dentro da modelagem de negócios.

5.3 Estudo de caso

Nas subseções seguintes, serão apresentados e detalhados todos os passos para a utilização da abordagem BPMNFR descrita no capítulo 4 através do estudo de caso. Apesar dessas aplicação ser real, não caberá a este estudo mostrar todos as atividades de todos os processos desta aplicação, ou corre-se o risco de perder o foco desta dissertação.

5.3.1 Modelagem do processo de desenvolvimento do produto

Nesta primeira etapa, foi feito o diagrama de processos para o desenvolvimento da aplicação que foi entregue ao MEC. Este processo é modelado utilizando a notação de BPMN, mostrada no capítulo 2.

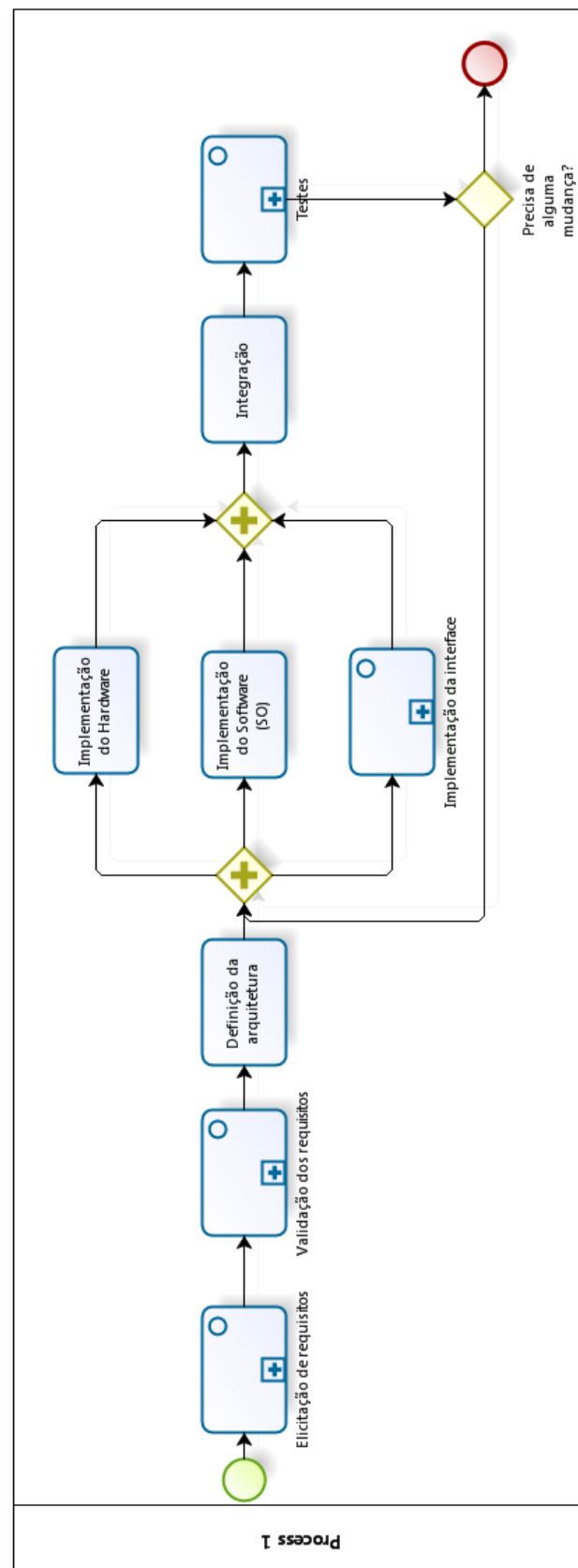
Os macro-processos identificados foram: elicitação de requisitos, validação dos requisitos, desenvolvimento dos componentes do sistema e Testes. Existem também atividades, tais como: Definição de arquitetura e Integração. Uma visão geral sobre os processos de desenvolvimento do lampejo (aplicação desenvolvida para o MEC) é apresentada na figura 5.1.

Para um melhor entendimento, os macro-processos de elicitação de requisitos, validação dos requisitos, implementação da interface e testes serão detalhados e mostrados a seguir. Uma visão detalhada desses macro-processos é importante para que se entenda todo o contexto e faça sentido nos próximos passos da validação da abordagem BPMNFR.

Elicitação de requisitos

Este macro-processo é o primeiro na lista de atividades no desenvolvimento da aplicação e é de suma importância para o bom entendimento das funcionalidades e das necessidades do usuário.

As atividades identificadas para esse macro-processo podem ser vistas na figura 5.2 e foram: análise de competidores, conversa com o *stakeholder* (no caso professores e pedagogos), uma análise qualitativa dos dados colhidos nas duas atividades anteriores,



a identificação das principais funcionalidades da aplicação usando como base a análise qualitativa e finalmente a elaboração da versão preliminar do documento de requisitos. O macro-processo de Elicitação de Requisitos gera como artefato o documento de requisitos.

Este processo é linear e não tem muitos pontos de decisão, pois ainda existirá a etapa de validação dos requisitos, apresentados no artefato que foi gerado. Com a figura 5.2 foi possível entender o detalhe das atividades do macro-processo de elicitação de requisitos e passar para o próximo macro-processo.

Validação de requisitos

Este macro-processo é de muita importância para o andamento do projeto, já que é nele que se tem a confirmação se o que foi levantado de requisitos foi suficiente e se atende às expectativas dos usuários.

Na validação de requisitos foi identificada a necessidade da presença dos usuários. Portanto, a tarefa de validar com usuários e elaborar documento de requisitos consolidado fazem parte deste macro-processo (vide figura 5.3).

No processo de validação, o documento de requisitos só precisará ser evoluído, se as funcionalidades identificadas anteriormente não forem aceitas pelo usuário. Se isso acontecer será gerado também um artefato, o documento de requisitos validado pelo usuário.

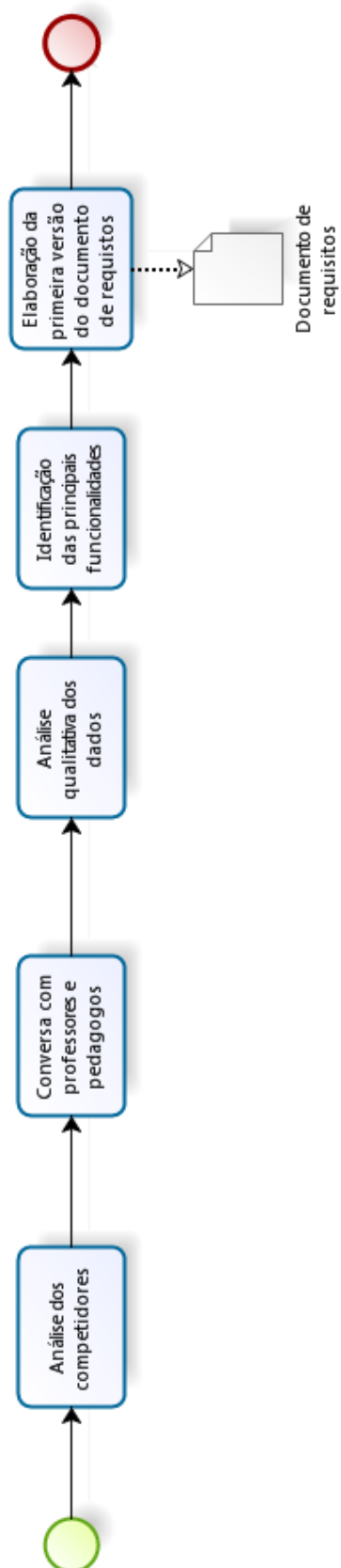
Desenvolvimento dos componentes do sistema

Como é mostrado na figura 5.1, os componentes do sistema são interface, *software* e *hardware*. Cada uma dessas três partes têm um subprocesso interno, porém para este trabalho só detalharemos o subprocesso de desenvolvimento da interface.

As atividades identificadas para o desenvolvimento da interface foram: levantamento da infra-estrutura de *hardware*, levantamento das necessidades específicas do usuário, desenho das telas, desenho dos ícones, aprovação do *layout* e desenvolvimento (vide figura 5.4).

