DENISE LAZZERI GASTALDO BOMBONATTI

PARNAFOA: UM PROCESSO DE ANÁLISE DE REQUISITOS
NÃO-FUNCIONAIS ORIENTADO A ASPECTOS

São Paulo
2010

DENISE LAZZERI GASTALDO BOMBONATTI

PARNAFOA: UM PROCESSO DE ANÁLISE DE REQUISITOS
NÃO-FUNCIONAIS ORIENTADO A ASPECTOS

Tese apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de
Doutor em Engenharia.

Área de Concentração:
Engenharia Elétrica

Orientadora:
Prof^a Dr^a Selma Shin Shimizu
Melnikoff

São Paulo
2010

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de setembro de 2010.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Bombonatti, Denise Lazzeri Gastaldo

**PARNAFOA: um processo de análise de requisitos não-funcionais orientado a aspectos / D.L.G. Bombonatti. -- ed.rev. -- São Paulo, 2010.
p. 273**

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Metodologia e técnicas de computação 2. Engenharia de software 3. Processo de software 4. Processos fundamentais de desenvolvimento de software I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II. t.

AGRADECIMENTOS

À minha orientadora Prof^a Dr^a Selma Shin Shimizu Melnikoff pelas diretrizes, confiança e permanente apoio, que foram essenciais para a conclusão deste trabalho.

Ao professor Dr^o Kechi Hiram e professor Dr^o Paulo Sergio Muniz Silva pela grande contribuição oferecida no exame de qualificação, cujos comentários foram fundamentais para a conclusão desta tese.

Ao pesquisador Sam Supakkul pelos preciosos esclarecimentos e suporte no uso da ferramenta de modelagem utilizada neste trabalho.

Ao meu marido Carlos Eduardo Acetto Bombonatti pela confiança, compreensão, paciência e incentivo durante as decisões mais difíceis deste percurso.

Aos meus pais Dulce Lazzeri Gastaldo e Valter Gastaldo que me apoiaram e me deram força para persistir e superar todos os desafios.

A todos aqueles que de alguma forma, direta ou indireta, contribuíram com a evolução desta tese, e colaboraram para a conclusão deste trabalho.

RESUMO

Esta tese tem o objetivo de definir um processo para análise de requisitos não-funcionais orientado a aspectos denominado PARNAFOA. Este processo utiliza, de maneira integrada, métodos de tratamento de requisitos não-funcionais, baseados no NFR *Framework*, e métodos orientados a aspectos. Como resultado principal obtém-se um modelo de casos de uso que incorpora novas funções relacionadas aos requisitos não-funcionais. A aplicação do PARNAFOA foi realizada em cinco sistemas de software, com domínios, características e complexidades diversos. A avaliação da aplicação deste processo mostrou que o tratamento dos requisitos não-funcionais, desde as fases iniciais do desenvolvimento dos sistemas de software, complementa o modelo de casos de uso com funções adicionais ou gera restrições de projeto. Se estes requisitos não forem considerados desde o início, a introdução posterior dessas funções pode causar alterações nos modelos consolidados ou as atividades de projeto podem ser realizadas sem considerar as restrições. As aplicações do PARNAFOA e sua conseqüente melhoria, incorporada após sua avaliação, permitiu torná-lo mais flexível do que sua versão inicial. Aplicações futuras, com outros tipos de requisitos não-funcionais, irão permitir o amadurecimento deste processo.

Palavras-chave: Requisitos não-funcionais. Orientado a aspectos. NFR *Framework*. Casos de uso. Sistemas de software.

ABSTRACT

The aim of this thesis is to define an aspect-oriented non-functional requirements analysis process named PARNAFOA. This process applies non-functional requirements methods in an integrated manner, based on NFR *Framework*, and aspect-oriented methods. A use case model that embodies non-functional requirements as new functions is the main result obtained from this process. PARNAFOA application was performed in five software systems, with diverse features, domains and complexities. The evaluation of this process application showed that the treatment of these non-functional requirements, from the early phases of software systems development, complements the use case model with additional new functions or generates project restrictions. If these requirements are not considered from the very beginning, the introduction of these functions at a later phase can generate modifications in consolidated models or project activities, that do not consider these restrictions, can be performed. The PARNAFOA applications and consequent improvement, incorporated after the assessment, allowed it to become more flexible than the initial version. Future applications, with other non-functional requirements types, will provide this process maturity.

Keywords: Non-functional requirements. Aspect-oriented. NFR *Framework*. Use cases. Software systems.

LISTA DE ILUSTRAÇÕES

Figura 1 – Desenvolvimento da pesquisa.	29
Figura 2 – Definição e detalhamento do processo.	30
Figura 3 – Aplicação e melhoria do processo.	30
Figura 4 – Classificação dos RNFs adaptada de Sommerville e Kotonya (1998).	36
Figura 5 – Tipos de <i>softgoals</i> adaptados de Chung et al. (2000).	38
Figura 6 – Catálogo de tipos de refinamentos adaptado de Chung et al. (2000).	39
Figura 7 – Catálogo de tipos de contribuições adaptado de Chung et al. (2000).	40
Figura 8 – Possíveis rótulos dos <i>softgoals</i> adaptados de Chung et al. (2000). 42	
Figura 9 – Catálogo de decomposição de RNFs adaptado de Chung et al. (2000).	44
Figura 10 – Método de decomposição de sub-classe aplicado ao controle de espaço para contas regulares e ouro adaptado de Chung et al. (2000). ...	45
Figura 11 – Método de decomposição de atributos aplicado ao controle de tempo de resposta de contas ouro adaptado de Chung et al. (2000).	46
Figura 12 – Exemplo de correlações positivas e negativas adaptado de Chung et al. (2000).	47
Figura 13 – SIGs iniciais de desempenho e segurança adaptados de Chung et al. (2000).	49
Figura 14 – Decomposição de tipo dos <i>softgoals</i> de desempenho e segurança adaptado de Chung et al. (2000).	49
Figura 15 – Definição de prioridade para a acurácia de contas adaptada de Chung et al. (2000).	50
Figura 16 – Exemplos de operacionalizações dos <i>softgoals</i> de desempenho e segurança adaptados de Chung et al. (2000).	50
Figura 17 – Definição de interdependências implícitas adaptada de Chung et al. (2000).	51
Figura 18 – Registro de <i>softgoals</i> de afirmação adaptado de Chung et al. (2000).	52

Figura 19 – Seleção entre alternativas adaptada de Chung et al. (2000).	53
Figura 20 – Aplicação do procedimento de avaliação adaptada de Chung et al. (2000).	53
Figura 21 – Relacionamento com o requisito funcional adaptado de Chung et al. (2000).	54
Figura 22 – Fluxo de trabalho para capturar requisitos como casos de uso adaptado de Jacobson, Booch e Rumbaugh (1999).....	55
Figura 23 – Pontos de associação entre <i>softgoals</i> e elementos do diagrama de casos de uso.....	58
Figura 24 – Propagações de RNFs adaptadas de Supakkul e Chung (2004)..	60
Figura 25 – Composição de interesses adaptada de Baniassad et al. (2006).	68
Figura 26 – Modelo de requisitos orientados a aspectos de Araújo et al. (2002).	70
Figura 27 – Exemplo de diagrama composto com o interesse transversal de tempo de resposta adaptado de Araújo et al. (2002).....	71
Figura 28 – Modelo de requisitos para atributos de qualidade de Brito, Moreira e Araújo (2002).	72
Figura 29 – Exemplo de diagrama composto com o interesse transversal de tempo de resposta de Moreira, Araújo e Brito (2002).	74
Figura 30 – Exemplo do uso de operadores de combinação no diagrama de casos de uso de Moreira, Araújo e Brito (2002).....	74
Figura 31 – Abordagem direcionada a casos de uso aspectuais adaptada de Araújo e Moreira (2003).....	75
Figura 32 – Exemplos de casos de uso aspectuais de Araújo e Moreira (2003).	76
Figura 33 – Modelo de separação de interesses adaptado de Brito e Moreira (2004).	77
Figura 34 – Atividades de requisitos adaptadas de Sousa (2004).	81
Figura 35 – Exemplo de um relacionamento <i>crosscuts</i> adaptado de Sousa (2004).	82
Figura 36 – Modelo de engenharia de requisitos baseada no tratamento uniforme de interesses adaptado de Moreira, Araújo e Rashid (2005).....	86

Figura 37 – Especificação do interesse de mobilidade em XML.	86
Figura 38 – Processo de engenharia de requisitos orientado a aspectos adaptado de Chitchyan et al. (2006).	89
Figura 39 – Visão geral do PARNAFOA.	100
Figura 40 – Sub-processo obter os requisitos do sistema de software (1).	101
Figura 41 – Sub-processo relacionar interesses funcionais e não-funcionais (2).	103
Figura 42 – Sub-processo identificar aspectos (3).	106
Figura 43 – Sub-processo decompor requisitos não-funcionais (4).	109
Figura 44 – Métodos de decomposição de tópicos para o PARNAFOA.	112
Figura 45 – Método de decomposição de limite de sistema.	113
Figura 46 – Método de decomposição de caso de uso.	114
Figura 47 – Método de decomposição de ator.	115
Figura 48 – Método de decomposição relação ator-caso de uso.	116
Figura 49 – Sub-processo elaborar modelo composto (5).	119
Figura 50 – Diagrama de casos de uso (SGH).	125
Figura 51 – <i>Softgoals</i> (SGH).	128
Figura 52 – Diagrama integrado (SGH).	130
Figura 53 – SIG1 de acessibilidade (SGH).	134
Figura 54 – SIG 2 de desempenho relacionado ao limite do sistema (SGH).	135
Figura 55 – SIG 3 de desempenho relacionado ao caso de uso Reservar Quarto (SGH).	136
Figura 56 – SIG 4 de desempenho relacionado ao caso de uso Reservar Quarto (SGH).	137
Figura 57 – SIG 4 atualizado (SGH).	141
Figura 58 – Diagrama de casos de uso atualizado (SGH).	142
Figura 59 – Diagrama de casos de uso (SBM).	144
Figura 60 – <i>Softgoals</i> (SBM).	147
Figura 61 – Diagrama integrado (SBM).	148

Figura 62 – SIG 1 de desempenho associado ao caso de uso Ler Sensores (SBM).....	153
Figura 63 –SIG 2 de desempenho associado ao caso de uso Transmitir Mensagem (SBM).....	154
Figura 64 – SIG 3 de modificabilidade (SBM).	155
Figura 65 – SIG 4 de acurácia (SBM).	156
Figura 66 – Diagrama de casos de uso atualizado (SBM).	161
Figura 67 – Diagrama de casos de uso (ONC).	164
Figura 68 – <i>Softgoals</i> (ONC).....	166
Figura 69 – Diagrama integrado (ONC).	168
Figura 70 – SIG1 de acurácia (ONC).	171
Figura 71 – SIG 2 de desempenho (ONC).	173
Figura 72 – SIG 3 de desempenho (ONC).	173
Figura 73 – Diagrama de casos de uso atualizado (ONC).	176
Figura 74 – Diagrama de casos de uso (ATC).	179
Figura 75 – Diagrama de estados do trem (ATC).....	182
Figura 76 – <i>Softgoals</i> (ATC).....	184
Figura 77 – Diagrama integrado (ATC).	186
Figura 78 – SIG1 de confiabilidade (ATC).....	191
Figura 79 – SIG 4 de acurácia (ATC).	192
Figura 80 – SIG 8 de <i>safety</i> (ATC).	193
Figura 81 – Diagrama de casos de uso atualizado (ATC).....	198
Figura 82 – Diagrama de casos de uso (SGC).....	200
Figura 83 – <i>Softgoals</i> (SGC).	206
Figura 84 – Diagrama integrado (SGC).....	210
Figura 85 – SIG1 de <i>safety</i> (SGC).	225
Figura 86 – SIG 2 de disponibilidade (SGC).	226
Figura 87 – SIG 4 de espaço (SGC).	227

Figura 88 – SIG 5 de tempo (SGC).	228
Figura 89 – Diagrama de casos de uso atualizado – tempo (SGC).	238

LISTA DE TABELAS

Tabela 1 – Modelo de especificação de casos de uso adaptado de Rational University (2003).....	56
Tabela 2 – Formulário de especificação de um interesse transversal.....	70
Tabela 3 – Formulário de especificação de um atributo de qualidade adaptado de Moreira, Araújo e Brito (2002).....	73
Tabela 4 – Formulário de especificação de interesses adaptado de Brito e Moreira (2004).	77
Tabela 5 – Matriz de partes interessadas x interesses adaptada de Brito e Moreira (2004).	79
Tabela 6 – Tabela de composição de nível 1 para um requisito aspectual de Sousa (2004).	83
Tabela 7 – Exemplo da tabela de composição de nível 1 preenchida de Sousa (2004).	84
Tabela 8 – Tabela de conflitos entre requisitos aspectuais de Sousa (2004). .	85
Tabela 9 – Descrição das ações e operações de restrição adaptada de Moreira, Araújo e Rashid (2005).	87
Tabela 10 – Exemplo de simetria na tabela de interesses x interesses adaptado de Moreira, Araújo e Rashid (2005).	88
Tabela 11 – Mapeamento entre os passos do NFR <i>Framework</i> e o sub-processo 4 do PARNAFOA.....	109
Tabela 12 – Operacionalizações e táticas utilizadas na decomposição de operacionalização.....	117
Tabela 13 – Glossário (SGH).	125
Tabela 14 – Especificação do caso de uso Reservar Quarto (SGH).....	126
Tabela 15 – Especificação de requisitos não-funcionais (SGH).....	127
Tabela 16 – Tabela de interesses (SGH).	128
Tabela 17 – Tabela de RNF (SGH).	129
Tabela 18 – Tabela de RNF com propagação (SGH).	131
Tabela 19 – Tabela de RNF com aspectos (SGH).	132

Tabela 20 – Tabela de RNF com metas operacionais e restrições de projeto (SGH).....	138
Tabela 21 – Tabela de RNF com composição (SGH).	139
Tabela 22 – Exemplo da especificação do caso de uso Autenticar Usuário (SGH).....	142
Tabela 23 – Glossário (SBM).	144
Tabela 24 – Especificação do caso de uso Ler Sensores (SBM).....	145
Tabela 25 – Especificação de requisitos não-funcionais (SBM).....	146
Tabela 26 – Tabela de interesses (SBM).	147
Tabela 27 – Tabela de RNF (SBM).	147
Tabela 28 – Tabela de RNF com propagação (SBM).	150
Tabela 29 – Tabela de RNF com aspectos (SBM).	151
Tabela 30 – Tabela de RNF com metas operacionais e restrições de projeto (SBM).....	158
Tabela 31 – Tabela de RNF com composição (SBM).	159
Tabela 32 – Tabela de RNF com prioridades – caso de uso Leitura de Sensores (SBM).	160
Tabela 33 – Tabela de RNF com prioridades – caso de uso Transmitir Mensagem (SBM).....	160
Tabela 34 – Exemplo da especificação do caso de uso Coletar Mensagem 30 segundos (SBM).	161
Tabela 35 – Glossário (ONC).....	163
Tabela 36 – Especificação do caso de uso Monitorar e Informar Tráfego (ONC).	165
Tabela 37 – Especificação de requisitos não-funcionais (ONC).....	166
Tabela 38 – Tabela de interesses (ONC).....	166
Tabela 39 – Tabela de RNF (ONC).....	167
Tabela 40 – Tabela de RNF com propagação (ONC).	169
Tabela 41 – Tabela de RNF com aspectos (ONC).....	169

Tabela 42 – Tabela de RNF com metas operacionais e restrições de projeto (ONC).	174
Tabela 43 – Tabela de RNF com composição (ONC).	175
Tabela 44 – Especificação do caso de uso Verificar Posicionamento (ONC).	177
Tabela 45 – Glossário (ATC).	178
Tabela 46 – Especificação do caso de uso Verificação de Limite de Velocidade (ATC).	180
Tabela 47 – Especificação de requisitos não-funcionais (ATC).	181
Tabela 48 – Tabela de interesses (ATC).	184
Tabela 49 – Tabela de RNF (ATC).	184
Tabela 50 – Tabela de RNF com propagação (ATC).	187
Tabela 51 – Tabela de RNF com aspectos (ATC).	189
Tabela 52 – Tabela de RNF com metas operacionais e restrições de projeto (ATC).	194
Tabela 53 – Tabela de RNF com composição (ATC).	196
Tabela 54 – Tabela de RNF com prioridades (ATC).	197
Tabela 55 – Especificação do caso de uso Verificação Cruzada (ATC).	198
Tabela 56 – Glossário (SGC).	200
Tabela 57 – Especificação do caso de uso Resolver Crise (SGC).	202
Tabela 58 – Especificação de requisitos não-funcionais (SGC).	204
Tabela 59 – Tabela de interesses (SGC).	206
Tabela 60 – Tabela de RNF (SGC).	207
Tabela 61 – Tabela de RNF com propagação (SGC).	211
Tabela 62 – Tabela de RNF com aspectos (SGC).	218
Tabela 63 – Tabela de RNF com metas operacionais e restrições de projeto (SGC).	230
Tabela 64 – Tabela de RNF com composição (SGC).	233
Tabela 65 – Tabela de RNF com prioridades – caso de uso Atribuir Recurso Interno (SGC).	235

Tabela 66 – Tabela de RNF com prioridades – caso de uso Solicitar Recurso Externo (SGC).	236
Tabela 67 – Tabela de RNF com prioridades – caso de uso Executar Missão Resgate (SGC).	236
Tabela 68 – Tabela de RNF com prioridades – caso de uso Executar Missão Super Observador	237
(SGC).	237
Tabela 69 – Tabela de RNF com prioridades – caso de uso Resolver Crise (SGC).	237
Tabela 70 – Especificação do caso de uso Atualizar Dados (SGC).	239
Tabela 71 – Avaliação do PARNAFOA.	240

LISTA DE ABREVIATURAS E SIGLAS

ATC	<i>Automatic Train Control</i>
ATO	<i>Automatic Train Operation</i>
BPMN	<i>Business Process Modeling Language</i>
FCD	<i>Floating Car Data</i>
GIS	<i>Geographic Information System</i>
IHC	Interface Humano Computador
NFR	<i>Non Functional Requirement</i>
ONC	Olhos No Céu
PARNAFOA	Processo de Análise de Requisitos Não-funcionais Orientado a Aspectos
RNF	Requisito Não-Funcional
RUP	<i>Rational Unified Process</i>
SBM	Sistema de Bóias Meteorológicas
SGC	Sistema de Gerenciamento de Crise
SGH	Sistema de Gerenciamento de Hotel
SIG	<i>Softgoal Interdependency Graph</i>
UML	<i>Unified Modeling Language</i>
V&V	Verificação & Validação

SUMÁRIO

1.	Introdução	21
1.1	Motivação	21
1.2	Objetivo.....	23
1.3	Justificativa	25
1.4	Metodologia de Trabalho.....	29
1.5	Estrutura da Tese.....	31
2.	Requisitos Não-Funcionais	33
2.1	Classificação dos Requisitos Não-Funcionais.....	33
2.2	<i>NFR Framework</i>	36
2.2.1	Conceitos Principais	36
2.2.2	Tipos de <i>Softgoals</i>	38
2.2.3	Interdependência entre <i>Softgoals</i>	39
2.2.4	Procedimento de Avaliação	41
2.2.5	Métodos de Refinamento de <i>Softgoals</i>	43
2.2.6	Correlação entre <i>Softgoals</i>	46
2.2.7	Construção do SIG	48
2.3	Requisitos Não-Funcionais no modelo de casos de uso	54
2.3.1	Casos de Uso.....	55
2.3.2	Abordagem Orientada a <i>Softgoals</i> e Casos de Uso.....	58
2.4	Análise e Validação de Requisitos	61
2.5	Considerações Finais	63
3.	Engenharia de Requisitos Orientada a Aspectos	65
3.1	Conceitos e Definições.....	65
3.2	Separação e Composição de Interesses.....	67
3.3	Trabalhos Relevantes da Engenharia de Requisitos Orientada a Aspectos.....	69
3.3.1	Modelo de Engenharia de Requisitos Orientado a Aspectos com UML ...	69

3.3.2	Modelo de Atributos de Qualidade Transversais para a Engenharia de Requisitos.....	72
3.3.3	Abordagem direcionada a Casos de Uso Aspectuais	75
3.3.4	Integração do NFR <i>Framework</i> no modelo de Engenharia de Requisitos	77
3.3.5	Abordagem Direcionada a Casos de Uso para o Desenvolvimento de Software Orientado a Aspectos	80
3.3.6	Modelo de Engenharia de Requisitos Orientado a Interesses.....	85
3.3.7	Processo da Engenharia de Requisitos Orientado a Aspectos AOSD-Europe	89
3.3.8	Outras Abordagens	91
3.4	Considerações Finais	93
4.	Descrição do Processo de Análise de Requisitos Não-Funcionais Orientado a Aspectos – PARNAFOA.....	95
4.1	Características Principais do PARNAFOA.....	95
4.2	Visão Geral do PARNAFOA.....	99
4.3	Sub-Processo 1: Obter os Requisitos do Sistema de Software.	101
4.4	Sub-Processo 2: Relacionar Interesses Funcionais e Não-Funcionais	102
4.5	Sub-Processo 3: Identificar Aspectos.....	105
4.6	Sub-Processo 4: Decompor Requisitos Não-Funcionais	108
4.6.1	Detalhamento da Atividade para Decompor <i>Softgoals</i>	111
4.6.2	Detalhamento da Atividade para Identificar Operacionalizações	116
4.7	Sub-Processo 5: Elaborar o Modelo Composto	119
4.8	Considerações Finais	122
5.	Aplicações do PARNAFOA.....	123
5.1	Sistema de Gerenciamento de Hotel	124
5.1.1	Sub-processo 1: Obter os requisitos do sistema de software.....	124
5.1.2	Sub-processo 2: Relacionar interesses funcionais e não-funcionais.....	128
5.1.3	Sub-processo 3: Identificar aspectos	131

5.1.4	Sub-processo 4: Decompor requisitos não-funcionais	133
5.1.5	Sub-processo 5: Elaborar modelo composto	137
5.2	Sistema de Bóias Meteorológicas	143
5.2.1	Sub-processo 1: Obter os requisitos do sistema de software.....	144
5.2.2	Sub-processo 2: Relacionar interesses funcionais e não-funcionais.....	146
5.2.3	Sub-processo 3: Identificar aspectos	149
5.2.4	Sub-processo 4: Decompor requisitos não-funcionais	152
5.2.5	Sub-processo 5: Elaborar modelo composto	157
5.3	Sistema Olhos no Céu	162
5.3.1	Sub-processo 1: Obter os requisitos do sistema de software.....	163
5.3.2	Sub-processo 2: Relacionar interesses funcionais e não-funcionais.....	166
5.3.3	Sub-processo 3: Identificar aspectos	168
5.3.4	Sub-processo 4: Decompor requisitos não-funcionais	170
5.3.5	Sub-processo 5: Elaborar modelo composto	174
5.4	Sistema de Controle Automático de Trens.....	177
5.4.1	Sub-processo 1: Obter os requisitos do sistema de software.....	178
5.4.2	Sub-processo 2: Relacionar interesses funcionais e não-funcionais.....	183
5.4.3	Sub-processo 3: Identificar aspectos	187
5.4.4	Sub-processo 4: Decompor requisitos não-funcionais	190
5.4.5	Sub-processo 5: Elaborar modelo composto	194
5.5	Sistema de Gerenciamento de Crise	199
5.5.1	Sub-processo 1: Obter os requisitos do sistema de software.....	200
5.5.2	Sub-processo 2: Relacionar interesses funcionais e não-funcionais.....	206
5.5.3	Sub-processo 3: Identificar aspectos	211
5.5.4	Sub-processo 4: Decompor requisitos não-funcionais	224
5.5.5	Sub-processo 5: Elaborar modelo composto	229
5.6	Avaliação das Aplicações do PARNAFOA.....	239
5.7	Considerações Finais	243

6.	Considerações Finais	246
6.1	Conclusões.....	246
6.2	Contribuições.....	247
6.3	Trabalhos Futuros.....	249
	REFERÊNCIAS.....	251
	APÊNDICE A – Notação BPMN (<i>Business Process Modeling Notation</i>).....	257
	APÊNDICE B – Descrição dos Artefatos.....	260
	APÊNDICE C – Descrição das Atividades	265

1. Introdução

"A melhor maneira de ter uma boa idéia é ter várias idéias." -- Linus Pauling

Este capítulo apresenta a motivação, o objetivo, a justificativa, a metodologia do desenvolvimento deste trabalho e a sua estrutura.

1.1 Motivação

As características dos sistemas de software devem ser conhecidas desde o início do seu processo de desenvolvimento, para que o produto resultante satisfaça às necessidades reais das partes interessadas¹. Dentre estas características podem ser citados os requisitos não-funcionais que possuem importância fundamental sobre as decisões de projeto (CHUNG, et al., 2000).

O tratamento dos requisitos não-funcionais desde as fases iniciais de desenvolvimento de sistemas de software nem sempre é trivial. No entanto, estes requisitos podem causar impactos no sistema de software devido a inserção de funções adicionais, necessidade de maior rigor no processo de desenvolvimento, necessidade de documentação mais detalhada e necessidade de análise de conflitos que estes requisitos podem apresentar entre si.

Muitas vezes, os requisitos não-funcionais não ficam confinados em um único módulo ou subsistema e ficam espalhados em diversos pontos do sistema de software. Isto causa dificuldade para identificar o local em que esses requisitos devem ser inseridos e, posteriormente, recuperar a informação de onde foram inseridos.

Ainda, pode-se dizer que, apesar dos grandes esforços despendidos no desenvolvimento de sistemas de software com maior qualidade, muitos defeitos

¹ Partes interessadas, traduzido do inglês *stakeholders*, são pessoas ou organizações, como clientes, patrocinadores, organizações executoras e o público, que estejam ativamente envolvidos no desenvolvimento ou cujos interesses possam ser afetados de forma positiva ou negativa pela execução ou término do desenvolvimento. As partes interessadas podem também exercer influência sobre o desenvolvimento e suas entregas (PMI, 2009).

ainda são encontrados na fase de produção. Uma das maiores causas destes defeitos está relacionada ao espalhamento e entrelaçamento de interesses² (EADDY et al., 2008).

Um interesse pertence ao desenvolvimento do sistema de software, sua operação ou qualquer questão que seja crítica ou importante para uma ou mais partes interessadas (BERG; CONEJERO; CHITCHYAN, 2005). Outros autores definem interesse como um conjunto de requisitos relacionados (MILLI; ELKHARRAZ; MCHEICK, 2004), ou qualquer coisa de interesse de alguma parte interessada, seja um usuário final, um responsável de projeto, ou mesmo um desenvolvedor (JACOBSON; NG, 2004), algum tipo de funcionalidade do sistema de software, uma característica ou um tipo de processamento (CLARKE; BANIASSAD, 2005).

Os interesses podem possuir uma característica transversal³, isto é, são responsáveis por produzirem representações entrelaçadas e espalhadas que são difíceis de entender e manter. O entrelaçamento ocorre quando múltiplos interesses ficam misturados em um único módulo, e o espalhamento faz com que um único interesse afete múltiplos módulos. Por módulo entende-se um artefato modular (BERG, CONEJERO, CHITCHYAN, 2005).

O entrelaçamento e o espalhamento causam impactos tais como dificuldade no controle de modificação, dificuldade de compreensão, dificuldade de reuso e deficiência na rastreabilidade (CLARKE; BANIASSAD, 2005). De acordo com Berg, Conejero e Chitchyan (2005) um interesse transversal não pode ser representado de maneira modular, através de um método de decomposição⁴ e, conseqüentemente, um interesse transversal é espalhado e entrelaçado entre outros interesses.

Os interesses do tipo segurança, desempenho, confiabilidade, disponibilidade, inerentes aos sistemas críticos são transversais e devem ser

² O uso do termo interesse foi escolhido como tradução para o termo *concern* (PIVETA et al., 2004).

³ O termo aspecto tem sido amplamente utilizado como um substituto equivalente do termo interesses transversais (NIU; EASTERBROOK; YU, 2007).

⁴ Decomposição é o particionamento de um problema em um conjunto de problemas menores que podem ser tratados individualmente (BERG; CONEJERO; CHITCHYAN, 2005).

tratados de forma sistemática, pois podem causar danos substanciais ao sistema e seus usuários, se não forem considerados adequadamente.

As técnicas orientadas a aspectos têm se tornado cada vez mais reconhecidas no meio acadêmico e profissional pelas melhorias na modularização de interesses transversais em todos os estágios do desenvolvimento de sistemas de software (GREENWOOD et al., 2007).

Entretanto, a identificação de interesses deve ser feita nas fases iniciais do desenvolvimento dos sistemas de software, pois o tratamento de interesses na fase de implementação é tardia e muito custosa, uma vez que muitas decisões de projeto já foram tomadas. Por este motivo, métodos de detecção de aspectos durante a fase de requisitos tem se tornado cada vez mais necessários (RASHID et al., 2002).

Adicionalmente, é necessário que a identificação e tratamento de interesses não-funcionais seja feita de maneira organizada desde as fases iniciais de desenvolvimento. A definição de processos para o tratamento de interesses, que utilizam técnicas orientadas a aspectos (CHITCHYAN et al., 2006) faz com que seja possível tratar estas questões de maneira organizada e disciplinada.

1.2 Objetivo

O objetivo deste trabalho é definir um processo de análise de requisitos não-funcionais (RNF) que integra os RNFs aos requisitos funcionais representados pelo modelo de casos de uso. Os RNFs considerados relevantes ao sistema de software são detalhados até que se possam identificar novas funções e restrições de projeto, relacionadas a estes requisitos, que possam ser incorporadas ao modelo do sistema de software. Através de abordagem orientada a aspectos são identificados os pontos onde estas novas funções são integradas ao modelo de casos de uso do sistema de software. Possíveis conflitos entre requisitos não-funcionais podem surgir durante a integração no modelo de casos de uso e por isso, devem ser detectados e tratados. Este processo foi denominado PARNAFOA (**P**rocesso de **A**nálise de **R**equisitos **N**ão-**F**uncionais **O**rientado a **A**spectos).

Do ponto de vista macroscópico, o PARNAFOA recebe, como entrada, o modelo de casos de uso e a descrição dos RNFs do sistema de software e gera como saída um modelo que integra os casos de uso às novas funções relacionadas aos RNFs e identifica as restrições de projeto provenientes desses RNFs.