Durante a etapa de desenvolvimento da interface, também existe a interação com o usuário final, através da atividade de aprovação do *layout*. O *layout* desenvolvido precisa ser validado por ele e se for necessária alguma modificação esse *layout* volta a ser trabalhado até que chegue num ponto que seja satisfatório para o usuário.

Desta forma se encerra a apresentação e detalhamento dos subprocessos mostrados na figura 5.1.



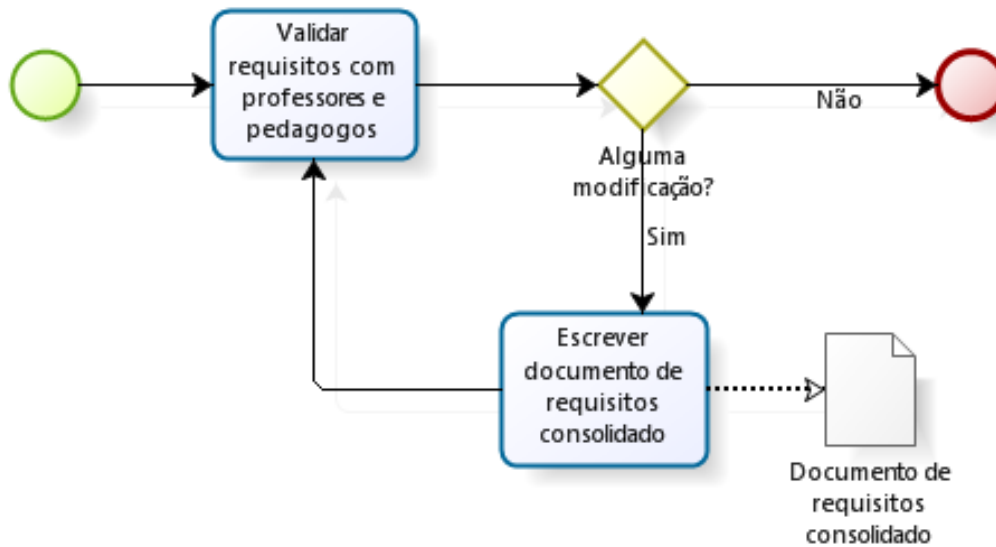


Figura 5.3 Subprocesso de validação de requisitos.

Testes

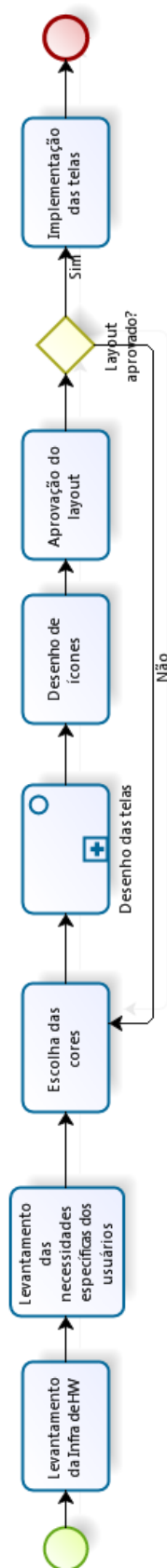
Este macro-processo é o último da cadeia de processos, ele é de suma importância pois nele são feitos os testes da aplicação como um todo e é a partir dele que é possível detectar os erros que ainda precisam ser consertados ou dar como encerrado o desenvolvimento.

Nos testes primeiro se faz o teste individual do *hardware* que foi construído, quando os erros forem encontrados, o processo é finalizado. Se não houver erros, a atividade de testar a aplicação é iniciada. Encontrando erros ou não a atividade é finalizada (vide figura 5.5).

Quando o processo é finalizado, existe um ponto decisão no macro-processo que encerra o desenvolvimento ou não, a dependência está relacionado à ter encontrado erros ou não no processo de testes.

Considerações

Durante a elaboração do macro-processo mostrado na figura 5.1 e de seus subprocessos, visto nas figuras 5.2, 5.3, 5.4 e 5.5, ainda não foi possível modelar com clareza os requisitos não-funcionais. O processo de desenvolvimento do produto foi modelado e a partir disto será dado início a próxima fase da abordagem BPMNFR, a fase de rotulação das atividades.



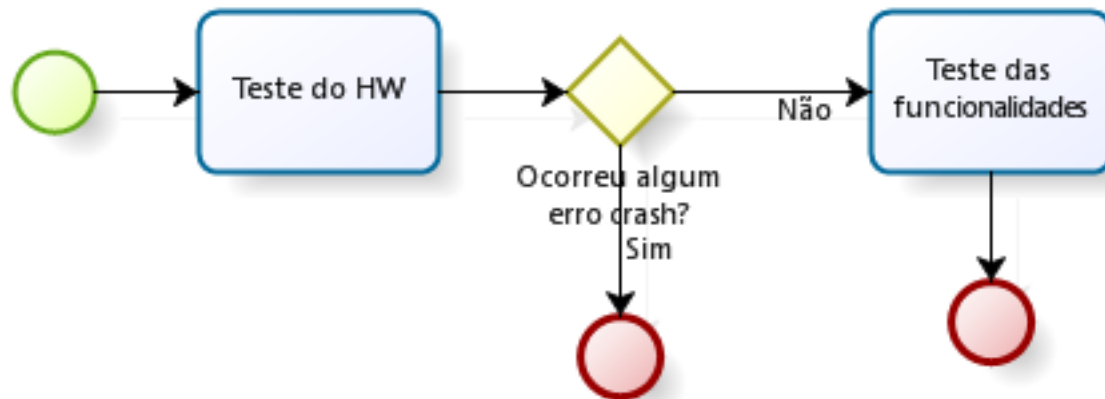


Figura 5.5 Subprocesso de Testes.

5.3.2 Inserção dos requisitos não-funcionais através dos rótulos

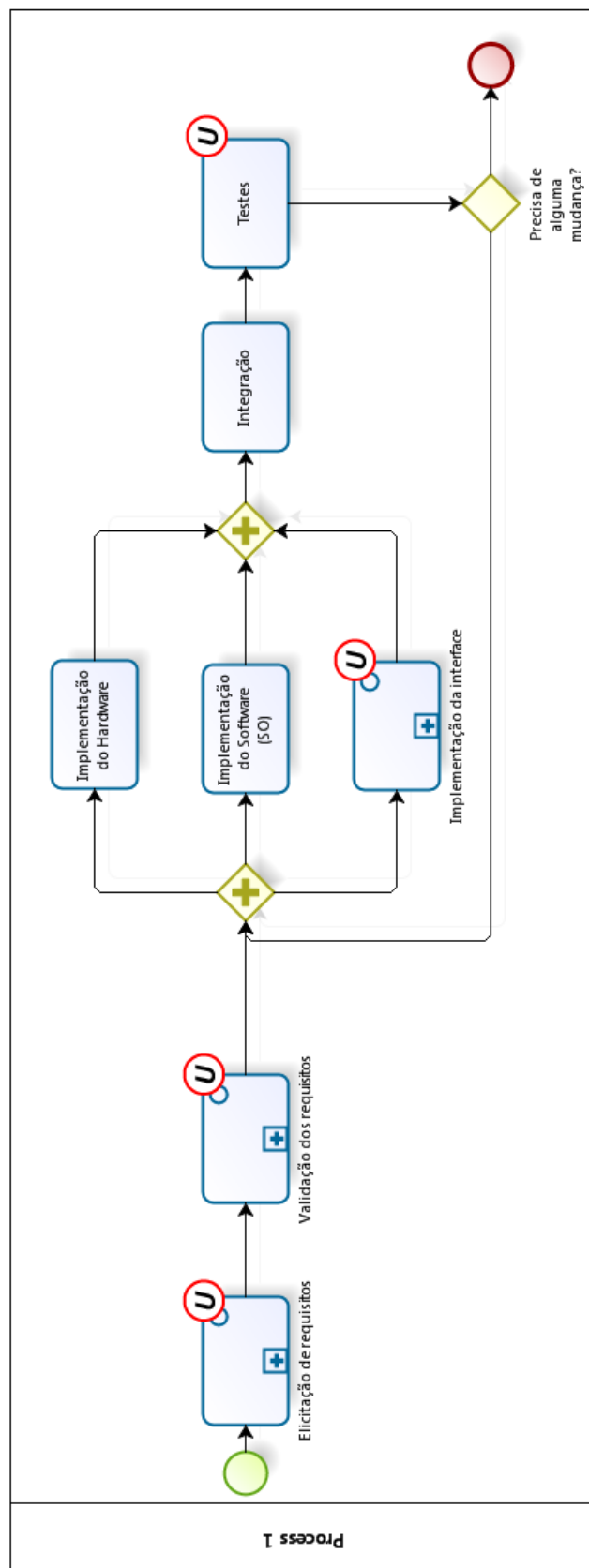
Os diagramas gerados na fase anterior servirão de artefato de entrada para esta nova fase. Durante a elicitação de requisitos os requisitos não-funcionais foram identificados e nesta fase serão associados de forma explícita às atividades dentro dos processos.