Dentre os principais recursos selecionados para compor o PARNAFOA encontram-se a modelagem de casos de uso (JACOBSON; BOOCH; RUMBAUGH, 1999), o NFR *Framework* (CHUNG et al., 2000), a proposta de integração do diagrama de casos de uso ao NFR *Framework* (SUPAKKUL; CHUNG, 2004) e os métodos orientados a aspectos que consideram o tratamento de RNFs (RASHID et al., 2002), (ARAÚJO et al., 2002), (BRITO; MOREIRA; ARAÚJO, 2002), (MOREIRA; ARAÚJO; BRITO, 2002), (ARAÚJO; MOREIRA, 2003), (BRITO; MOREIRA, 2004), (BRITO; MOREIRA, 2003), (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003), (RASHID; MOREIRA; ARAÚJO, 2003), (MOREIRA; ARAÚJO; RASHID, 2005), (MOREIRA; RASHID; ARAÚJO, 2005), (CHITCHYAN et al., 2006). Estes recursos foram detalhados para que fossem integrados às atividades do PARNAFOA.

O modelo de casos de uso foi selecionado para representar os requisitos funcionais, pois permite uma descrição detalhada e padronizada das funções do sistema de software. É também um recurso consolidado há mais de uma década para elicitación de requisitos (JACOBSON; NG, 2004), (SUPAKKUL; CHUNG, 2004).

O tratamento de RNF é feito usando o NFR *Framework*. Este método foi selecionado por ser um método referenciado na maior parte dos trabalhos da literatura, para tratamento de requisitos não-funcionais, e por apresentar uma proposta sistemática e menos subjetiva para tratamento dos RNFs (CHUNG et al., 2000).

Para a integração do NFR *Framework* com o modelo de casos de uso foi utilizada a proposta de Supakkul e Chung (2004), por apresentar diretrizes para relacionar elementos do NFR *Framework* aos elementos do modelo de casos de uso, estabelecer regras a respeito dos impactos que um RNF pode apresentar nos diversos elementos do modelo de casos de uso e por possuir

uma ferramenta adaptada e disponível para uso, a StarUML (LEE et al., 1995), para representar esta integração.

Como os RNFs podem ser caracterizados como interesses transversais ou não-transversais de um sistema de software, foram considerados os métodos de identificação de aspectos nas fases iniciais de desenvolvimento, para que o resultado do detalhamento dos requisitos não-funcionais possa ser representado posteriormente como aspectos (RASHID et al., 2002), (ARAÚJO et al., 2002), (BRITO; MOREIRA; ARAÚJO, 2002), (MOREIRA; ARAÚJO; BRITO, 2002), (ARAÚJO; MOREIRA, 2003), (BRITO; MOREIRA, 2004), (BRITO; MOREIRA, 2003), (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003), (RASHID; MOREIRA; ARAÚJO, 2003), (MOREIRA; ARAÚJO; RASHID, 2005), (MOREIRA; RASHID; ARAÚJO, 2005), (CHITCHYAN et al., 2006).

1.3 Justificativa

Diversas abordagens que tratam RNFs, modelo de casos de uso e aspectos e que os relacionam foram encontradas na literatura.

O modelo de atributos de qualidade transversais para a engenharia de requisitos propõe um modelo de processo para identificar, especificar e integrar interesses não-funcionais, que foram chamados de atributos de qualidade. Os atributos de qualidade, classificados como transversais, que afetam mais de um caso de uso, são especificados através de tabelas e incluídos ao modelo de casos de uso do sistema de software através de relacionamentos de inclusão (BRITO; MOREIRA; ARAÚJO, 2002), (MOREIRA; ARAÚJO; BRITO, 2002).

A proposta de integração do NFR *Framework* no modelo de engenharia de requisitos define um modelo para separar interesses durante as etapas da engenharia de requisitos. Interesses transversais, requeridos por mais de um interesse, são identificados, especificados e compostos ao modelo de casos de uso do sistema de software. Nesta proposta, o NFR *Framework* é utilizado para decompor os requisitos não-funcionais até o nível da definição de estratégias de projeto. Esta informação é utilizada para descrever os interesses e

identificar outros interesses que possam ser afetados (BRITO; MOREIRA, 2004), (BRITO; MOREIRA, 2003).

A abordagem direcionada a casos de uso para o desenvolvimento de software orientado a aspectos propõe um fluxo de atividades de requisitos, sendo que um conjunto das atividades possui foco no tratamento de interesses não-funcionais. Os interesses não-funcionais são identificados e decompostos até que se obtenham estratégias de projeto, através da aplicação do *NFR Framework*. Estes interesses são especificados e incluídos no modelo de caso de uso através de um novo tipo de relacionamento, chamado *crosscuts* (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003).

O desenvolvimento de software orientado a aspectos com casos de uso define que todos os casos de uso são interesses transversais em potencial, uma vez que sua realização atravessa diversas classes. Interesses não-funcionais relacionados ao sistema de software são adicionados como novos casos de uso, porém, as diretrizes para a identificação e integração de interesses não-funcionais durante a fase de requisitos não é detalhada (JACONSON; NG, 2004).

A abordagem direcionada a casos de uso aspectuais estabelece um conjunto de passos para identificar casos de uso aspectuais. Um caso de uso é considerado aspectual quando for estendido ou incluído por outros casos de uso. Interesses não-funcionais transversais são integrados ao diagrama de casos de uso do sistema de software, através de um novo tipo de relacionamento, chamado restrição (ARAÚJO; MOREIRA, 2003).

O modelo de engenharia de requisitos orientado a interesses propõe um modelo de engenharia de requisitos para tratamento uniforme de interesses funcionais e não-funcionais. Os interesses transversais, que afetam mais de um interesse, são identificados e especificados através da linguagem XML e posteriormente compostos (RASHID; MOREIRA; ARAÚJO, 2003), (MOREIRA; ARAÚJO; RASHID, 2005), (MOREIRA; RASHID; ARAÚJO, 2005).

Em relação à detecção de aspectos na fase inicial de desenvolvimento destaca-se o modelo de engenharia de requisitos orientado a aspectos com UML, que propõe um conjunto de atividades para lidar com interesses transversais na fase de requisitos. Os interesses transversais não-funcionais que afetam mais de um caso de uso são identificados, especificados e

integrados ao modelo de casos de uso do sistema de software através de novos tipos de relacionamentos (RASHID et al., 2002), (ARAÚJO et al., 2002).

O processo da engenharia de requisitos orientada a aspectos AOSD-Europe apresenta uma visão consolidada a respeito de diversas abordagens relacionadas à área de engenharia de requisitos orientada a aspectos, porém, não detalha a execução de suas atividades nem mesmo os artefatos utilizados (CHITCHYAN et al., 2006).

Nenhuma das abordagens descritas anteriormente apresenta um processo definido e documentado de forma sistemática, através de sub-processos, atividades, procedimentos e artefatos. Neste contexto, a relevância da definição de um processo que dê diretrizes para que sua execução possa ser repetida em diversos tipos de aplicações é de reconhecida importância. Além disso, um processo definido pode balancear a influência das pessoas, tecnologia e o ambiente, tornando as organizações melhor preparadas para um mundo em constante mudança (GARCIA; TURNER, 2007).

A proposta de integração do NFR *Framework* no modelo de engenharia de requisitos (BRITO; MOREIRA, 2004), (BRITO; MOREIRA, 2003) e a abordagem direcionada a casos de uso para o desenvolvimento de software orientado a aspectos de (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003) fazem uso do NFR *Framework*. Porém, no primeiro trabalho as estratégias de projeto identificadas são utilizadas apenas para descrever os interesses e identificar possíveis interesses afetados. No segundo trabalho, restrições de projeto, isto é, restrições que não se transformam em funções a serem adicionadas no modelo de casos de uso do sistema de software não fazem parte do escopo do trabalho.

A utilização do NFR *Framework* traz benefícios na identificação de novas funções e restrições a serem incorporadas ao modelo do sistema de software. O resultado do detalhamento dos requisitos não-funcionais deve ser considerado para que o modelo do sistema de software possa ser atualizado com novas informações, sejam elas novas funções a serem incluídas ou mesmo restrição de projeto. O uso do NFR *Framework* durante a fase inicial de desenvolvimento traz o benefício da antecipação da definição de possíveis estratégias de projeto, que somente seriam detectadas em fases posteriores.

Nenhum dos trabalhos citados anteriormente apresentou propostas de detalhamentos dos métodos utilizados. Os detalhamentos constituem um fator importante para que as atividades do processo sejam executadas de maneira consistente. É importante ressaltar que os detalhamentos propostos neste trabalho respeitam os padrões previamente estabelecidos pelos métodos em suas versões originais.

Trabalhos que tratam do modelo de engenharia de requisitos orientado a aspectos com UML (RASHID et al., 2002), (ARAÚJO et al., 2002); abordagem direcionada a casos de uso aspectuais (ARAÚJO; MOREIRA, 2003), integração do NFR *Framework*, no modelo de engenharia de requisitos (BRITO; MOREIRA, 2004), (BRITO; MOREIRA, 2003); abordagem direcionada a casos de uso para o desenvolvimento de software orientado a aspectos (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003) apresentaram propostas de criação de novos tipos de relacionamentos entre os novos casos de uso adicionados ao modelo do sistema de software. Neste trabalho optou-se pela utilização do relacionamento de inclusão, respeitando o padrão UML, para representar novos comportamentos adicionados.

A proposta de Supakkul e Chung (2004) apresenta uma integração entre o modelo de casos de uso e o NFR *Framework* com o objetivo de detectar os impactos que os requisitos não-funcionais podem causar nos diversos elementos do modelo de casos de uso. Entretanto, esta proposta não possui uma abordagem orientada a aspectos, o que seria interessante, pois os requisitos não-funcionais podem ficar espalhados e entrelaçados em vários elementos do modelo de casos de uso.

Portanto, esta tese justifica-se pela necessidade da elaboração de um processo orientado a aspectos para tratamento de RNFs a fim de que possa ser aplicado por equipes que desenvolvam sistemas de software em diferentes domínios de aplicação. A necessidade de definição de um processo que utiliza métodos de maneira integrada e detalhada é também de igual relevância. A decomposição dos requisitos não-funcionais, a análise do comportamento transversal e não-transversal de cada um deles e a integração destes RNFs no modelo de casos de uso do sistema de software complementam a proposta desta tese.

1.4 Metodologia de Trabalho

O trabalho foi iniciado com a identificação dos trabalhos considerados relevantes para a definição do PARNAFOA. Estes trabalhos passaram por uma análise crítica, a fim de levantar o estado da arte sobre o assunto e identificar sua aplicabilidade e contribuição.

O desenvolvimento da pesquisa foi realizado conforme Figura 1. Foi utilizada a notação BPMN (OMG-BPMN, 2009). Um resumo da notação BPMN é apresentado no Apêndice A.

Figura 1 – Desenvolvimento da pesquisa.

O PARNAFOA foi definido e detalhado, com base nos métodos e nas abordagens da literatura, considerados relevantes. Posteriormente foi aplicado em cinco sistemas de software de domínios e complexidades diversos. A escolha desses sistemas de software foi feita de maneira que cada uma das aplicações pudesse exercitar o processo de maneira diferente, para obter mais informações e contribuir para a melhoria do processo final.

Segundo Mendonça et al. (2008), a aplicação de um processo pode ser interna ou externa. Uma aplicação é interna, quando uma equipe define o processo e conduz as aplicações; a aplicação é externa, quando uma equipe define o processo e uma outra conduz a aplicação.

As aplicações do PARNAFOA foram internas, para que fosse possível avaliar em detalhes as suas vantagens e desvantagens. Por isso, optou-se pela aplicação em cinco sistemas de software de domínios e complexidades diversos. A escolha desses sistemas de software foi feita de forma que cada uma das aplicações pudesse exercitar o processo de maneira diferente, para obter mais informação e contribuir para a melhoria do processo final.

A melhoria do PARNAFOA foi realizada, considerando-se os resultados da sua aplicação aos sistemas de software selecionados.

As atividades do sub-processo de definição e detalhamento do processo são apresentadas na Figura 2.

Figura 2 – Definição e detalhamento do processo.

A atividade para definir versão inicial do processo estabeleceu a versão inicial do PARNAFOA. Durante a atividade para selecionar métodos e técnicas a serem utilizadas, métodos e técnicas da literatura foram selecionadas, detalhadas e incorporadas ao processo, complementando a definição das suas atividades. Durante a atividade para detalhar sub-processos, atividades, procedimentos e artefatos, estes elementos do PARNAFOA foram definidos. Durante a atividade para documentar o processo, a informação relativa ao processo foi registrada.

A Figura 3 apresenta o detalhamento dos subprocessos de aplicação do processo e de melhoria do processo. As atividades da Figura 3 foram adaptadas da proposta de Wohlin, Host e Henningsson (2003) e Lindvall et al. (2005).

Figura 3 – Aplicação e melhoria do processo.

O sub-processo de aplicação do processo, foi executado iterativamente para cada sistema de software. Durante a atividade para selecionar sistema de software, foi escolhido um sistema de software que introduzisse abordagens relevantes para a aplicação do PARNAFOA. Um dos principais critérios para a seleção foi a consideração de um sistema que introduzisse RNFs não tratados anteriormente. A atividade para preparar os artefatos necessários para o sistema de software contou com a identificação, elaboração e adaptação dos artefatos de entrada, tal que fossem compatíveis com a necessidade do PARNAFOA. Durante a atividade para executar a aplicação do processo, foram definidas as principais questões a serem respondidas durante o estudo dos resultados de cada uma delas, além da própria execução da aplicação, de acordo com o PARNAFOA. A atividade para analisar e documentar os resultados da aplicação do processo teve o objetivo de registrar as informações coletadas durante a execução da aplicação e analisá-las, buscando identificar pontos de melhoria do PARNAFOA.

O sub-processo de melhoria do processo é composto das atividades para melhorar o processo com base na análise dos resultados e para atualizar os artefatos das aplicações. A primeira atividade teve a finalidade de atualizar o processo original com base no resultado das análises das aplicações. A segunda atividade teve o objetivo de atualizar os artefatos de cada um dos sistemas, com base no processo atualizado.

1.5 Estrutura da Tese

Esta tese é composta por mais cinco capítulos. No Capítulo 2, são apresentados os conceitos sobre requisitos não-funcionais e os métodos da literatura sobre seu tratamento, que foram considerados relevantes para o trabalho. No Capítulo 3, são apresentados os conceitos de Engenharia de Requisitos com foco na orientação a aspectos e os trabalhos relacionados da literatura. No Capítulo 4, é apresentada a descrição do PARNAFOA, através de seus sub-processos, suas atividades e seus artefatos de entrada e saída. No Capítulo 5, é feito o relato da aplicação do processo na análise de cinco sistemas de software, ressaltando os pontos mais relevantes, além da

apresentação da avaliação dos resultados. No Capítulo 6, são apresentadas as considerações finais, através das conclusões, das contribuições deste trabalho e dos trabalhos futuros que podem ser desenvolvidos.

2. Requisitos Não-Funcionais

“A velocidade só é útil se você estiver correndo na direção certa.”-- Tom Chung

Diversas abordagens têm sido propostas para incorporar requisitos não-funcionais (RNFs) ao desenvolvimento de sistemas de software (DOERR et al., 2005). Dentre elas foram selecionadas três: casos de uso (JACOBSON; BOOCH; RUMBAUGH, 1999), *NFR Framework* (CHUNG et al., 2000) e uma proposta de combinação de casos de uso com conceitos do *NFR Framework* (SUPAKKUL; CHUNG, 2004).

Este capítulo tem o objetivo de apresentar conceitos sobre RNFs e os métodos da literatura que foram considerados relevantes para o trabalho.

2.1 Classificação dos Requisitos Não-Funcionais

Requisitos podem ser descrições sobre o comportamento do sistema de software, informações sobre o domínio da aplicação, restrições de operação, ou especificação de suas propriedades ou atributos. Os requisitos podem também representar restrições no processo de desenvolvimento: por exemplo, no caso de sistemas críticos, as atividades de verificação e validação são mais rigorosas. Os requisitos devem ser definidos nas fases iniciais do desenvolvimento como uma especificação do que deve ser implementado (KOTONYA; SOMMERVILLE, 1998).

Uma outra definição é dada por Leffingwell e Widrig (2001), em que requisitos de sistemas de software expressam as necessidades do usuário a fim de que seja possível resolver um determinado problema. Além disso, um requisito de software representa a capacidade que precisa ser obtida ou possuída pelo sistema de software, ou seu componente, para satisfazer contrato, padrão, especificação, ou outra documentação formalmente imposta.

Existem dois tipos de requisitos de software, os funcionais e os não-funcionais. Os requisitos funcionais especificam funções que o sistema, sistema de software ou componente de sistema ou software deve ser capaz de executar (THAYER, 1990 apud CHUNG et al., 2000). Definem o

comportamento do sistema, através da transformação que os componentes de software ou hardware de um sistema realizam nas entradas para produzirem saídas.

Ainda, de acordo com Cysneiros (2001), requisitos funcionais são definições que expressam funções ou serviços que um sistema de software deve ou pode ser capaz de executar ou fornecer. Funções ou serviços são entendidos como processos que utilizam entradas para produzir saídas.

Requisitos não-funcionais, por outro lado, são requisitos que declaram restrições. De acordo com Xavier (2009) requisitos não-funcionais se relacionam aos padrões de qualidade e definem se o sistema de software será eficiente e adequado para a tarefa que se propõe a fazer.

Os requisitos não-funcionais, ao contrário dos requisitos funcionais, não determinam as funções a serem realizadas pelos sistemas de software, mas os comportamentos e restrições que os sistemas de software devem satisfazer (CYSNEIROS; LEITE, 1998 apud XAVIER 2009).

Existem outras denominações para RNFs como atributos de qualidade, restrições, objetivos. O termo requisito não-funcional tem se consolidado no meio acadêmico e, portanto, foi adotado neste trabalho.

A complexidade de um sistema de software é determinada por seus requisitos funcionais e não-funcionais, tais como: desempenho, confiabilidade, manutenibilidade, portabilidade, custos operacionais, entre outros. Os requisitos não-funcionais, por sua vez, desempenham um papel crítico durante o desenvolvimento de sistemas de software pois, se definidos de maneira incorreta, são de difícil correção e podem gerar grandes impactos em custos (CHUNG et al., 2000).

Neste contexto Chung et al. (2000) destacam que os requisitos não-funcionais representam um papel importante durante o desenvolvimento de sistemas de software, servindo como base para critérios de seleção entre as alternativas de projeto e implementação. Além disso, os autores consideram que os requisitos não-funcionais são subjetivos, relativos e interativos. São subjetivos, pois podem ser interpretados e avaliados de maneira diversa. São relativos, pois sua interpretação e importância podem variar dependendo do sistema de software considerado. Finalmente, são interativos, pois o atendimento de um RNF pode ferir ou beneficiar o atendimento de outros

RNFs; portanto, soluções localizadas, que não consideram o sistema de software como um todo, podem não ser as mais adequadas. Desta forma, o contexto de aplicação de um RNF precisa ser definido para que um RNF possa ser considerado atendido ou não-atendido. Exemplos de contextos aplicáveis para RNFs são: “o sistema de software deve ter acurácia especialmente quando existirem tentativas de fraudes” e “o sistema de software deve ser rápido especialmente em períodos de pico”.

Existem diversas classificações para os tipos de RNFs na literatura. Para ilustrar, foram consideradas as classificações da norma IEEE *Standard* 830 (IEEE, 1998), do padrão internacional ISO/IEC 9126 (ISO/IEC, 2001), de Chung et al. (2000) e de Sommerville e Kotonya (1998).

A norma IEEE *Standard* 830 (IEEE, 1998) apresenta uma lista de exemplos de tipos de RNFs: desempenho, interface, recursos, verificação, aceitação, documentação, *security*, *safety*⁵, portabilidade, qualidade, confiabilidade e manutenibilidade.

O padrão internacional ISO/IEC 9126 (ISO/IEC, 2001) apresenta seis características que definem a qualidade de software: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Estas características de qualidade são subdivididas em sub-características para melhor definir os RNFs dos sistemas de software.

Chung et al. (2000) apresentam uma lista extensa com mais de uma centena de tipos de RNFs. Dentre eles, os autores detalham o tratamento relativo aos requisitos de acurácia, *security* e desempenho.

Sommerville e Kotonya (1998) apresentam uma outra forma de classificação de RNFs, agrupando-os em aqueles relacionados com produto, processo e ambiente externo. Os requisitos de produto são aqueles que especificam as características desejadas que um sistema de software ou subsistema devem possuir e geralmente são os mais usados. Os requisitos de processo são restrições ao processo de desenvolvimento do sistema de software e correspondem a definição de padrões, uso de ferramentas para gerenciamento, equipe treinada segundo um modelo de qualidade, por

⁵ Os termos *security* e *safety* serão mantidos em inglês neste trabalho pela inexistência de traduções adequadas em português.

exemplo, CMMi (CHRISSIS; KONRAD; SHRUM, 2007) ou a norma ISO/IEC 12207. Os requisitos de ambiente externo influem nos requisitos de produto e de processo e podem ser classificados em: legal, econômico e de interoperabilidade. A Figura 4 apresenta esta classificação.

Figura 4 – Classificação dos RNFs adaptada de Sommerville e Kotonya (1998).

2.2 NFR Framework

O método mais referenciado na literatura para tratamento de RNFs é o NFR *Framework* e foi proposto por Chung et al. (2000). Esta seção apresenta uma descrição resumida deste método.

2.2.1 Conceitos Principais

O NFR *Framework* é um método sistemático de representação e análise de requisitos não-funcionais (CHUNG et al., 2000).

Como se trata de um requisito, um RNF deve ser atendido por um sistema de software. No entanto, como um RNF é considerado subjetivo, relativo e interativo pelos autores, o conceito de meta ou *goal* para representá-lo não é adequado. Desta forma, foi definido o conceito de *softgoal*, para representar o RNF, o qual deve ser suficientemente atendido, mas não absolutamente atendido, como no caso de um *goal*.

O termo *softgoal* será mantido em inglês neste trabalho, pois não existe uma tradução consolidada. Seu significado está relacionado ao conceito do termo *satisficing*, em inglês, que corresponde à satisfação em um determinado nível. O termo *satisfice* foi usado nos anos cinquenta e, atualmente, não existe mais nos dicionários atuais (CHUNG et al., 2000).

Os *softgoals* são representados graficamente como nuvens e conectados entre si através de relações de interdependência. A análise do atendimento de um *softgoal* deve ser feita conjuntamente com os demais *softgoals* de um sistema de software. Um *softgoal* pode contribuir positivamente ou negativamente, suficientemente ou parcialmente, para o atendimento de outro *softgoal*.

O NFR *Framework* possui, como principal representação, o grafo de interdependência de *softgoals* ou SIG (*Softgoal Interdependency Graph*). Este grafo representa os *softgoals* e seus relacionamentos, bem como as considerações dos desenvolvedores sobre o nível de satisfação dos RNFs. A elaboração deste grafo é feita através de cinco componentes principais que são: *softgoals*, interdependências, procedimento de avaliação, métodos e correlações. Estes cinco componentes são detalhados nas seções seguintes.

Outro componente importante do NFR *Framework* são os catálogos. Um catálogo é um agrupamento organizado de um corpo de conhecimento sobre os RNFs, que pode ser usado em diferentes domínios de aplicação. Existem três tipos de catálogos:

- Catálogo de tipos de RNFs (segurança, desempenho, entre outros);
- Catálogo de métodos de refinamento e técnicas de desenvolvimento que auxiliam no atendimento dos RNFs;
- Catálogo de interdependências implícitas, conhecidas como correlações entre *softgoals*.

2.2.2 Tipos de *Softgoals*

Os *softgoals* podem ser de três tipos: de RNFs, de operacionalização e de afirmação⁶. Sua representação gráfica encontra-se na Figura 5.

Figura 5 – Tipos de *softgoals* adaptados de Chung et al. (2000).

Os *softgoals* de RNFs representam os RNFs propriamente ditos e são apresentados graficamente por uma nuvem, conforme Figura 5 (a).

Os *softgoals* de operacionalização têm o objetivo de identificar estratégias de implementação resultantes da decomposição de um RNF. Estes *softgoals* são representados graficamente por uma nuvem com bordas mais grossas, conforme Figura 5 (b). As operacionalizações não são limitadas somente a funções e operações, e podem também corresponder a dados e restrições de projeto.

Nos dois casos, Tipo corresponde ao tipo de RNF e Tópico corresponde à estrutura relacionada aos requisitos funcionais. Detalhes sobre tipo e tópico são apresentados na seção 2.2.5 deste trabalho.

Os *softgoals* de afirmação têm o objetivo de auxiliar no registro de decisões e do motivo da escolha de uma alternativa de projeto, no registro de decisões sobre prioridades definidas para os *softgoals* ou meramente no registro de fatos e opiniões. A afirmação corresponde a um texto que é incluído pela equipe de projeto para expressar decisões, fatos e opiniões. *Softgoals* de afirmação são apresentados graficamente por nuvens tracejadas, conforme Figura 5(c).

⁶ O termo *softgoal* de afirmação será utilizado como tradução para *claim softgoal*.

2.2.3 Interdependência entre *Softgoals*

Os *softgoals* estão relacionados através de relações de interdependência, que podem ser de refinamento e de contribuição.

A interdependência de refinamento registra *softgoals* descendentes⁷ a partir de um *softgoal* ascendente⁸.

Existem três tipos de interdependências de refinamento: decomposição, operacionalização e argumentação. Suas representações gráficas estão apresentadas na Figura 6.

Figura 6 – Catálogo de tipos de refinamentos adaptado de Chung et al. (2000).

O refinamento de decomposição refina um *softgoal* em outro *softgoal* do mesmo tipo e pode ser aplicado ao RNF, à operacionalização, à argumentação e à priorização de um *softgoal*. O refinamento de operacionalização refina um *softgoal* em *softgoals* de operacionalização. O refinamento de argumentação registra as decisões de projeto através de *softgoals* de afirmação.

A interdependência de contribuição registra o nível de satisfação com que um *softgoal* descentente representa em direção a um *softgoal* ascendente. A interdependência de contribuição entre *softgoals* representa o grau com que um *softgoal* resultante do refinamento satisfaz um *softgoal* ascendente. Existem nove tipos de interdependências de contribuição, conforme apresentado no catálogo da Figura 7.

⁷ O termo descendente será utilizado neste trabalho como tradução da palavra *offspring*.

⁸ O termo ascendente será utilizado neste trabalho como tradução da palavra *parent*.

Figura 7 – Catálogo de tipos de contribuições adaptado de Chung et al. (2000).

O significado dos tipos de contribuições é apresentado a seguir:

- A contribuição E estabelece que, se todos os *softgoals* descendentes estiverem suficientemente satisfeitos, o *softgoal* ascendente também estará suficientemente satisfeito.
- A contribuição OU estabelece que se um dos *softgoals* descendentes estiver suficientemente satisfeito, o *softgoal* ascendente também estará suficientemente satisfeito.
- A contribuição *MAKE* é o nível mais alto de satisfação e estabelece uma contribuição suficientemente positiva entre o *softgoal* descendente e o *softgoal* ascendente, isto é, se o *softgoal* descendente for suficientemente satisfeito, o *softgoal* ascendente também será suficientemente satisfeito.
- A contribuição *HELP* estabelece uma contribuição parcialmente positiva entre o *softgoal* descendente e o *softgoal* ascendente, isto é, se o *softgoal* descendente for parcialmente satisfeito, o *softgoal* ascendente também será parcialmente satisfeito.
- A contribuição *SOME +* estabelece alguma contribuição positiva entre o *softgoal* descendente e o *softgoal* ascendente.
- A contribuição *EQUAL* estabelece que um *softgoal* descendente será suficientemente satisfeito ou negado, se e somente se, o *softgoal* de ascendente também for.
- A contribuição *SOME -* estabelece alguma contribuição negativa entre o *softgoal* descendente e o *softgoal* ascendente.

- A contribuição *HURT* estabelece uma contribuição parcialmente negativa entre o *softgoal* descendente e o *softgoal* ascendente, isto é, se o *softgoal* descendente for parcialmente negado, o *softgoal* ascendente será parcialmente negado.
- A contribuição *BREAK* é o nível mais alto de negação e estabelece uma contribuição suficientemente negativa entre o *softgoal* descendente e o *softgoal* ascendente, isto é, se o *softgoal* descendente estiver suficientemente satisfeito, o *softgoal* ascendente será negado.
- A contribuição *UNKNOWN* estabelece que existe alguma contribuição entre um *softgoal* descendente e um *softgoal* ascendente, porém, a contribuição pode ser positiva, negativa, suficientemente positiva ou suficientemente negativa.

2.2.4 Procedimento de Avaliação

O procedimento de avaliação determina o grau de satisfação dos RNFs. Dado um SIG, completo ou incompleto, o procedimento de avaliação determina se cada *softgoal* ou interdependência no SIG é suficientemente satisfeita. O resultado da avaliação é apresentado através da atribuição de um rótulo, que é o símbolo sobre o *softgoal*. Desta forma, um rótulo representa o grau de satisfação de um *softgoal* individualmente.

Decisões sobre aceitar ou rejeitar *softgoals* estabelecem um conjunto inicial de rótulos. Estas decisões são sempre tomadas de baixo para cima no SIG. O procedimento de avaliação utiliza estes rótulos para determinar o impacto das decisões nos *softgoals* ascendentes, para que ao final do processo, o *softgoal* do topo seja avaliado.

Os tipos de rótulos que podem ser atribuídos para um *softgoal* são: negado (X), fracamente negado (W-), indeterminado (U), fracamente satisfeito (W+), suficientemente satisfeito (✓) e conflitante (↯). A Figura 8 apresenta os tipos de rótulos.

Figura 8 – Possíveis rótulos dos *softgoals* adaptados de Chung et al. (2000).

O procedimento de avaliação é composto de dois passos. O primeiro passo determina o impacto individual de um *softgoal* descendente em um *softgoal* ascendente. O segundo passo tem o objetivo de combinar os rótulos individuais resultantes em um único rótulo que permita determinar o nível de satisfação de um RNF representado pelo SIG.

Têm-se, a seguir, as regras a serem consideradas na propagação de um rótulo de um *softgoal* descendente para um *softgoal* ascendente:

- **MAKE ($\checkmark$)**: propaga $\checkmark$ de um *softgoal* descendente para um *softgoal* ascendente
- **MAKE (X)**: propaga X de um *softgoal* descendente para um *softgoal* ascendente.
- **HELP (W^+)**: mantém o mesmo tipo de contribuição, porém a enfraquece, isto é, $\checkmark$ em um *softgoal* descendente é propagado como W^+ em um *softgoal* ascendente, e X em um *softgoal* descendente é propagado como W^- em um *softgoal* ascendente.
- **BREAK (X)**: inverte a contribuição de um *softgoal* descendente, $\checkmark$ para X, em um *softgoal* ascendente.
- **HURT (W^-)**: inverte a contribuição e a enfraquece: $\checkmark$ em um *softgoal* descendente é propagado como W^- no *softgoal* ascendente, e X em um *softgoal* descendente é propagado como W^+ em um *softgoal* ascendente.
- **SOME $^+$** : enfraquece a contribuição do *softgoal* descendente no *softgoal* ascendente. Um *softgoal* descendente com o rótulo $\checkmark$ ou W^+ resulta em W^+ para um *softgoal* ascendente; um *softgoal* descendente com X ou W^- resulta em W^- para um *softgoal* ascendente.
- **SOME $^-$** : inverte a contribuição do *softgoal* descendente no *softgoal* ascendente. Um *softgoal* descendente com $\checkmark$ ou W^+ resulta em W^- para

um *softgoal* ascendente; um *softgoal* descendente com X ou W^- resulta em W^+ para um *softgoal* ascendente.