No projeto lampejo, os requisitos não-funcionais identificados foram: mobilidade, acessibilidade, usabilidade, auto-contido, robustez, facilidade no transporte, baixo custo de manutenção, baixo custo com infra-estrutura. Dentre esses requisitos não-funcionais, os mais críticos são usabilidade, mobilidade e acessibilidade. Neste trabalho, usaremos o requisito usabilidade para o estudo de caso.

Levando-se em consideração a usabilidade, as atividades que estão relacionadas são elicitação de requisitos, implementação da interface, validação dos requisitos e testes do produto, conforme figura 5.6.

Se, após a identificação das operacionalizações, o analista notar que ficou faltando rotular alguma tarefa, esta rotulação ainda poderá ser feita, levando-se em consideração que a abordagem BPMNFR propõe um processo incremental.

Depois de identificadas as atividades que são relacionadas aos determinados requisitos não-funcionais, isto é, as atividades que foram rotuladas, poderemos dar início a próxima etapa.



5.3.3 Operacionalizações dos requisitos não-funcionais

Após os requisitos não-funcionais serem identificados, é necessário que sejam descobertas as operacionalizações desses requisitos para que se transformem em atividades dentro do **BPD**. Essas operacionalizações podem ser descobertas analisando um catálogo baseado em *NFR-Framework* já existente, como os catálogos encontrados em (Chung *et al.*, 2000), ou estendendo um catálogo já existente para adaptar a um domínio específico, ou, se o catálogo não existir, definindo-o usando a notação do *NFR-Framework* de (Chung *et al.*, 2000).

No estudo de caso, o requisito não-funcional que está sendo trabalhado é usabilidade. Como mostrado no capítulo 4 foi criado um catálogo para usabilidade baseado nas dez heurísticas criadas por Nielsen em (Nielsen, 1994). O catálogo criado ainda foi estendido para atender às particularidades deste domínio e desta aplicação. O detalhe do refinamento do catálogo será mostrado na próxima seção.

Refinamento do Catálogo de Usabilidade

Utilizando o *NFR-Framework*, mostrado no capítulo 3, e as heurísticas de (Nielsen, 1994) chegou-se ao catálogo de usabilidade mostrado nas figuras 3.8, 3.9, 3.10, 3.11 que estão no capítulo 3. Este primeiro catálogo criado serviu como referência para a extensão do catálogo utilizado no estudo de caso, que tem as operacionalizações específicas para essa aplicação, neste domínio.

Com este catálogo genérico pronto, para o estudo de caso foi necessário detalhar um pouco mais as operacionalizações além de criar um novo *softgoal* para o entendimento específico do domínio no usuário. A figura 5.7 mostra o refinamento da sub-árvore de Fácil de aprender e usar. Esta foi a parte do catálogo que foi refinada.

Na figura 5.7 a operacionalização “Seguir convenções da plataforma” foi mais detalhada e derivou uma nova operacionalização chamada “Cores adequadas ao projetor”. Outra operacionalização derivada foi “Instruções visíveis” que agora tem mais um detalhamento com a operacionalização “Ícones grandes”. Finalmente a última operacionalização que foi expandida foi a “Prototipação” que recebeu quatro operacionalizações que contribuem para seu cumprimento, são elas: “Modelar navegação”, “Desenhar esboço de telas”, “Preparar protótipo” e “Aplicar protótipo”.

O novo *softgoal* criado serviu para o entendimento nas necessidades específicas do domínio do usuário. O *softgoal* “Entender o domínio do usuário” já aparece com três operacionalizações (vide figura 5.8): “Observações em sala”, “Sessão de tarefas” e

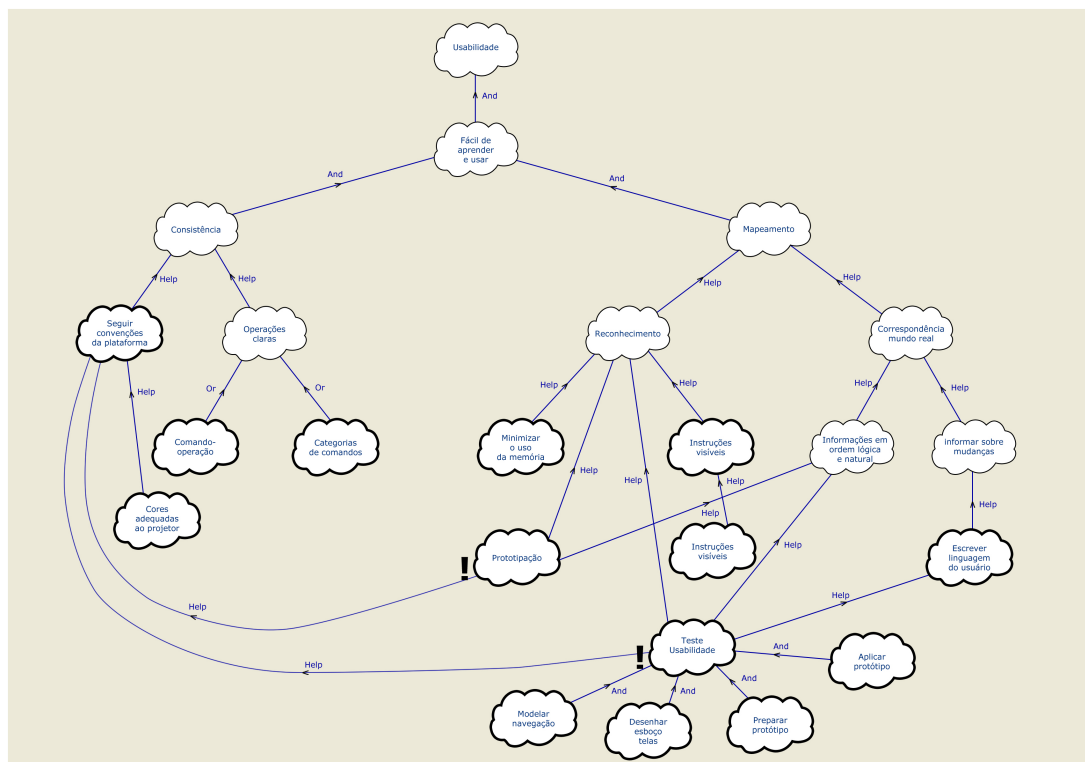


Figura 5.7 Catálogo de usabilidade refinado - Fácil de aprender e usar.

“Entrevistas com pedagogos”.