- **UNKNOWN (U):** propaga U independentemente do tipo de contribuição.
- **CONFLICT ($\bowtie$):** propaga $\bowtie$ a não ser que o tipo de contribuição seja U.

Depois que todos os rótulos foram propagados de *softgoals* descendentes para os *softgoals* ascendentes, o segundo passo da avaliação tem o objetivo de combiná-los em um único rótulo (CHUNG et al., 2000).

Todos os possíveis rótulos resultantes são coletados para que sejam combinados. Os possíveis rótulos resultantes para combinação são: X , W^- , $\bowtie$, U , W^+ e $\surd$.

Contribuições positivas indicadas por um W^+ podem ser combinadas em um $\surd$. O mesmo ocorre com as contribuições W^- que podem ser combinadas em um X .

Os rótulos W^+ e W^- são primeiramente combinados pelo desenvolvedor em um ou mais rótulos $\surd$, X , $\bowtie$, e U . Tipicamente rótulos do tipo W^+ resultam em $\surd$ ou U , e W^- resultam em X ou U . A mistura de rótulos W^+ e W^- resultam tipicamente em $\surd$, X ou $\bowtie$.

O conjunto resultante é combinado em um único rótulo, pela escolha de um dos rótulos seguintes: $\bowtie \leq U \leq X \approx \surd$.

Se existirem $\surd$ e X para combinar, sem a presença de $SOME^+$ ou $SOME^-$, o resultado é conflitante ($\bowtie$) e a situação deve ser resolvida pelo desenvolvedor.

Na avaliação do segundo passo, algumas decisões podem ser tomadas automaticamente, sem intervenção do desenvolvedor; entretanto, o desenvolvedor pode utilizar sua experiência sobre os RNFs e o domínio, a fim de resolver conflitos e determinar os rótulos.

2.2.5 Métodos de Refinamento de *Softgoals*

Métodos de refinamento de *softgoals* (ou simplesmente métodos) são procedimentos que têm o objetivo de refinar um *softgoal* ou interdependência em um ou mais *softgoals* descendentes. Quando um conjunto de métodos é

coletado e catalogado, é possível oferecer vocabulário apropriado para manipular RNFs.

Na seção 2 foram definidos os três tipos de refinamentos para relacionar *softgoals* e, para cada um deles, existe um conjunto de métodos de refinamento. Estes métodos possuem o mesmo nome dos tipos e são: métodos de decomposição de RNF, métodos de operacionalização e métodos de argumentação.

O método de decomposição de RNFs registra o conhecimento sobre o detalhamento dos RNFs em requisitos mais específicos. O método de decomposição de operacionalização é utilizado para refinar *softgoals* em *softgoals* de operacionalização. *Softgoals* de operacionalização são candidatos para representação de entidades, atividades, restrições, ou mesmo possíveis componentes de projeto e implementação. O método de decomposição de argumentação auxilia na construção de uma estrutura de argumentação como parte do SIG. A forma mais simples de uso de uma argumentação é o uso de *softgoals* de afirmação relacionados aos *softgoals* do SIG.

Em relação à decomposição de RNF pode-se dizer que este detalhamento permite a coexistência de diferentes interpretações sobre um mesmo RNF, mesmo conflitantes. A Figura 9 apresenta o catálogo de método de decomposição de RNFs.

Figura 9 – Catálogo de decomposição de RNFs adaptado de Chung et al. (2000).

Os métodos de decomposição de RNFs específicos referem-se aos catálogos de RNFs definidos no *NFR Framework* para desempenho, segurança

e acurácia. Os métodos genéricos referem-se aos refinamentos de tipo e tópico.

Na decomposição genérica de tipo, os RNFs podem ser decompostos em sub-tipos de RNFs, enquanto na decomposição de tópico são detalhadas estruturas primitivas dos requisitos funcionais, arquiteturas de sistemas e software e arquiteturas de processos. Estas primitivas incluem classes, super-classes e sub-classes, atributos, visões, contextos, perspectivas, programas, pacotes, componentes, entre outros (CHUNG et al., 2000).

Chung et al. (2000) apresentam três tipos de métodos genéricos de decomposição de tópicos: por sub-classes, atributos e subconjunto.

Os métodos genéricos de decomposição por sub-classes estão exemplificados na Figura 10, para mostrar o controle de espaço de armazenamento para contas bancárias tipo regular e ouro.

Figura 10 – Método de decomposição de sub-classe aplicado ao controle de espaço para contas regulares e ouro adaptado de Chung et al. (2000).

A Figura 10 (a) apresenta a definição do método de decomposição por sub-classes. A Figura 10 (b) apresenta a hierarquia de sub-classes para a classe conta; esta hierarquia representa a estrutura funcional a ser utilizada no SIG. A Figura 10 (c) apresenta o SIG inicial para a classe conta, e a Figura 10 (d) apresenta o SIG resultante com o preenchimento da estrutura de sub-classes.

Os métodos genéricos de decomposição por atributos estão exemplificados na Figura 11, para mostrar o controle do tempo de resposta das contas ouro. A

Figura 11 (a) apresenta a definição do método de decomposição por atributos, a Figura 11 (b) apresenta os atributos relacionados à classe conta ouro, a Figura 11 (c) apresenta o SIG inicial para a classe conta, a Figura 11 (d) apresenta o SIG resultante com o preenchimento da estrutura de atributos.

Figura 11 – Método de decomposição de atributos aplicado ao controle de tempo de resposta de contas ouro adaptado de Chung et al. (2000).

O método de decomposição de sub-conjunto, segundo os autores, utiliza múltiplas decomposições, que devem ser selecionadas pelos desenvolvedores com base no domínio e na experiência.

2.2.6 Correlação entre *Softgoals*

Os RNFs de um software podem estar em conflito ou em harmonia entre si. Por exemplo, muitas vezes, o melhor tempo de resposta pode estar em conflito com custo de desenvolvimento. Mas, por outro lado, um sistema de software de melhor qualidade pode estar em harmonia com melhor tempo de resposta.

No tratamento de RNFs é importante que o conflito e a harmonia sejam identificados, para auxiliar nas análises de *tradeoff*, bem como nas análises de sinergia entre RNFs durante o desenvolvimento do sistema de software.

No NFR *Framework*, o conflito é representado pelas correlações negativas e a harmonia pelas correlações positivas. Estas correlações podem ser

identificadas através de catálogos de correlações (CHUNG et al., 2000) ou durante a elaboração dos SIGs, conforme novas interdependências são incluídas.

Normalmente, as correlações podem ser provenientes de mudanças nos SIGs, como por exemplo, a adição de um novo *softgoal*. Estas mudanças podem ser gatilhos para a identificação de uma nova correlação.

Desta forma, as interdependências geradas pelo resultado da aplicação de um método de refinamento de *softgoals* são chamadas de interdependências explícitas, enquanto que as interdependências provenientes de uma correlação são chamadas de interdependências implícitas.

A detecção das correlações pode ser feita pela comparação de partes do SIG com padrões definidos nos catálogos de correlações (CHUNG et al., 2000). Fica a critério do desenvolvedor decidir se as interdependências detectadas serão ou não incorporadas ao SIG. A Figura 12 apresenta um exemplo de SIG que possui correlações positivas e negativas.

Figura 12 – Exemplo de correlações positivas e negativas adaptado de Chung et al. (2000).

No exemplo da Figura 12, o uso do *softgoal* de operacionalização *Formato Comprimido* causa deterioração do *softgoal* de tempo de resposta, porque a informação precisa ser descomprimida para depois ser recuperada. Este relacionamento pode ser expresso pela seguinte regra de correlação: *Formato Comprimido [Informação] HURTS Tempo de Resposta [Informação]*, onde informação é um item de informação como uma classe, ou um atributo de classe.

Ainda no exemplo da Figura 12, o *softgoal* de operacionalização *Executar Primeiro* contribui positivamente para a acurácia das contas ouro, que são de

execução prioritária. A operacionalização *Validação* contribui positivamente para o *softgoal* de confiabilidade e negativamente para o *softgoal* de tempo de resposta.

2.2.7 Construção do SIG

O artefato central do NFR *Framework* é o SIG que permite representar e registrar decisões de projeto. Chung et al. (2000) definem um conjunto de dez passos para elaboração deste grafo:

1. Aquisição do conhecimento do domínio;
2. Aquisição e catalogação do conhecimento sobre os RNFs;
3. Identificação dos RNFs do sistema de software;
4. Decomposição dos *softgoals* dos SIGs iniciais;
5. Definição de prioridades dos *softgoals*;
6. Identificação de possíveis operacionalizações;
7. Definição das interdependências implícitas entre *softgoals*;
8. Registro de decisões de projeto;
9. Seleção de alternativas de projeto;
10. Avaliação do impacto das decisões.

Os passos listados não precisam ser realizados necessariamente na seqüência apresentada; a seqüência a ser adotada depende das características de um sistema de software a ser desenvolvido.

Durante a aquisição do conhecimento do domínio, o desenvolvedor deve formalizar o seu conhecimento relativo ao domínio do sistema de software a ser desenvolvido e identificar as prioridades que podem influenciar o desenvolvimento. A documentação do domínio deve fazer parte da especificação dos requisitos de software. As prioridades podem originar-se de considerações sobre diversos fatores, tais como as prioridades organizacionais (exemplo: um banco pode considerar que a segurança de informação é muito importante) e a carga de trabalho organizacional (exemplo: um sistema de cartão de crédito pode ter um grande número de vendas para autorizar em um dia). Além disso, pode-se priorizar os requisitos durante o desenvolvimento.

Durante a aquisição e a catalogação do conhecimento sobre os RNFs, o desenvolvedor deve tomar conhecimento dos catálogos do NFR *Framework* relacionados aos requisitos não-funcionais, métodos, relacionamentos e correlações, para poder desenvolver o sistema de software de forma aderente ao NFR *Framework*. Estes catálogos determinam a terminologia e a classificação dos conceitos a serem utilizados e podem ser estendidos conforme a necessidade.

Durante a identificação dos RNFs do sistema de software são definidos os SIGs iniciais que representam os principais RNFs do sistema de software. São representados os tipos de requisitos (desempenho, segurança, etc), e tópicos relacionados. A Figura 13 apresenta um exemplo de SIGs iniciais para contas bancárias.

Figura 13 – SIGs iniciais de desempenho e segurança adaptados de Chung et al. (2000).

Durante a decomposição dos *softgoals* dos SIGs iniciais, o desenvolvedor pode optar pela decomposição de tipo e/ou tópico. No exemplo da Figura 14, foi escolhida a decomposição de tipo de RNFs.

Figura 14 – Decomposição de tipo dos *softgoals* de desempenho e segurança adaptado de Chung et al. (2000).

Durante a definição de prioridades dos *softgoals*, o desenvolvedor estabelece os pontos relevantes do SIG, introduzindo *softgoals* que indicam a importância do ponto de vista de prioridade. Estes dados podem ser úteis para

análises de *tradeoff* entre *softgoals*. A Figura 15 mostra um exemplo de prioridade definido para a acurácia das contas bancárias.

Figura 15 – Definição de prioridade para a acurácia de contas adaptada de Chung et al. (2000).

Durante a identificação de possíveis *softgoals* de operacionalização, os RNFs devem estar suficientemente refinados, para que os desenvolvedores definam as estratégias de projeto a serem utilizadas. O exemplo da Figura 16 apresenta os *softgoals* de operacionalização identificados para os RNFs de desempenho e segurança.

Figura 16 – Exemplos de operacionalizações dos *softgoals* de desempenho e segurança adaptados de Chung et al. (2000).

Neste exemplo, as nuvens com bordas mais grossas são *softgoals* de operacionalização, que representam possíveis alternativas de projeto identificadas para os RNFs de desempenho e segurança.

Durante a identificação de interdependências implícitas entre *softgoals*, os desenvolvedores devem consultar os catálogos de correlações do NFR *Framework* ou avaliar as possíveis interdependências, positivas e negativas, entre os *softgoals*, com base na experiência a respeito dos RNFs. A Figura 17 apresenta exemplos das interdependências implícitas identificadas para os *softgoals* de desempenho e segurança.

Figura 17 – Definição de interdependências implícitas adaptada de Chung et al. (2000).

No exemplo da Figura 17 foram identificadas correlações negativas entre o uso de formato descomprimido e o *softgoal* de espaço, entre a validação de acesso contra regras de elegibilidade e o *softgoal* de tempo de resposta, a requisição de identificação adicional e o *softgoal* de usabilidade. Uma correlação positiva foi detectada entre a validação de acesso contra regras de elegibilidade e o *softgoal* de acurácia.

Durante o registro de decisões de projeto, decisões, fatos, opiniões e resultados de análises de *tradeoffs* devem ser registrados através de *softgoals* de afirmação. Um exemplo é apresentado visualizado na Figura 18.

Figura 18 – Registro de *softgoals* de afirmação adaptado de Chung et al. (2000).

Os *softgoals* de afirmação, neste exemplo, estão associados à validação de acesso contra regras de elegibilidade, em que o desenvolvedor afirma que uma validação de acesso otimizada não irá interferir no tempo de resposta. O *softgoal* de afirmação relacionado a acurácia enfatiza que este requisito é vital para o sistema de software.

Durante a seleção de alternativas de projeto, o desenvolvedor toma decisões relativas às alternativas de projeto que foram identificadas nos passos anteriores, considerando informações do domínio, prioridades, decisões de projeto, interdependências entre *softgoals* e o conhecimento registrado em catálogos. A Figura 19 ilustra os *softgoals* de operacionalização selecionados (✓) e rejeitados (X).

Após conclusão dos dez passos devem ser definidos os relacionamentos de meta de operacionalização, o texto das metas, e o relacionamento entre os requisitos funcionais e as decisões tomadas. A Figura 21 exemplifica estes relacionamentos.

Figura 21 – Relacionamento com o requisito funcional adaptado de Chung et al. (2000).

A caixa de nota representa a meta a ser atingida. A meta deve ser conectada a todas as operacionalizações relacionadas a ela através dos relacionamentos de meta de operacionalização. O requisito funcional *Manter Contas* é relacionado à meta através do relacionamento de decisão de projeto.

2.3 Requisitos Não-Funcionais no modelo de casos de uso

A abordagem baseada em casos de uso é utilizada para descrever requisitos funcionais externamente visíveis de um sistema de software (SCHNEIDER; WINTERS, 1997). Leffingwell e Widrig (2001) destacam que o método de casos de uso apresenta uma forma para descrever o comportamento do sistema de software sob a perspectiva dos usuários. Este método faz com que seja possível explorar o comportamento do sistema de software em suas fases iniciais de desenvolvimento pelo envolvimento do usuário.

Esta abordagem foi utilizada por diversos autores para ser complementada com os resultados do tratamento dos RNFs. Esta seção apresenta os trabalhos mais significativos encontrados na literatura (JACOBSON; BOOCH; RUMBAUGH, 1999), (JACOBSON; NG, 2004), (SUPAKKUL; CHUNG, 2004), (RATIONAL UNIVERSITY, 2003).

2.3.1 Casos de Uso

Os casos de uso oferecem uma maneira sistemática de capturar requisitos funcionais, provenientes de questões de valor na visão de cada tipo de usuário do sistema de software. A utilização de casos de uso faz com que o analista defina o sistema de software nos termos das necessidades dos usuários e suas necessidades de negócio perante o sistema de software a ser desenvolvido (JACOBSON; BOOCH; RUMBAUGH, 1999).