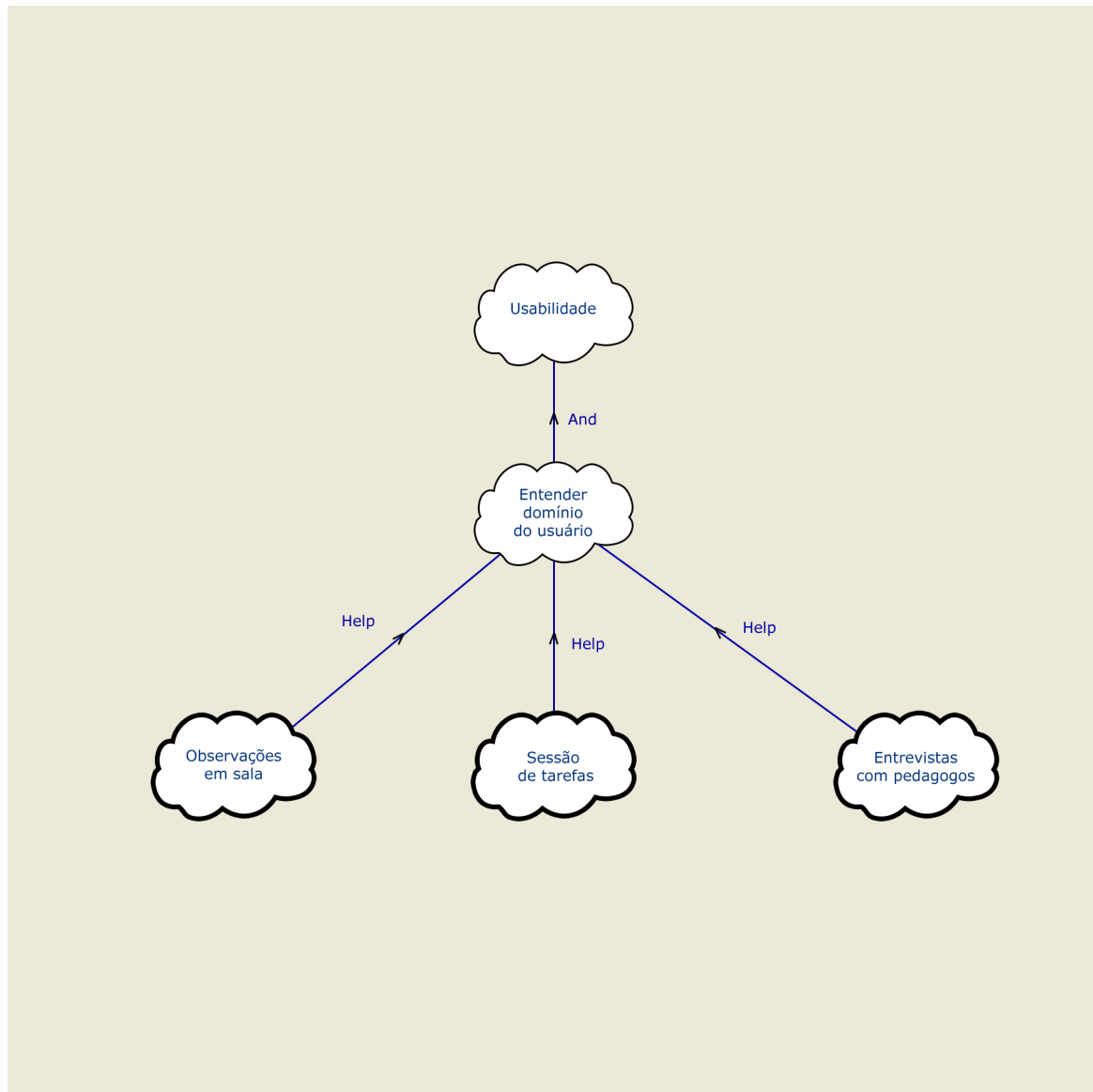


Figura 5.8 Catálogo de usabilidade novo *softgoal* - Entender domínio do usuário.

Com o catálogo baseado no *NFR-Framework* devidamente adaptado ao domínio da aplicação podemos identificar quais operacionalizações farão parte do **BPD**.

Identificação das operacionalizações

Depois de ter o catálogo do requisito não-funcional com todas as operacionalizações possíveis, cabe ao analista de negócios identificar quais dessas operacionalizações deverão

fazer parte do **BPD**. Essa escolha deverá ser norteada pelo domínio da aplicação, pois depende dele a existência ou não de uma operacionalização no **BPD**.

No estudo de caso, as operacionalizações que fazem sentido para o domínio da aplicação são “Desenhar ícones grandes”, “Escolher cores que se adequem ao uso do projetor”, “Observações em sala de aula”, “Sessões de tarefas com os alunos”, “Entrevistas com pedagogos e professores”, “Modelar navegação das telas”, “Desenhar esboço das telas”, “Preparar a aplicação do protótipo”, “Aplicar protótipo com os usuários”, “Teste de usabilidade”.

Inserção das operacionalizações no BPD

Com a escolha das operacionalizações do requisito não-funcional que irão fazer parte do **BPD** já feita, finalmente pode-se inseri-las a partir do ponto de marcação feito no começo, os rótulos.

Neste estudo de caso, as atividades que tinham operacionalizações associadas foram elicitação de requisitos, implementação da interface e testes do produto, apresentada na figura 4.4.

Em cada atividade as operacionalizações foram associadas e identificadas através do grupo lógico. Cada atividade, que teve os requisitos não-funcionais associados, serão mostradas nas figuras seguintes.

Na figura 5.9, é inserido um processo a mais, o de prototipação, referente à operacionalização prototipação que aparece no catálogo de usabilidade refinado. Esta operacionalização está ligado com a elicitação de requisitos, pois era esta atividade que antes estava rotulada.

Na figura 5.10, são inseridas as operacionalizações ‘Observações em sala de aula’, “Sessões de tarefas com os alunos” e “Entrevistas com pedagogos e professores”. Essas operacionalizações estão relacionados ao novo *softgoal* que precisou ser criado no refinamento do catálogo de usabilidade. Estas operacionalizações foram inseridas dentro do macro-processo de elicitação de requisitos. A atividade de conversa com professores foi substituída pela entrevista.

Na figura 5.11, relacionada à validação de requisitos, são criadas atividades que representam o detalhamento da operacionalização prototipação. Essas atividades são “Modelar navegação das telas”, “Desenhar esboço das telas”, “Preparar a aplicação do protótipo”, “Aplicar protótipo com os usuários”.

Na figura 5.12, é inserida uma atividade a mais que só existe por conta da operacionalização prototipação, essa atividade é “Colher resultados da prototipação”.