A Figura 22 apresenta o fluxo de trabalho do desenvolvimento baseado em casos de uso do RUP (*Rational Unified Process*) (JACOBSON; BOOCH; RUMBAUGH, 1999).

Figura 22 – Fluxo de trabalho para capturar requisitos como casos de uso adaptado de Jacobson, Booch e Rumbaugh (1999).

A atividade inicial é a de identificar os atores e dos casos de uso e tem os objetivos de (1) delimitar a fronteira entre o sistema de software e o ambiente; (2) identificar os atores que interagem com o sistema de software; (3) identificar

as funções esperadas do sistema de software e (4) capturar o glossário de termos comuns que são essenciais para detalhar a funcionalidade do sistema de software.

A atividade para priorizar casos de uso tem o objetivo de planejar o desenvolvimento das iterações, em função da importância dos casos de uso com base em critérios estabelecidos para o desenvolvimento.

A atividade para detalhar casos de uso tem o objetivo de especificar os fluxos de eventos de cada caso de uso em detalhe. A Tabela 1 apresenta o modelo com os campos da especificação conforme Rational University (2003).

Tabela 1 – Modelo de especificação de casos de uso adaptado de Rational University (2003).

Item da Especificação	Descrição
Nome do caso de uso	Nome do caso de uso.
Descrição breve	Descrição resumida da função do caso de uso.
Atores	Lista dos atores envolvidos no caso de uso.
Evento Inicial	Evento iniciador do caso de uso.
Fluxo Básico	Descrição do fluxo principal com os passos numerados seqüencialmente.
Fluxos Alternativos	Descrição dos fluxos alternativos com os passos numerados seqüencialmente.
Sub-fluxos	Descrição dos sub-fluxos com os passos numerados seqüencialmente.
Requisitos Especiais	Requisitos não-funcionais aplicáveis ao caso de uso.
Pré-condição	Condição verdadeira antes da execução do caso de uso.
Pós-condição	Condição verdadeira após execução do caso de uso finalizar.
Pontos de Extensão	Lista dos pontos de extensão do caso de uso.
Pontos de Inclusão	Lista dos pontos de inclusão no caso de uso.

A descrição de casos de uso prevê a inclusão de RNFs no campo de requisitos especiais (JACOBSON; BOOCH; RUMBAUGH, 1999).

A elaboração da descrição de casos de uso não prevê um tratamento sistemático dos RNFs na elaboração dos demais modelos UML. Uma das formas para garantir a inserção de um requisito não-funcional na descrição de casos de uso é identificar todos os casos de uso que sofrem impacto de cada requisito não-funcional especificado para o sistema de software. Geralmente esta prática é manual e, muitas vezes, não considera que os requisitos não-funcionais podem ser conflitantes entre si, como foi visto na seção 2.2.6.

A atividade para elaborar protótipo de interface com o usuário tem o objetivo de criar um conjunto de esboços de interface do usuário e protótipos de

interface que permitam aos principais atores do sistema de software avaliarem os casos de uso.

A atividade para estruturar o modelo de casos de uso tem o objetivo de extrair descrições gerais, compartilhadas, adicionais ou opcionais das especificações de casos de uso originais, a fim de minimizar a redundância. Estas descrições podem ser extraídas e descritas em casos de uso separados que podem ser reutilizados pelos casos de uso originais.

Os casos de uso podem estar relacionados através de generalização, extensão e inclusão.

O relacionamento de generalização é utilizado com a finalidade de disponibilizar comportamentos parcialmente definidos em casos de uso conhecidos como casos de uso abstratos. Casos de uso concretos reusam o comportamento existente nos casos de uso abstratos, porém, são constituídos pelas seqüências completas de ações a serem executadas pelo sistema de software. Um caso de uso abstrato não é iniciado por um ator, somente instâncias de casos de uso concretos. Este relacionamento somente está disponível em versões anteriores da UML 2 (OMG-UML, 2009).

O relacionamento de extensão representa uma descrição adicional e/ou opcional a ser executada pelo caso de uso base, sob determinadas condições especificadas. Se a condição definida pelo ponto de extensão do caso de uso base for atendida, o fluxo de execução é transferido para o caso de uso estendido.

O relacionamento de inclusão ocorre quando um mesmo comportamento é compartilhado entre vários casos de uso. Nestes casos, estes fluxos comuns, podem ser separados como casos de uso de inclusão.

A proposta de Jacobson e Ng (2004) apresenta uma abordagem de casos de uso para desenvolvimento de software orientado a aspectos. Os RNFs, nesta proposta, são tratados como casos de uso de infra-estrutura que lidam com questões relacionadas, por exemplo, à plataforma com o uso de *middlewares*. Esta abordagem não trata os RNFs de maneira detalhada em suas fases iniciais.

2.3.2 Abordagem Orientada a *Softgoals* e Casos de Uso

Vários trabalhos propuseram abordagens com a finalidade de integrar visões funcional e não-funcional, utilizando o NFR *Framework*.

Inicialmente Cysneiros (2001) e Cysneiros, Leite e Neto (2001) propuseram uma abordagem para combinar o NFR *Framework* ao modelo de classes orientado a objetos. Os modelos de classes e os requisitos não-funcionais são detalhados separadamente, e posteriormente combinados a fim de satisfazer os RNFs de um sistema de software.

Supakkul e Chung (2004) fizeram uma proposta baseada em *softgoals* e casos de uso a fim de integrar a representação utilizada no NFR *Framework* e as representações mais comumente utilizadas, como a UML. O principal objetivo é fazer com que as partes interessadas tenham uma visão integrada dos requisitos funcionais e não-funcionais de sistemas de software.

O foco principal do método de Supakkul e Chung (2004) é definir os elementos do modelo de casos de uso que são afetados pelos RNFs especificados para o sistema de software.

A associação entre os elementos do modelo de casos de uso e os requisitos não-funcionais é realizada através dos pontos de associação definidos no diagrama de casos de uso. São quatro tipos de pontos de associação: limite do sistema, caso de uso, ator e relação ator-caso de uso, conforme apresentado na Figura 23.

Figura 23 – Pontos de associação entre *softgoals* e elementos do diagrama de casos de uso.

O ponto de associação com o limite do sistema de software significa que um RNF é global, ou seja, aplicável ao sistema como um todo e não no contexto de um elemento específico do diagrama de casos de uso (Figura 23 (a)). O requisito de portabilidade é um exemplo de um requisito não-funcional de abrangência global. Este tipo de ponto de associação é o único que não é conectado explicitamente à representação de limite do sistema. Isso ocorre pelo fato de que a representação do limite do sistema não é um elemento definido pela linguagem UML.

O ponto de associação com um caso de uso indica que o RNF afeta um caso de uso específico (Figura 23 (b)). Um exemplo é o tempo de resposta aceitável para a execução de um determinado caso de uso.

O ponto de associação com um ator indica que o RNF afeta uma entidade externa (Figura 23 (c)). Um exemplo é a escalabilidade que indica que o sistema deve estar disponível para o ator mesmo quando existir grande quantidade de acessos por diversos usuários diferentes.

O ponto de associação com a relação ator – caso de uso indica que o RNF se refere às características da interface entre o ator e o sistema, tais como sistema de acesso, comunicação, interface entre usuário e sistema de software, troca de informação entre o sistema de software e os atores (Figura 23 (d)). Um exemplo é *security* associada entre o ator e o caso de uso para indicar que um caso de uso precisa de controle de acesso.

Com o objetivo de incluir os RNFs nos múltiplos casos de uso, o engenheiro de requisitos precisa associar o RNF a um elemento mais geral do diagrama de casos de uso para que este requisito possa ser propagado aos seus elementos especializados. Em outras palavras, pode-se dizer que um RNF é propagado de um elemento generalizado para seus elementos especializados. Entretanto, esta propriedade não é aplicável para um ponto de associação com o limite do sistema de software, pois este requisito não está diretamente associado a nenhum elemento generalizado ou especializado do diagrama de casos de uso (SUPAKKUL; CHUNG, 2004).

A fim de executar as análises da propagação de um RNF, o método de Supakkul e Chung (2004) propôs as regras de propagação, que estão apresentadas na Figura 24. Estas regras foram detalhadas formalmente em Chung e Supakkul (2004).

Figura 24 – Propagações de RNFs adaptadas de Supakkul e Chung (2004).

A regra de propagação de ator define que um RNF associado a um ator é também propagado para todas as especializações de atores que estiverem direta ou indiretamente associadas ao ator generalizado. No exemplo da Figura 24 (a), o requisito não-funcional associado ao ator 2 é propagado para o ator 3 e ator 4, porém não propagado para o ator 1, que não é uma especialização do ator 2.

A regra de propagação de caso de uso define que um RNF associado a um caso de uso é propagado para casos de uso que são especializados a partir dele ou incluídos por ele. No exemplo da Figura 24 (b), o RNF associado ao caso de uso 3 é propagado para os casos de uso 4 e caso de uso 6 (casos de uso especializados). O RNF é também propagado para os casos de uso 7 e 5 (casos de uso incluídos). É importante ressaltar que o relacionamento de generalização somente existe em versões anteriores à UML 2 (OMG-UML, 2009).

A regra de propagação de relação ator-caso de uso define que um RNF associado a uma relação ator-caso de uso é também aplicável para todas as especializações de atores, com seus respectivos casos de uso, que estiverem direta ou indiretamente associados ao ator generalizado. No exemplo da Figura 24 (c) o RNF é propagado para os atores 8 e 9 e seus casos de uso 11 e 12 respectivamente.

2.4 Análise e Validação de Requisitos

A análise e validação de requisitos são fases típicas do processo de engenharia de requisitos como, por exemplo, no processo proposto por Kotonya e Sommerville (1998). Os requisitos estabelecidos para os sistemas de software, através da análise, devem ser validados, ou seja, confirmados, para que o sistema de software possa ser desenvolvido a partir destes requisitos, que satisfaçam a necessidade das partes interessadas.

O objetivo da análise de requisitos é verificar se os requisitos contribuem para a solução adotada de um sistema e software de maneira completa (KOTONYA; SOMMERVILLE, 1998). Além disso, a análise de requisitos resulta na identificação de requisitos desnecessários, conflitantes ou mesmo inviáveis para a solução.

Para realizar a análise de requisitos, pode-se utilizar recursos como listas de verificação e matrizes de interação de requisitos (KOTONYA; SOMMERVILLE, 1998).

As listas de verificação relacionam os pontos que devem ser verificados durante a análise de requisitos. Estes pontos abordam temas importantes, tais como: utilização de padrões de software e hardware, verificação da ambigüidade de requisitos, definição de requisitos não-funcionais, existência de requisitos desnecessários, entre outras questões. As listas de verificação servem para avaliar se nenhum ponto relacionado à análise dos requisitos foi esquecido, além de sinalizar problemas em potencial.

As matrizes de interações entre requisitos, por sua vez, têm o objetivo de identificar requisitos conflitantes e sobrepostos.

Os conflitos detectados entre os requisitos podem ser resolvidos através de reuniões entre as partes interessadas (KOTONYA; SOMMERVILLE, 1998).

A validação de requisitos tem como objetivo certificar que os requisitos provenientes da fase de análise representam uma descrição aceitável do sistema de software que será desenvolvido. Esta fase deve envolver as partes interessadas, que devem verificar se os problemas dos requisitos, omissões e ambigüidades ainda permanecem durante a validação (KOTONYA; SOMMERVILLE, 1998).

Como exemplos de problemas detectados na fase de validação, podem-se citar: não-conformidade do documento de requisitos perante padrões da organização, ambigüidade entre requisitos, erros nos modelos do sistema de software, conflitos entre requisitos que ainda não foram detectados na fase de análise.

Um dos maiores problemas no processo de validação é a inexistência de um documento que possa servir como fonte de requisitos. A validação dos requisitos deve assegurar que o documento de especificação de requisitos apresente uma descrição clara do sistema de software para que possa ser seguida nas próximas fases do desenvolvimento, além de uma confirmação final de que os requisitos atendem às necessidades das partes interessadas.

A validação de requisitos pode ser um processo prolongado que envolve pessoas, leituras, análises de documentação e reuniões. É possível que dure semanas ou mesmo meses para que se valide os requisitos de um sistema complexo, principalmente quando muitas partes interessadas estão envolvidas. Se esta fase não for realizada, o retrabalho é inevitável, pois as deficiências podem não ser detectadas em fases posteriores. A solução de problemas decorrentes de erros de requisitos, em sistemas já terminados, pode custar até 100 vezes mais em relação ao que foi previsto na estimativa inicial (KOTONYA; SOMMERVILLE, 1998).

Existem vários métodos que podem ser utilizados como base para a validação de requisitos; entre eles podem-se citar a revisão de requisitos, aplicação de listas de verificação, prototipação com cenários, elaboração de versão preliminar de manual de usuário e modelagem da análise do sistema através de diagramas.

A revisão de requisitos é o método mais utilizado para validar requisitos (KOTONYA; SOMMERVILLE, 1998). O objetivo é envolver um grupo de pessoas que possa ler o documento e analisar os requisitos, procurar problemas, realizar reuniões para discutir e gerar uma lista de ações para resolver os problemas identificados.

As listas de verificação devem conter as características e os erros mais freqüentes cometidos nas especificações. A verificação é uma inspeção, em que cada item da lista de verificação é um item a ser analisado.

A prototipação pode ser utilizada tanto para a elicitação de requisitos quanto para a validação. Os protótipos para executar a validação devem ser mais completos que os protótipos da elicitação, pois além de implementar os requisitos, devem dar uma idéia do uso prático sistema através da execução de cenários. Os protótipos utilizados na fase de validação deverão ser utilizados posteriormente para desenvolver o sistema.

A elaboração de uma versão preliminar do manual do usuário causa a reescrita de requisitos de forma diferente da inicial, permitindo entender melhor os requisitos, além de detectar o relacionamento entre eles. Este manual deve incluir informações sobre a funcionalidade a ser desenvolvida e as partes do sistema de software que não serão desenvolvidas, uma descrição sobre como enfrentar dificuldades na implantação de um sistema novo, além das instruções sobre como instalar os protótipos e o próprio sistema final. A vantagem da elaboração do manual do usuário é que este documento será a base para a elaboração da documentação final do usuário.

A técnica de validação dos modelos do sistema possui três objetivos principais: (1) mostrar que um modelo em particular é consistente e que não conflita com outras partes do sistema; (2) mostrar que os modelos são consistentes internamente e externamente; (3) mostrar que o modelo reflete os requisitos das partes interessadas. Um ponto importante é que os modelos devem ser descritos também em linguagem natural, pois muitas partes interessadas podem não entender a notação utilizada e ter dificuldade em identificar os requisitos.

2.5 Considerações Finais

As classificações dos RNFs utilizadas neste trabalho são as propostas por Sommerville e Kotonya (1998) com o agrupamento em requisitos externos, requisitos de processo e requisitos de produto, e pela norma ISO 9126 (ISO/IEC, 2001) com seu conjunto de características e sub-características de qualidade.

A opção pelo uso do método baseado em casos de uso justifica-se por ser um método consolidado e reconhecido no mercado. A opção pelo método

baseado no NFR *Framework* justifica-se por ser ele o método mais referenciado e aplicado no meio acadêmico para tratamento de requisitos não-funcionais. Além disso, o NFR *Framework* possui diretrizes que auxiliam a determinação de estratégias de projeto, desde as fases iniciais do desenvolvimento de sistemas de software.

A proposta de Supakkul e Chung (2004) foi selecionada por apresentar a integração entre o modelo de casos de uso e o NFR *Framework*, preservando as definições originais dos métodos de casos de uso e NFR *Framework*.