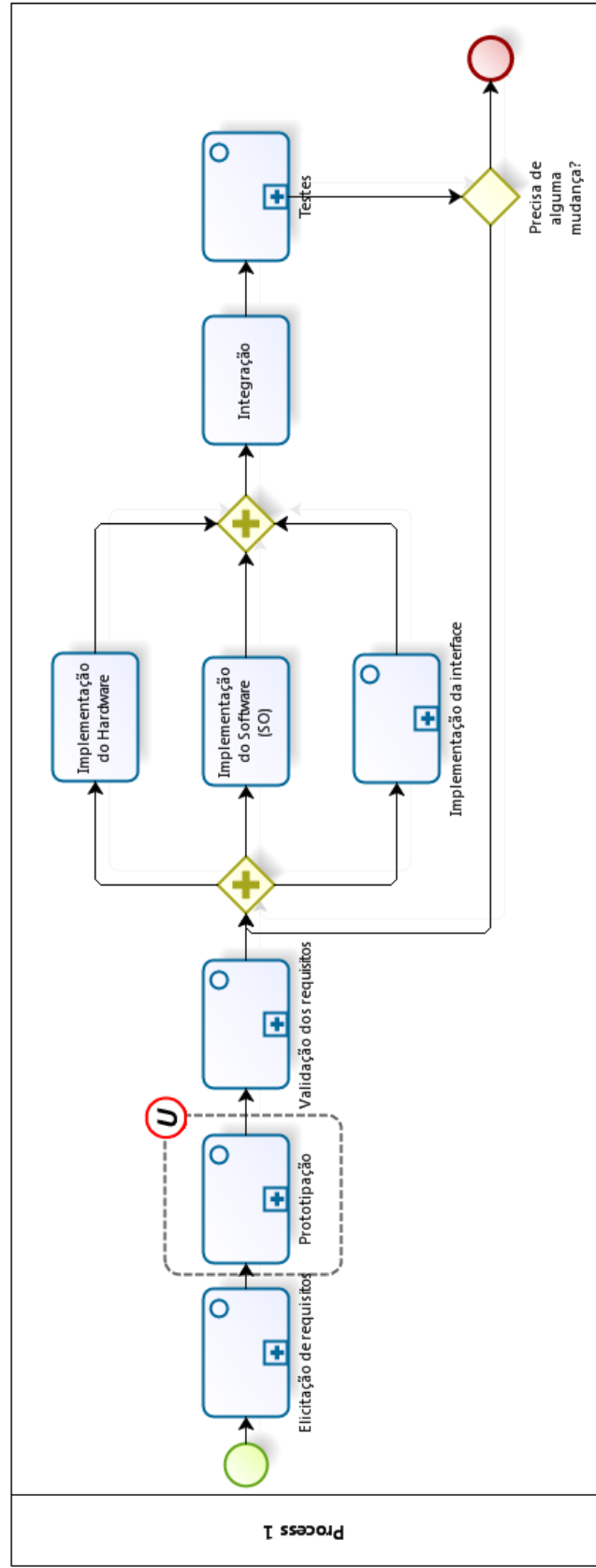
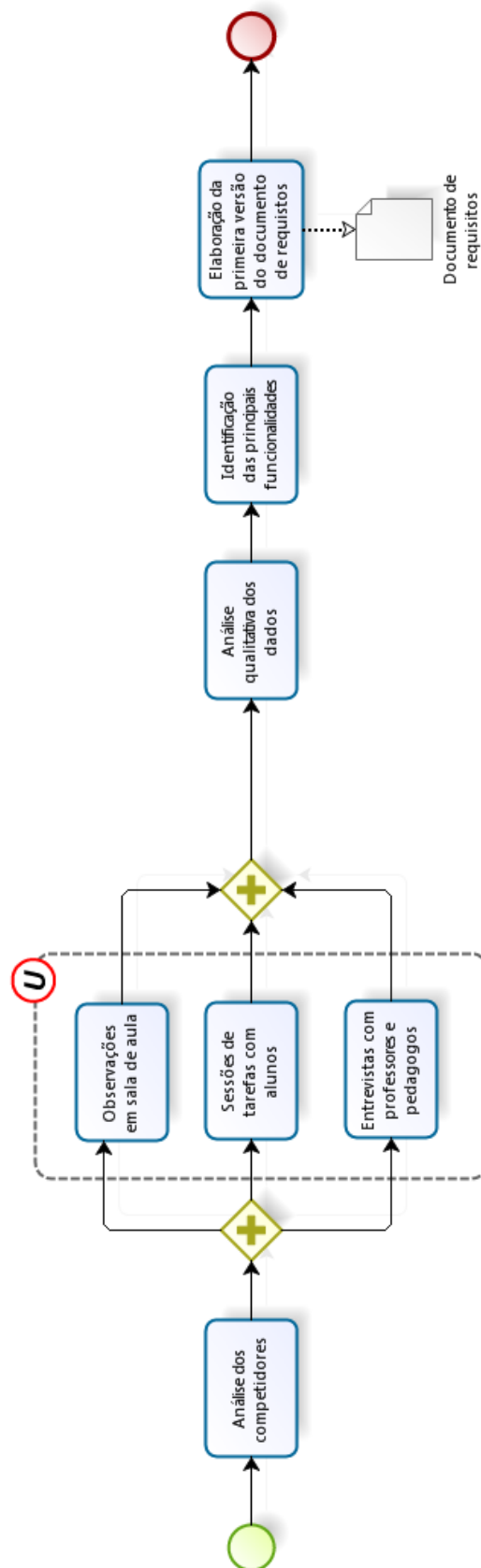
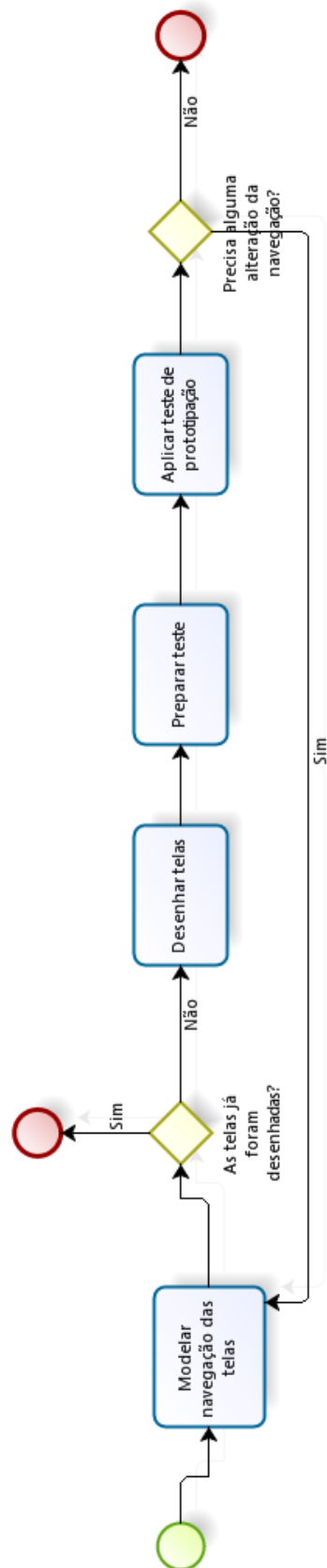
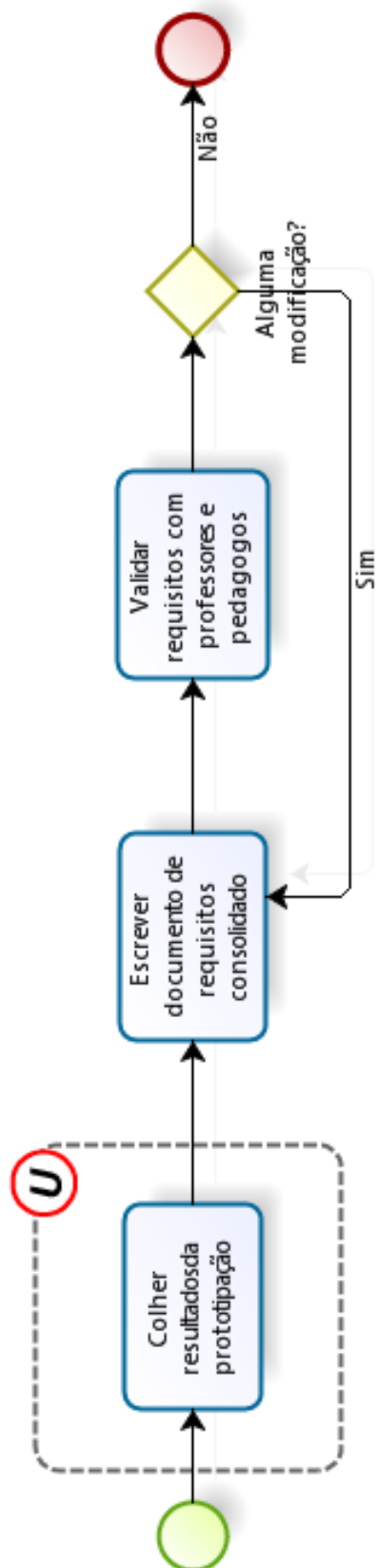


Figura 5.9 Operacionalizações dos requisitos não-funcionais no **BPD** do lampejo]. 75



76
Figura 5.10 Operacionalizações dos requisitos não-funcionais no **BPD** de elicitação de requisitos.

**Figura 5.11** Detalhamento da operacionalização “prototipação”.



78
Figura 5.12 Operacionalizações dos requisitos não-funcionais no **BPD** de validação de requisitos.

O sub-processo de implementação da interface descrito na figura 5.4 também é revista na figura 5.13, onde é criada atividades que representam o detalhamento da operacionalização do desenvolvimento da interface. Essa atividade é “Escolher cores que se adequem ao uso do projetor”. Além dessa atividade, também é criada uma nova atividade que surgiu a partir da operacionalização de “prototipação”, que é a “Criação das telas baseadas na prototipação”. Outro ponto importante é que desenhar ícones deixou de ser apenas uma atividade para representar um processo que será detalhado na figura 5.14.

Na figura 5.14, são criadas atividades que representam as operacionalização no desenho dos ícones. Essas atividades são “Escolha de imagens que se assemelhem com o mundo real” e “Desenho de ícones grandes”.

Na figura 5.15, é criada a atividade “Teste de usabilidade” dentro do processo de testes que representa uma das operacionalizações escolhidas no catálogo.

Para facilitar a análise, a tabela 5.1 foi criada e com ela é possível fazer a correspondência entre as tarefas rotuladas do BPD e as operacionalizações que foram criadas a partir do requisito não-funcional usabilidade.

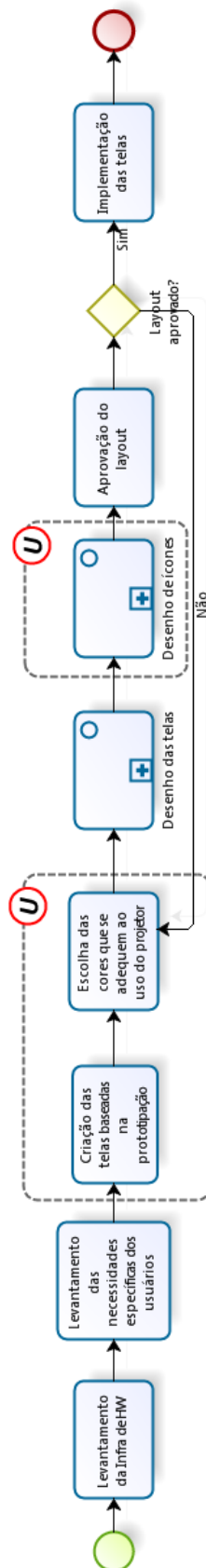
Tarefa rotulada	RNF associado	Operacionalizações	Tipo
Elicitação de Requisitos	Usabilidade	Observações em sala de aula	S / T
		Sessão de tarefas com alunos	S / T
		Entrevistas com Professores	S / T
		Prototipação	A / MP
Validação de Requisitos	Usabilidade	Colher resultados da prototipação	S / T
Implementação da Interface	Usabilidade	Criação das telas baseadas na prototipação	S / T
		Escolha das cores que se adequem ao projetor	S / T
		Desenho dos ícones	S / MP
Testes	Usabilidade	Teste de Usabilidade	S / MP

Tabela 5.1 Correspondências de operacionalizações

Todas as tarefas que foram rotuladas, tinham o mesmo requisito não-funcional associado, Usabilidade.

A elicitación de requisitos derivou cinco operacionalizações da usabilidade que foram associadas posteriormente, são elas: Observações em sala de aula, sessão de tarefas com alunos e entrevistas com professores que foram alocadas dentro do sub-processo de elicitación e foram inseridas como tarefas, assim como Prototipação que é uma atividade que está associada ao requisito de elicitación por meio de um fluxo associação e é um macro-processo.

A validação de requisitos teve apenas uma operacionalização associada através de uma atividade dentro de seu sub-processo, essa operacionalizações foi “Colher resultados da prototipação”.



80
Figura 5.13 Operacionalizações dos requisitos não-funcionais no BPD de desenvolvimento da interface.

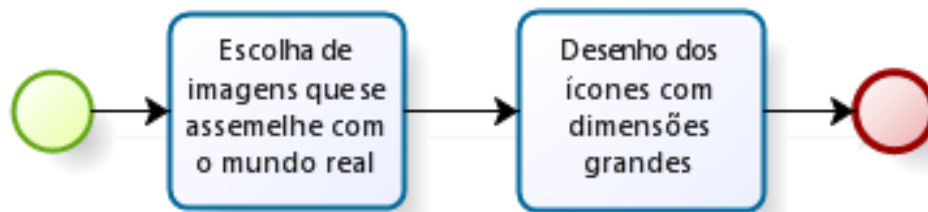


Figura 5.14 Operacionalizações do desenho dos ícones.

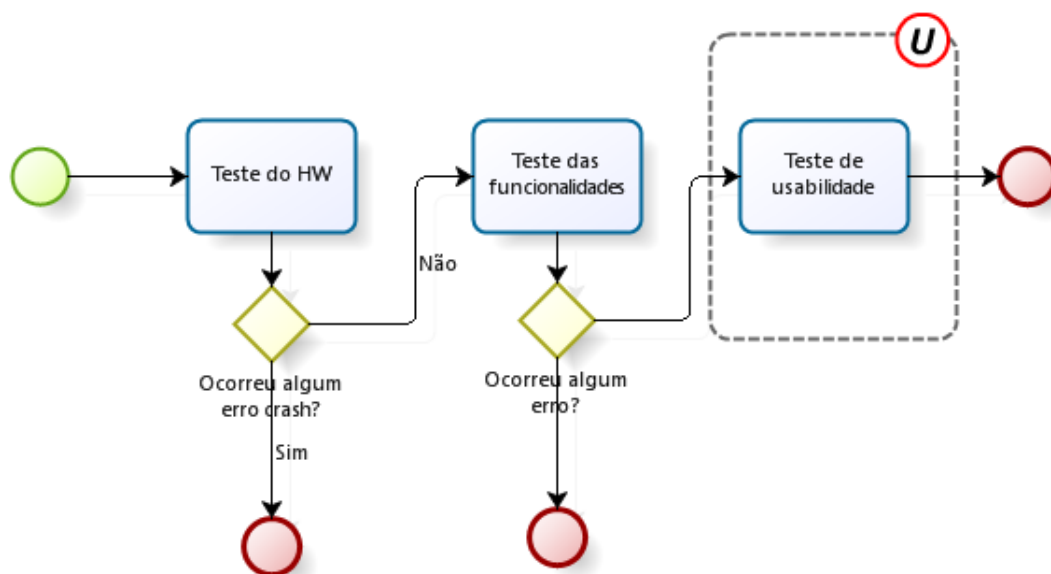


Figura 5.15 Operacionalização dos testes.

Na implementação da interface três operacionalizações foram associadas, duas (Criação de telas baseadas na prototipação e Escolha das cores que se adequem ao projetor) dentro do seu subprocesso em forma de atividades. A última operacionalização, Desenho de ícone, foi colocada dentro do sub-processo de implementação de interface, na forma de um macro-processo.

E finalmente na tarefa de Teste, foi inserida a operacionalização “Teste de Usabilidade” na forma de macro-processo e localizado dentro do sub-processo de Teste.

Desta forma, os requisitos não-funcionais foram colocados num diagrama de modelagem de processos de negócio de forma explícita.

5.4 Considerações finais

Este Capítulo apresentou a utilização da abordagem BPMNFR na modelagem do desenvolvimento de uma aplicação real. Os resultados obtidos nesta utilização foram animadores pois atingiram os objetivos deste trabalho, que foi apresentar uma pesquisa baseada na análise de requisitos não-funcionais e na modelagem de processos de negócio, onde foi proposta a abordagem BPMNFR para integrar os requisitos não-funcionais à notação de **BPMN**, como forma de solucionar sua deficiência de expressividade desses requisitos. Assim, foi proposta a criação de novos símbolos para que esta notação pudesse contemplar os requisitos não-funcionais.

Evidentemente, este foi um caso e precisaríamos aplicar a abordagem BPMNFR em outros domínios e com outros requisitos não-funcionais para que tivéssemos posições mais sólidas acerca das suas contribuições e limitações, entretanto foi possível, com o estudo de caso, fazer a crítica da abordagem BPMNFR e pensar em aspectos que contribuirão para a evolução posterior do modelo.

*“I ought to be testing myself daily,
and asking myself what I am really achieving.”*

Margot Asquith



Conclusão

Neste capítulo serão apresentadas as conclusões obtidas no desenvolvimento desta dissertação, bem como descritos alguns trabalhos futuros que poderão ser desenvolvidos.

6.1 Conclusões

A Engenharia de Requisitos ainda é uma fase do desenvolvimento de *software* onde erros e enganos são comuns. Portanto, pode ser a fonte de problemas nas próximas fases do desenvolvimento, podendo assim, não satisfazer as reais necessidades da organização onde o *software* for implantado. Alguns dos erros detectados na prática são a falta de compreensão do negócio da empresa pelo analista de sistemas, a falta de foco no escopo do *software* e falta de comunicação entre os *stakeholders* e o analista de sistema.