As representações de casos de uso utilizadas neste trabalho são compatíveis com a UML 2 (*Unified Modeling Language*) (OMG-UML, 2009), porém, situações em que definições relacionadas a versões anteriores forem aplicáveis são destacadas, quando apropriado.

O padrão apresentado em Rational University (2003) foi escolhido para a especificação dos casos de uso neste trabalho por ser considerado completo e abrangente.

Foi utilizada a ferramenta StarUML (LEE et al., 2005) para auxiliar na representação dos diagramas de casos de uso e SIGs. A ferramenta executa automaticamente o primeiro passo da avaliação do NFR *Framework* e permite a intervenção do desenvolvedor no segundo passo da avaliação. A ferramenta StarUML (LEE et al., 2005) não possui representações para metas e relacionamento de meta de operacionalização. Nestes casos serão utilizadas as representações de nota e relacionamento de nota com esta finalidade.

A análise de requisitos foi feita no PARNAFOA através do uso do RNF *Framework*, proposto por Chung et al. (2000). Este método dá subsídios para que os requisitos sejam avaliados individualmente e conjuntamente.

A validação de requisitos foi realizada através da revisão dos documentos de requisitos pela equipe que conduziu as aplicações do PARNAFOA.

3. Engenharia de Requisitos Orientada a Aspectos

“É como a semente colocada no solo – quanto mais se semeia, melhor a colheita.” – Orison Swett Marden

Este capítulo tem o objetivo de apresentar os conceitos relacionados à engenharia de requisitos com foco na orientação a aspectos e os trabalhos relevantes selecionados da literatura.

3.1 Conceitos e Definições

Conforme já definido na seção 1.1 de motivação deste trabalho, interesses constituem um conjunto de requisitos relacionados (MILI; ELKHARRAZ; MCHEICK, 2004), ou qualquer coisa de interesse de alguma parte interessada, seja um usuário final, um responsável de projeto, ou mesmo um desenvolvedor (JACOBSON; NG, 2004). Pode também representar algum tipo de funcionalidade do sistema de software, uma característica ou um tipo de processamento (CLARKE; BANIASSAD, 2004).

Aspectos, por sua vez, são unidades criadas com a finalidade de modularizar interesses transversais (BERG; CONEJERO; CHITCHYAN, 2005), (RASHID et al, 2002). A modularização é o particinamento de artefatos em entidades chamadas módulos (BERG; CONEJERO; CHITCHYAN, 2005).

A utilização de aspectos auxilia na modularização de interesses transversais funcionais, não-funcionais e especificidades de plataforma, mantendo-os mais facilmente identificáveis no sistema de software (JACOBSON; NG, 2004). A identificação de aspectos em sistemas de software resulta em melhor suporte para a modularização de interesses transversais, reduzindo os custos de desenvolvimento, manutenção e evolução (RASHID et al., 2002).

Os sistemas de software modernos ficam susceptíveis à alta volatilidade dos ambientes em que estão inseridos e às constantes mudanças das regras de negócio e precisam se adaptar e evoluir rapidamente. Caso os interesses transversais não sejam manipulados adequadamente, a adaptabilidade dos

sistemas de software pode ser prejudicada. Desta forma, é essencial pensar em interesses transversais tão cedo quanto possível (RASHID et al., 2002).

Neste contexto, a engenharia de requisitos orientada a aspectos surgiu da necessidade do tratamento de interesses transversais desde a fase de requisitos (CHITCHYAN et al., 2006).

As primeiras propostas relacionadas à engenharia de requisitos orientada a aspectos podem ser encontradas em Rashid et al. (2002), Rashid, Moreira e Araújo (2003), Moreira, Araújo e Rashid (2005) e Clark e Baniassad (2005). Uma síntese dos trabalhos relacionados com engenharia de requisitos, com foco na orientação a aspectos, pode também ser encontrada em Chitchyan et al. (2005). Uma síntese de trabalhos que tratam especificamente da identificação de interesses não-funcionais em trabalhos relacionados à engenharia de requisitos orientada a aspectos pode ser encontrada em Bombonatti e Melnikoff (2010).

Grande parte dos trabalhos relacionados com a engenharia de requisitos orientada a aspectos utiliza os termos *early aspects* e *candidate aspects*. O termo *early aspect*, ou aspecto antecipado refere-se à identificação de um aspecto antes da fase de implementação, em qualquer iteração do desenvolvimento de um sistema de software (BANIASSAD et al., 2006). O termo *candidate aspect*, ou aspecto candidato, é utilizado para representar um aspecto antecipado que pode ou não ser modelado como um aspecto nas fases posteriores do desenvolvimento do sistema de software (DUAN et al., 2006). O termo requisitos aspectuais também é utilizado em alguns trabalhos da área para referir-se aos aspectos identificados na fase inicial de requisitos, porém, este termo ainda não está consolidado no meio acadêmico (NIU; EASTERBROOK; YU, 2007).

Chitchyan et al. (2005) consideram que uma abordagem relacionada com a engenharia de requisitos orientada a aspectos deve tratar da identificação de interesses transversais e não-transversais funcionais e não-funcionais, resolução de *trade-off* entre interesses e rastreabilidade de requisitos. Seus resultados podem ser utilizados em fases posteriores do desenvolvimento, verificação de requisitos pelas partes interessadas, validação de requisitos para garantir o entendimento do sistema de software a ser desenvolvido, e homogeneidade no tratamento de interesses funcionais e não-funcionais.

3.2 Separação e Composição de Interesses

A separação de interesses é um dos princípios básicos da engenharia de software (ROSENHAINER, 2004), o qual propõe o encapsulamento de características em entidades separadas para confinar mudanças e manipular um ponto importante por vez (DIJKSTRA, 1976 apud RASHID al., 2002).

O tratamento de interesses consiste em solucionar problemas, primeiro em partes separadas (separação de interesses), e depois combinar as soluções parciais para que sejam compostas (composição de interesses), fazendo com que a solução obtida seja próxima da ideal (MILI; ELKHARRAZ; MCHEICK, 2004).

A separação de interesses pode trazer muitos benefícios, incluindo a melhoria da reusabilidade, gerenciamento da complexidade e maior facilidade nas evoluções dos sistemas de software. A adequada separação de interesses, por sua vez, depende dos requisitos do sistema de software e do tipo de metodologia selecionada para separar e compor interesses (TARR et al., 1999).

Neste contexto, a separação de interesses pode levar ao que se chama de tirania da decomposição dominante. A decomposição dominante ocorre quando se utiliza apenas uma dimensão dominante como base para a decomposição. Por exemplo, o NFR *Framework* utiliza apenas requisitos não-funcionais como dimensão dominante, e a dimensão funcional é adicionada posteriormente. Outras abordagens como pontos de vista e casos de uso utilizam requisitos funcionais como decomposição dominante (TARR et al., 1999), (MOREIRA; ARAÚJO; RASHID, 2005).

Para contornar esta inconveniência, abordagens de separação multidimensional de interesses foram propostas a fim de permitir que múltiplas dimensões pudessem ser consideradas durante a decomposição e composição de interesses. Exemplos destas abordagens são os N graus de separação (TARR et al., 1999) e a separação multidimensional de interesses na engenharia de requisitos (MOREIRA; RASHID; ARAÚJO, 2005). Uma ontologia relacionada à separação multidimensional de interesses na engenharia de requisitos foi proposta por Alves (2010).

A composição de interesses, por sua vez, deve ser realizada a partir dos resultados obtidos pelo tratamento de interesses já separados. O objetivo da composição é identificar pontos bem definidos nos interesses do sistema de software, onde outros comportamentos provenientes de interesses transversais (aspectos) ou não-transversais possam ser adicionados (BRITO et al., 2007).

A composição de interesses faz referência a alguns termos provenientes da programação orientada a aspectos, como *joinpoint*, ou ponto de junção, *pointcut* ou ponto de intersecção, *advice* e *weaving* ou combinação (KICZALES et al., 1997). As traduções dos termos em inglês foram baseadas em Piveta et al. (2004), sendo que nenhuma tradução para a palavra *advice* é sugerida.

Um ponto de junção é um ponto de interesse específico em algum artefato, onde dois ou mais interesses podem ser compostos (BERG; CONEJERO; CHITCHYAN, 2005). Um ponto de intersecção é um predicado que integra pontos de junção (BERG; CONEJERO; CHITCHYAN, 2005). Um *advice* é o comportamento a ser executado no ponto de junção (BERG; CONEJERO; CHITCHYAN, 2005). A combinação refere-se à composição de aspectos com outros interesses do sistema de software (BERG; CONEJERO; CHITCHYAN, 2005).

Um exemplo simplificado, adaptado de Baniassad et al. (2006), utiliza esses conceitos durante a composição e pode ser observado na Figura 25.

Interesses:	Composição:
<p>1Δ. Pagar juros de uma certa porcentagem em cada conta.</p> <p>2Δ. Permitir clientes retirem dinheiro de suas contas.</p> <p>3Δ. Ter certeza de que todas as transações foram completadas.</p> <p>4Δ. Manter histórico de todas as transações.</p> <p>(a)</p>	<p>1) <u>Mesclar</u> {1Δ, 2Δ} com 3Δ ou <u>Mesclar pagar juros e retirem dinheiro</u> com “transações completas”.</p> <p>2) <u>Mesclar</u> {1Δ, 2Δ} com 4Δ ou <u>Mesclar pagar juros e retirem dinheiro</u> com “manter histórico”.</p> <p>(b)</p>

Figura 25 – Composição de interesses adaptada de Baniassad et al. (2006).

A Figura 25 (a) apresenta a lista de interesses do sistema de software e a Figura 25 (b), as regras de composição. Os interesses 1Δ e 2Δ foram compostos com o interesse 3Δ, na composição 1 e ao 4Δ, na composição 2. O

texto em *itálico* representa os pontos de intersecção, o texto sublinhado representa os tipos de *advices*, e o texto entre aspas representa o comportamento dos *advices*.

Durante o processo de composição, conflitos podem ocorrer nos pontos de junção, pois mais de um interesse pode precisar ser integrado no mesmo ponto. Os tratamentos de conflitos são considerados na maior parte dos métodos relacionados à engenharia de requisitos orientada a aspectos através de técnicas que vão desde a negociação de prioridade com partes interessadas até o uso de um processo hierárquico analítico, ou AHP (*Analytical Hierarchical Process*). O AHP é um método que classifica alternativas para tomada de decisão com base em um conjunto de critérios. Este método utiliza um processo matemático rigoroso para dar suporte à priorização de alternativas que servem como base para tomada de decisão (BRITO et al., 2007). Formulações matemáticas para resolução de conflitos na engenharia de requisitos orientada a aspectos podem ser encontradas em Sardinha et al. (2010).

3.3 Trabalhos Relevantes da Engenharia de Requisitos Orientada a Aspectos

Esta seção tem a finalidade de apresentar os trabalhos relevantes da engenharia de requisitos orientada a aspectos.

3.3.1 Modelo de Engenharia de Requisitos Orientado a Aspectos com UML

O modelo de engenharia de requisitos orientado a aspectos foi proposto inicialmente por Rashid et al. (2002) e, posteriormente, foi adaptado com o uso do modelo de casos de uso e diagramas de seqüência em UML, por Araújo et al. (2002). Este modelo tem a finalidade de separar interesses transversais não-funcionais na fase de requisitos e posteriormente compô-los utilizando o modelo de casos de usos (RASHID et al., 2002). O modelo de requisitos orientado a aspectos com UML é apresentado na Figura 26.

Figura 26 – Modelo de requisitos orientados a aspectos de Araújo et al. (2002).

Com base nos requisitos do sistema de software a ser desenvolvido, os interesses não-funcionais são elicitados a partir de discussões com as partes interessadas. Paralelamente, os requisitos funcionais são especificados através de diagramas de casos de uso e diagramas de seqüência.

Os interesses não-funcionais são analisados e, caso afetem mais de um caso de uso, são especificados conforme formulário da Tabela 2.

Tabela 2 – Formulário de especificação de um interesse transversal.

Interesse Transversal	<Nome>
Descrição	<Descrição do interesse>
Prioridade	<Prioridade pode ser Max, Méd e Min>
Lista de Requisitos	<Requisitos que descrevem o interesse>
Lista de Modelos	<Modelos UML influenciados pelo interesse>

A composição de um interesse não-funcional no diagrama de casos de uso é feita incluindo-se um novo caso de uso com o nome do interesse

transversal, definido na Tabela 2, porém com um relacionamento do tipo <<sobreposição>>, <<substituição>> ou <<empacotamento>>.

O relacionamento do tipo <<sobreposição>> significa que o interesse transversal modifica o requisito funcional que ele afeta. Neste caso, o interesse transversal pode ser executado antes ou depois da execução do requisito funcional. O relacionamento de <<substituição>> especifica que o interesse transversal substitui o requisito funcional que ele atravessa. O relacionamento de <<empacotamento>> especifica que o interesse transversal encapsula o requisito funcional que ele atravessa. Neste caso, o comportamento descrito pelo requisito funcional é empacotado pelo comportamento descrito pelo interesse transversal. Um exemplo do diagrama composto pode ser visualizado na Figura 27.

Figura 27 – Exemplo de diagrama composto com o interesse transversal de tempo de resposta adaptado de Araújo et al. (2002).

A Figura 27 apresenta um exemplo de diagrama de casos de uso referente ao sistema de rodovias portuguesas de alta velocidade. Em um sistema de tarifação do tráfego rodoviário, motoristas de veículos autorizados são cobrados automaticamente assim que passam por portões de cobrança. O interesse transversal não-funcional *Tempo de Resposta do Portão* empacota os casos de uso entrar na rodovia, sair da rodovia e passar sinalização. Isto significa que o comportamento funcional descrito pelos casos de uso são empacotados pelo comportamento descrito no caso de uso de tempo de resposta do portão. Diagramas de seqüência devem ser construídos para representar cenários do sistema de software e a seqüência de eventos referentes à composição.

Durante a composição de interesses transversais, situações de conflito podem ocorrer. No exemplo, o interesse transversal de segurança pode conflitar com o de tempo de resposta. Desta forma, as contribuições entre

interesses devem ser avaliadas e, em geral, o interesse transversal de maior prioridade deve ser composto primeiro. Para a priorização, análises de *trade-off* também devem ser conduzidas entre as partes interessadas.

3.3.2 Modelo de Atributos de Qualidade Transversais para a Engenharia de Requisitos

Um modelo para atributos de qualidade transversais, utilizando UML, foi proposto em Brito, Moreira e Araújo (2002) e Moreira, Araújo e Brito (2002). Este modelo tem o objetivo de identificar e especificar atributos de qualidade com o objetivo de incluí-los na descrição funcional do sistema de software durante a fase de requisitos (MOREIRA; ARAÚJO; BRITO, 2002). Este modelo está apresentado na Figura 28.

Figura 28 – Modelo de requisitos para atributos de qualidade de Brito, Moreira e Araújo (2002).

Este modelo é composto de três atividades principais: identificar, especificar e integrar requisitos.

A atividade para identificar requisitos consiste em detectar todos os requisitos do sistema de software e selecionar os atributos de qualidade relevantes para as partes interessadas.

A atividade para especificar requisitos é composta por: construir o diagrama de casos de uso, especificar de requisitos funcionais utilizando casos de uso, especificar atributos de qualidade usando formulários e identificar atributos transversais de qualidade, a fim de identificar os atributos de qualidade que afetam os requisitos funcionais. O formulário de especificação dos atributos de qualidade é apresentado na Tabela 3.