Este trabalho descreveu uma abordagem para tentar evitar esses tipos de problemas baseando-se na modelagem dos processos de negócio por meio de **BPMN** e dos requisitos não-funcionais descritos utilizando o *NFR-Framework*. A abordagem BPMNFR possibilitará que analistas de sistema compreendam e analisem corretamente a organização, suas necessidades e os atributos de qualidade que a ela são importantes.

Analistas de negócio e analistas de sistema compartilham uma linguagem que seja compreensível a ambos graças ao **BPMN**. **BPDs** são a base para os usuários, para validar se a estrutura organizacional e de comportamento têm sido adequadamente entendida pelo analista de sistemas. Além disso, a abordagem BPMNFR tenta atenuar as deficiências da utilização apenas do **BPMN** sem a utilização de nenhuma abordagem para detalhar requisitos não-funcionais. Sendo assim propomos a utilização do *NFR-Framework* para modelar e facilitar a descoberta das operacionalizações de requisitos não-funcionais.

A dissertação foi dividida da seguinte forma: os primeiros capítulos mostraram a base do conhecimento necessário para o entendimento da abordagem BPMNFR proposta e os

capítulos seguintes apresentaram o processo proposto e um estudo de caso.

No capítulo 2, Modelagem de Processos de Negócio, foi apresentado um resumo sobre modelos de negócios e sua aplicação em desenvolvimento de sistemas, a modelagem de processos de negócio e algumas abordagens existentes; além de uma introdução sobre a notação BPMN.

No capítulo 3, Requisitos Não-Funcionais, inicialmente, foi feita uma introdução sobre os requisitos não-funcionais, e algumas definições básicas. Posteriormente, foi apresentado o *NFR-Framework* e sua utilização. Depois foi mostrada de forma mais detalhada a obtenção de operacionalização no *NFR-Framework*, além de um catálogo de usabilidade.

No capítulo 4, a Abordagem BPMNFR foi apresentada e foi detalhado o processo proposto para integrar os requisitos não-funcionais à modelagem de processos de negócio, bem como os passos para utilização do mesmo.

No capítulo 5, Estudo de caso, apresentou-se a execução de um estudo de caso. Foi feita a modelagem de uma aplicação real para testar a efetividade do uso da abordagem BPMNFR criada.

O resultado da pesquisa desta dissertação foi um processo que viabiliza a inserção das operacionalizações dos requisitos não-funcionais nos diagramas de processos de negócio. Tratou da natureza subjetiva dos requisitos não-funcionais e contribui com mais uma evidência do uso da modelagem de processos de negócio como um passo inicial do processo de engenharia de requisitos.

Uma extensão de BPMN para a modelagem de requisitos não-funcionais foi o principal resultado desta dissertação. A viabilidade de aplicação da abordagem BPMNFR proposta foi verificada através do estudo de caso, que demonstrou a possibilidade de utilização da mesma.

O estudo e a apresentação dos conceitos e aspectos básicos relacionados aos requisitos não-funcionais e à notação para modelagem de processos de negócio foi uma contribuição secundária deste trabalho.

A apresentação da importância da modelagem dos processos de negócios na fase de engenharia de requisitos para o desenvolvimento de uma aplicação de *software* também foi uma contribuição importante deste trabalho. Além da criação e detalhamento de um catálogo do requisito não-funcional usabilidade.

6.2 Trabalhos futuros

Para validar de forma mais definitiva a abordagem proposta, se faz necessária a utilização dela tanto no mesmo domínio com outros requisitos não-funcionais, como em outros domínios e com outros requisitos não-funcionais. Poderemos explorar os requisitos que já têm algum catálogo no *NFR-Framework*, como outros que ainda não foram catalogados.

A publicação do resultado desta dissertação também é uma das metas futuras.

Além da utilização em outras aplicações, domínios e requisitos não-funcionais, também é importante, para a evolução deste trabalho, que uma ferramenta seja implementada para suportar a nova notação que estende o *BPMN*.

Avaliação do processo e da notação em outros estudos de caso. Para a validação da abordagem BPMNFR proposta, a utilização do processo e da nova notação em outros estudos de caso é muito importante. Os estudos de caso primeiramente deverão abranger o mesmo domínio e outros requisitos não-funcionais, tais como, acessibilidade, robustez, segurança. Depois de validado no mesmo domínio será necessária a utilização do processo em estudos de caso de outros domínios com os mesmos requisitos não-funcionais anteriormente testados, assim como com outros requisitos. Esses estudos serão importantes para a avaliação da abordagem BPMNFR e de sua eficácia.

Desenvolvimento de catálogos de requisitos não-funcionais. A criação dos catálogos de requisitos não-funcionais que ainda não foram catalogados contribui para a utilização da abordagem BPMNFR proposta, tendo em vista que os catálogos são parte crucial para a utilização da abordagem BPMNFRs. É necessário que sejam desenvolvidos mais catálogos de requisitos não-funcionais para que se potencialize a aceitação e uso da abordagem proposta.

Desenvolvimento de ferramentas de apoio. Uma nova notação que estende o *BPMN* foi criada e, portanto, é necessária a criação de ferramentas que dêem apoio a utilização da mesma. Essas ferramentas devem prever a utilização da notação graficamente, assim como, semanticamente.

Aprimoramento do catálogo de usabilidade. O catálogo de usabilidade criado nesta dissertação foi um passo inicial para elaboração de um catálogo de usabilidade mais completo, ele utilizou conceitos de três autores, portanto é possível o aprimoramento, principalmente através da utilização da norma ISO/IEC 9126-1:2001 para qualidade de produto de software.

Referências Bibliográficas

- Berio, G. and Vernadat, F. (2001). Enterprise modelling with cimos: functional and organizational aspects. *Production Planning & Control: The Management of Operations*, 2001, **Vol. 12**, 128–136.
- Bittencourt, R. M. V. (2009). *Identificando Expectativas de Qualidade de SIs com o Apoio de Modelos de Negócio*. Master's thesis, Universidade Federal do Estado do Rio de Janeiro - Centro de Ciências Exatas e Tecnologia.
- Boehm, B. W. (1988). A spiral model of software development and enhancement. *A spiral model of software development and enhancement*, pages 61–72.
- Booch, G., Rumbaugh, J., and Jacobson, I. (1998). *The Unified Modeling Language User Guide*. Addison Wesley.
- BPML.org (2008). *Business Process Modeling Notation, OMG Available Specification*. Object Management Group, 1.1 edition.
- Brooks, F. P. (1987). No silver bullet: Essence and accidents of software engineering. *IEEE computer*, **Vol. 20**, 10–19.
- Castro, J., Kolp, M., and Mylopoulos, J. (2002). Towards requirements-driven information systems engineering: The tropos project. *Information Systems*, **Vol. 27**, 365–389.
- Chung, L. and Nixon, B. A. (1995). Dealing with non-functional requirements: three experimental studies of a process-oriented approach. In *ICSE '95: Proceedings of the 17th international conference on Software engineering*, pages 25–37, New York, NY, USA. ACM.
- Chung, L., Nixon, B. A., Yu, E., and Mylopoulos, J. (2000). *Non-Functional Requirements in Software Engineering*. Springer.
- Cysneiros, L. M. (2001). *Requisitos Não Funcionais: Da Elicitação ao Modelo Conceptual*. Ph.D. thesis, PUC/RJ.
- Cysneiros, L. M. (2007). Evaluating the effectiveness of using catalogues to elicit non-functional requirements. In *10th Workshop in Requirements Engineering*.
- Cysneiros, L. M. and do Prado Leite, J. C. S. (1998). Utilizando requisitos não funcionais para análise de modelos orientados a dados. *Workshop de Engenharia de Requisitos-XII SBES*.

REFERÊNCIAS BIBLIOGRÁFICAS

- Davenport, T. H. (1993). *Process Innovation: Reengineering work through information technology*. Harvard Business Press.
- Davenport, T. H. (1994). *Reengenharia de processos: como inovar na empresa através da tecnologia da informacao*. Editora Campus.
- Davenport, T. H. and Stoddard, D. B. (1994). Reengineering: Business change of mythic proportions? *MIS Quarterly*, **Vol. 18**(2), 121–127.
- Davis, A. M. (1993). *Software requirements: objects, functions, and states*. Prentice-Hall, Inc.
- de La Vara, J. L., Sánchez, J., and Óscar Pastor (2008). *Business Process Modelling and Purpose Analysis for Requirements Analysis of Information Systems*, pages 213–227. Springer-Verlag, Berlin, Heidelberg.
- Dubray, J.-J. (2004). Business process modeling notation. *ebPML.org*.
- Dumas, M., van der Aalst, W., and Hofstede, A. T. (2005). *Process-aware information systems*. John Wiley and Sons.
- Eriksson, H.-E. and Penker, M. (2000). *Business Modeling With UML: Business Patterns at Work*. John Wiley & Sons.
- Estrada, H., Rebollar, A. M., Pastor, O., and Mylopoulos, J. (2006). *Advanced Information Systems Engineering*, chapter An Empirical Evaluation of the i* Framework in a Model-Based Software Generation Environment, pages 513 – 527. Springer.
- Goldsby, H. J., Konrad, S., and Cheng, B. H. (2007). Goal-oriented patterns for uml-based modeling of embedded systems requirements. In *Proceedings of the 10th IEEE High Assurance Systems Engineering Symposium*, Dallas, TX, USA.
- Gomes, A. S., de Araújo, C. R., de Arruda, D. A., Sultanum, N. B., de Paiva, L. J., and de Albuquerque Menezes, L. (2006). Participação de professores de matemática no desenvolvimento de aplicações educativas computacionais para o ensino de estruturas aditivas. In *VI EPEM- Encontro Pernambucano de Educação Matemática*, volume Vol. 1, pages 1–13, Caruaru.
- Group, T. S. (1995). Chaos reports. Technical report, The Standish Group.

- Harmon, P. and Wolf, C. (2008). The state of business process management. Technical report, Business Process Trends.
- Harrington, H. J. (1991). *Business process improvement*. McGraw-Hill Professional.
- Holtzblatt, K. and Beyer, H. R. (1995). Requirements gathering: the human factor. *Commun. ACM*, **Vol. 38**(5), 31–32.
- Johansson, H. J. and Pendlebury, A. (1993). *Business Process Reengineering: BreakPoint Strategies for Market Dominance*. John Wiley & Sons.
- Jones, S. and Maiden, N. (2004). *Requirements Engineering for Sociotechnical Systems*, chapter RESCUE: An Integrated Method for Specifying Requirements for Complex Socio- Technical Systems. Information Science Publishing.
- Kavakli, E. and Loucopoulos, P. (2003). Goal driven requirements engineering: Evaluation of current methods. In *Eighth International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design*, Klagenfurt/ Velden, Austria.
- Kirikova, M. and Bubenko, J. A. (1994). Enterprise modelling: Improving the quality of requirements specifications. In *Information systems Research seminar In Scandinavia*, Finland.
- Kirner, T. G. and Davis, A. M. (1996). Nonfunctional requirements of real-time systems. *Advances in Computers*, **Vol. 42**, 1–37.
- Knight, D. (2004). *Elicitação de Requisitos de Software a partir do Modelo de Negócio*. Master's thesis, UFRJ/NCE.
- Luftman, J. N., Papp, R., and Brier, T. (1999). Enablers and inhibitors of business-it alignment. *Communications of AIS*, page p. 1.
- Muehlen, M. Z. and Ho, D. T. (2008). Service process innovation: A case study of bpmn in practice. *41th Annual Hawaii International Conference on System Sciences*, **Vol. 0**, 372.
- Muehlen, M. Z. and Recker, J. (2008). How much language is enough? theoretical and practical use of the business process modeling notation. In *Advanced Information Systems Engineering*, volume Volume 5074/2008, pages 465–479. Springer Berlin / Heidelberg.
-

REFERÊNCIAS BIBLIOGRÁFICAS

- Mylopoulos, J., Chung, L., and Nixon, B. (1992). Representing and using non-functional requirements: A process oriented approach. *IEEE Transactions on Software Engineering*, **vol. 18**(n. 6), 483 – 497.
- Mylopoulos, J., Chung, L., Liao, S. S., Wang, H., and Yu, E. (2001). Exploring alternatives during requirements analysis. In *IEEE Software*, volume Vol. 18, pages 92–96.
- Nielsen, J. (1994). *Usability Inspection Methods*, chapter Heuristic evaluation, pages 413–414. ACM, New York, NY, USA.
- Norman, D. A. (1988). *The Design of Everyday Things*. Basic Civitas Books, New York.
- of Business Analysis, I. I. (2009). *A Guide to the Business Analysis Body of Knowledge*.
- Owen, M. and Raj, J. (2003). Bpmn and business process management - introduction to the new business process modeling standard. *Popkin Software*.
- Paim, R. (2007). *As Tarefas para Gestão de Processos*. Ph.D. thesis, Universidade Federal do Rio de Janeiro.
- Pavlovski, C. J. and Zou, J. (2008). Non-functional requirements in business process modeling. In *APCCM '08: Proceedings of the fifth on Asia-Pacific conference on conceptual modelling*, pages 103–112, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.
- Preece, J., Rogers, Y., and Sharp, H. (2005). *Design de Interação: Além da Interação Homem-Computador*. Bookman.
- Recker, J. (2008). Bpmn modeling who, where, how and why. *BPTrends*, **5**, 1–8.
- Recker, J., Indulska, M., Rosemann, M., and Green, P. (2006). How good is bpmn really? insights from theory and practice. In *Proceedings 14th European Conference on Information Systems*, pages 1582–1593, Goeteborg, Sweden.
- Reich, B. H. and Benbasat, I. (2000). Factors that influence the social dimension of alignment between business and information technology. *MIS quarterly*, **Vol. 24**(No. 1), pp. 81–113.
- Rolland, C. (2007). *Conceptual Modelling in Information Systems Engineering*, chapter Capturing System Intentionality with Maps, pages 141–158. Springer Berlin Heidelberg.

- Rummler, G. A. and Brache, A. P. (1995). *Improving Performance: How to manage the white space on the organizational chart*. Jossey-Bass Publishers, San Francisco, CA.
- Santander, V. F. and Castro, J. F. B. (2000). Desenvolvendo use cases a partir de modelagem organizacional. In *III Workshop of Requirement Engineering*, pages pp 158–180, Rio de Janeiro, Brasil.
- Sharp, A. and McDermott, P. (2009). *Workflow Modeling: Tools for Process Improvement and Application Development*. Artech House.
- Sommerville, I. and Sawyer, P. (1997). *Requirements engineering: a good practice guide*.
- Sousa, G. M. C. (2004). *Uma Abordagem Direcionada a Casos de Uso para o Desenvolvimento de Software Orientado a Aspectos*. Master’s thesis, Universidade Federal de Pernambuco.
- Taylor-Cummings, A. (1998). Bridging the user-is gap: a study of major information systems projects. *Journal of Information Technology*, **Vol. 13**, 29–54.
- Wagner, S., Deissenboeck, F., and Winter, S. (2008). Managing quality requirements using activity-based quality models. In *WoSQ ’08: Proceedings of the 6th international workshop on Software quality*, pages 29–34, New York, NY, USA. ACM.
- Wahl, T. and Sindre, G. (2005). *Advanced Topics in Database Research*, chapter Capítulo 6 - An Analytical Evaluation of BPMN Using a Semiotic Quality Framework, pages 94 – 105. Idea Group Inc.
- Weiß, D., Leukel, J., and Kirn, S. (2008). A method for aligning business process modeling and software requirements engineering. In *Process Innovation with Business Software*.
- White, S. A. (2006). Introduction to bpmn. *IBM Corporation*.
- Wiley, D. A. (2002). *The Instructional Use of Learning Objects*. Association for Instructional Technology.
- Wohlin, C., Höst, M., Runeson, P., and Ohlsson, M. (2000). *Experimentation in software engineering*, volume Vol. 6. Springer.
- Yu, E. (1995). *Modeling Strategic Relationships for Process Reengineering*. Ph.D. thesis, Department of Computer Science, University of Toronto.