Tabela 3 – Formulário de especificação de um atributo de qualidade adaptado de Moreira, Araújo e Brito (2002).

Nome	<Nome do atributo de qualidade>
Descrição	<Descrição breve>
Foco	<Sistema de software ou processo de desenvolvimento>
Origem	<Origem da informação, por exemplo, partes interessadas, documentos, etc>
Decomposição	<Decomposição de atributos de qualidade. Quando todos os sub-atributos de qualidade são necessários para atender um atributo de qualidade, deve existir um relacionamento do tipo E entre eles. Se nem todos os sub-atributos de qualidade são necessários para atender um atributo de qualidade, deve existir um relacionamento do tipo OU entre eles>
Prioridade	<Importância do atributo de qualidade para as partes interessadas. Pode ser MÁXIMA, ALTA, BAIXA e MÍNIMA>
Obrigatoriedade	<Opcional ou mandatória>
Onde	<Lista dos modelos afetados pelo atributo de qualidade>
Requisitos	<Lista dos requisitos que descrevem o atributo de qualidade>
Contribuição	<Descreve como um atributo de qualidade pode ser afetado por outros atributos de qualidade. Esta contribuição pode ser positiva (+) ou negativa (-)>

A sub-atividade para identificar atributos de qualidade transversais é feita a partir da análise do campo *onde* da Tabela 3. Se um atributo de qualidade afetar mais de um caso de uso, este atributo será considerado transversal. Esta informação é confirmada através da análise do campo *requisitos*.

A atividade para integrar requisitos propõe um conjunto de modelos para representar a integração dos requisitos de qualidade transversais e requisitos funcionais. Os atributos de qualidade transversais são adicionados ao diagrama de casos de uso como novos casos de uso. Utiliza-se o relacionamento de inclusão entre os elementos. Um exemplo pode ser observado na Figura 29.

Figura 29 – Exemplo de diagrama composto com o interesse transversal de tempo de resposta de Moreira, Araújo e Brito (2002).

Na atividade integrar requisitos são utilizados os operadores de combinação (*weaving*), sobreposição, substituição e empacotamento. Um exemplo da utilização destes operadores na abordagem proposta em Brito, Moreira e Araújo (2002) é apresentada na Figura 30.

Figura 30 – Exemplo do uso de operadores de combinação no diagrama de casos de uso de Moreira, Araújo e Brito (2002).

Segundo os autores, neste exemplo ocorreu uma situação de sobreposição entre os atributos de qualidade de integridade e confidencialidade e a confidencialidade deve ser aplicada antes de execução da integridade. A linha tracejada mostra a condição *depois* e a linha cheia a condição *antes*.

3.3.3 Abordagem direcionada a Casos de Uso Aspectuais

A abordagem direcionada a casos de uso aspectuais tem o objetivo de integrar RNFs ao modelo de casos de uso e identificar quais casos de uso e quais RNFs são interesses transversais. Também é discutida a composição de interesses transversais, sejam casos de uso ou RNFs, com casos de uso que não são transversais (ARAÚJO; MOREIRA, 2003).

A representação desta abordagem é apresentada na Figura 31.

Figura 31 – Abordagem direcionada a casos de uso aspectuais adaptada de Araújo e Moreira (2003).

Inicialmente é realizado o passo identificar e definir os requisitos funcionais a partir dos requisitos do sistema de software. Os requisitos são inspecionados

a fim de identificar e definir atores e casos de uso para a construção do modelo inicial de casos de uso. Os atores são definidos a partir de suas funções.

O passo refinar os casos de uso tem o objetivo de identificar exceções ou funções comuns no modelo original de casos de uso. O resultado desta operação é apresentado através dos relacionamentos de <<extensão>> e <<inclusão>>.

O passo identificar e definir os requisitos não-funcionais obtém os RNFs através da análise dos requisitos elicitados e outras informações obtidas das partes interessadas. Cada RNF é descrito em um formulário, com o nome, descrição breve, e lista dos requisitos associados.

O passo integrar os requisitos não-funcionais no modelo de casos de uso faz a extensão do modelo de casos de uso com um novo estereótipo de relacionamento <<restrição>> que relaciona casos de uso funcionais com casos de uso correspondentes ao RNF, que recebem o estereótipo <<NFR>>.

O passo identificar os aspectos candidatos busca casos de uso relacionados a mais de um caso de uso, pois se entende que este caso de uso seja um candidato a aspecto. Um caso de uso será transversal se restringir, estender ou for incluído por mais de um caso de uso. O exemplo da Figura 32 apresenta uma situação com casos de uso transversais.

Figura 32 – Exemplos de casos de uso aspectuais de Araújo e Moreira (2003).

Neste exemplo os casos de uso *Tempo de Resposta*, *Promoção de Fim de Semana*, e *Validar de Cartão* são transversais, pois restringem, estendem e são incluídos por outros casos de uso. Conseqüentemente estes casos de uso

são candidatos a aspectos. O caso de uso *Debitar Cartão* não é transversal, pois é incluído apenas uma vez pelo caso de uso *Sair do Metrô*.

3.3.4 Integração do NFR Framework no modelo de Engenharia de Requisitos

A integração do NFR Framework no modelo de engenharia de requisitos tem o objetivo de identificar, especificar e integrar interesses, incluindo interesses transversais no modelo de casos de uso. Além disso, este modelo utiliza o NFR Framework para auxiliar na identificação e na especificação de interesses (BRITO; MOREIRA, 2004), (BRITO; MOREIRA, 2003). Este modelo é apresentado na Figura 33.

Figura 33 – Modelo de separação de interesses adaptado de Brito e Moreira (2004).

A atividade 1 para identificar interesses é executada através de uma análise extensa de documentos e outras informações provenientes das partes interessadas. Cada interesse deve ser descrito conforme formulário da Tabela 4.

Tabela 4 – Formulário de especificação de interesses adaptado de Brito e Moreira (2004).

Nome	<Nome do interesse>
Origem	<Origem da informação, como partes interessadas, documentos, domínio, catálogos e processos de negócio>
Partes Interessadas	<Pessoas que precisam do interesse>

Descrição	<Explica o comportamento pretendido do interesse>
Classificação	<Ajuda a seleção da abordagem mais apropriada para especificar o interesse. Por exemplo: funcional, não-funcional, <i>goal</i> >
Contribuição	<Representa como um interesse pode ser afetado por outros atributos de qualidade. Esta contribuição pode ser positiva (+) ou negativa (-)>
Prioridade	<Expressa a importância de um interesse para as partes interessadas. Pode ser muito importante, importante, médio, baixo e muito baixo>
Interesses Requeridos	<Lista dos interesses relacionados ao interesse em questão>

Os campos *Nome*, *Origem*, *Partes Interessadas* e *Descrição* são preenchidos durante esta atividade. Os demais campos são preenchidos na atividade 2.

A atividade 2, de especificar interesses, é detalhada em quatro sub-atividades:

- Usar a melhor abordagem: sugere o uso de técnicas existentes para a especificação de requisitos, incluindo o NFR *Framework*. Em alguns casos, a escolha pode ser feita durante a atividade 1, especialmente, se alguma técnica em particular foi utilizada durante o processo de eliciação. Nesta sub-atividade o campo *Classificação* da Tabela 4 deve ser preenchido.
- Identificar contribuições: consiste em definir o tipo de contribuição positiva (+) ou negativa (-) entre interesses. Esta informação deve ser registrada no campo *Contribuição* da Tabela 4.
- Identificar prioridades: tem o objetivo de identificar o grau de importância do interesse para o sistema de software a ser desenvolvido. Esta sub-atividade é feita com a colaboração das partes interessadas. O campo *Prioridade* da Tabela 4 deve ser preenchido neste momento.
- Identificar interesses requeridos: deve resultar em uma lista de interesses que o interesse em questão precisa para que sua função possa ser executada. Se um interesse não requer nenhum outro interesse, o campo *Interesses Requeridos* da Tabela 4 deve ser preenchido com <nenhum>.

A atividade 3, de identificar interesses transversais, tem o objetivo de analisar os campos *Interesses Requeridos* e *Descrição*, da Tabela 4, a fim de identificar quais interesses possuem comportamento transversal. Um interesse é considerado transversal se for requerido por mais de um interesse.

A atividade 4, para compor interesses é detalhada através de quatro sub-atividades:

- Identificar pontos de composição: é realizada com base no campo *Interesses Requeridos*, da Tabela 4, a qual define os pontos em que as composições são realizadas. Os pontos de composição são mapeados conforme Tabela 5. Cada célula da tabela deve ser preenchida com a lista dos interesses transversais (IT) que afetam um dado interesse e representa um ponto de composição (PC). Uma regra de composição deve ser definida para cada ponto de composição.

Tabela 5 – Matriz de partes interessadas x interesses adaptada de Brito e Moreira (2004).

	Interesse ₁	Interesse ₂	Interesse _n
Parte Interessada ₁	IT ₁ , IT ₂ (PC _A)	...	IT ₂ , IT ₅ (PC _J)
Parte Interessada ₂	IT ₁ , IT ₃ (PC _B)		
Parte Interessada _n			IT ₃ , IT ₅ (PC _Z)

- Identificar conflitos: deve ser executada para cada um dos pontos de composição com base no campo *Contribuição* da Tabela 4.
- Identificar interesses dominantes: auxilia na resolução de conflitos identificados na atividade anterior, e é baseada na informação presente no campo *Prioridade* da Tabela 4. Se a prioridade atribuída para cada interesse for diferente, o problema não é tão difícil de ser resolvido, sendo que o interesse dominante será aquele com a maior prioridade. Entretanto, se ao menos dois interesses transversais possuírem a mesma prioridade, um *trade-off* deverá ser negociado entre as partes interessadas.
- Definir regras de composição: tem a finalidade de definir a ordem na qual os interesses serão aplicados em um ponto de composição. Os operadores sugeridos por esta abordagem são:
 - Autorização $I_1 \gg I_2$: o comportamento do interesse 2 começa se o comportamento do interesse 1 foi concluído com sucesso.

- Desativação $I_1[<I_2$: o comportamento do interesse 2, quando iniciar, interrompe a execução do interesse 1.
- Intercalação Pura $I_1|||I_2$: o comportamento do interesse 1 evolui separadamente do comportamento do interesse 2.
- Sincronização Pura $I_1||I_2$: o interesse 1 deve ser sincronizado com o comportamento do interesse 2.

A saída da atividade de composição de interesses é a especificação dos interesses com o uso dos operadores de composição.

3.3.5 Abordagem Direcionada a Casos de Uso para o Desenvolvimento de Software Orientado a Aspectos

A abordagem direcionada a casos de uso para o desenvolvimento de software orientado a aspectos tem também o objetivo de utilizar os conceitos relacionados à orientação a aspectos desde as atividades iniciais do desenvolvimento (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003).

Esta abordagem propõe adaptações nos fluxos de trabalho de requisitos, análise e projeto do processo unificado (JACOBSON; BOOCH; RUMBAUGH, 1999) para integrar as atividades relacionadas à manipulação de requisitos aspectuais. Como esta tese tem foco na etapa de requisitos, apenas o fluxo de trabalho de requisitos é apresentado com detalhes na Figura 34.

Figura 34 – Atividades de requisitos adaptadas de Sousa (2004).

As atividades identificar atores e casos de uso e especificar casos de uso seguem as diretrizes de Jacobson, Booch e Rumbaugh (1999) sem modificações.

A atividade identificar e analisar requisitos não-funcionais faz uso do NFR *Framework* (CHUNG et al., 2000) sem modificações.

A atividade especificar requisitos não-funcionais detalha as operacionalizações identificadas através da aplicação do NFR *Framework* (CHUNG et al., 2000). Sousa (2004) classifica as operacionalizações em dois tipos, restrições operacionais e restrições de projeto. As restrições operacionais são funções que devem ser incorporadas ao sistema de software; as restrições de projeto são aquelas que contribuem para que uma restrição operacional possa ser satisfeita, tais como infra-estrutura de hardware, restrições de implementação, documentação ou gerenciamento de projeto.

Somente as restrições operacionais são especificadas de maneira semelhante a uma especificação de casos de uso. As restrições de projeto estão fora do escopo do trabalho de Sousa (2004).

A atividade estruturar o modelo de casos de uso é realizada através dos seguintes passos: definição do relacionamento entre um requisito aspectual e os casos de uso que ele afeta, definição da tabela de composição e resolução de conflitos entre requisitos aspectuais.

O passo de definição do relacionamento entre um requisito aspectual e os caso de uso que ele afeta utiliza o relacionamento do tipo *crosscuts*. Este relacionamento é utilizado quando um requisito aspectual deve ser relacionado a um ou mais casos de uso. O relacionamento do tipo *crosscuts* é apresentado na Figura 35.

Figura 35 – Exemplo de um relacionamento *crosscuts* adaptado de Sousa (2004).

O losango representado na Figura 35 é o requisito aspectual *Verificar Débito*. Neste exemplo, a verificação de débito deve ser executada nos três casos de uso, *Emitir Histórico*, *Fazer Matrícula* e *Trancar Matrícula*.

Três heurísticas são propostas para identificar o relacionamento do tipo *crosscuts* (SOUSA, 2004):

1. Quando um interesse precisa ser inserido em um caso de uso, alterando o seu fluxo de execução, porém seu propósito não está diretamente associado ao objetivo deste caso de uso;
2. Quando um interesse pode ser desvinculado do caso de uso sem que isso impeça o seu funcionamento;
3. Quando um interesse pode ou poderá ser utilizado futuramente em vários casos de uso, e não apenas em um caso de uso específico.

O passo de definição da tabela de composição tem a finalidade de identificar informações a respeito da composição entre um requisito aspectual e os casos de uso afetados por ele. Esta tabela é chamada de tabela de composição de nível 1 e é utilizada no fluxo de trabalho de requisitos. As

tabelas de níveis 2 e 3 são utilizadas nos fluxos de trabalho de análise e projeto respectivamente. A Tabela 6 apresenta o formato da tabela de composição de nível 1.

Tabela 6 – Tabela de composição de nível 1 para um requisito aspectual de Sousa (2004).

REQUISITO ASPECTUAL: # ID <NOME>				
Caso de Uso Afetado	Condição (Opcional)	Regra de Composição	Ponto de Junção	Informações Adicionais
#ID <nome>	Condição	{sobreposição.depois sobreposição.antes substituição empacotamento}	Passo do cenário	Informações relativas à composição

Para cada ponto em que um requisito aspectual afetar um caso de uso haverá uma entrada na tabela de composição nível 1. As colunas desta tabela devem ser preenchidas da seguinte forma:

- Caso de uso afetado: preencher com o nome do caso de uso no qual o comportamento do requisito aspectual deve ser adicionado.
- Condição (opcional): preencher caso exista alguma condição que deva ser satisfeita para que o comportamento do requisito aspectual possa ser adicionado no caso de uso.
- Regra de composição: preencher com um dos operadores de composição. A escolha de um dos operadores dependerá do comportamento do requisito aspectual a ser aplicado no ponto de junção. Os operadores de composição foram definidos com base nas definições de Moreira, Araújo e Brito (2002):
 - Sobreposição: indica que o comportamento do aspecto deve ser aplicado antes (*before*) ou depois (*after*) da execução do ponto de junção que ele afeta.
 - Substituição: indica que o comportamento do aspecto substitui o ponto de junção que ele afeta.
 - Empacotamento: indica que o comportamento do aspecto encapsula o comportamento do ponto de junção que ele afeta, ou seja, indica que a maneira na qual o ponto de junção será executado, ou até mesmo se ele será executado, será de responsabilidade do aspecto.

- Ponto de junção: preencher com o passo do cenário do caso de uso no qual o requisito aspectual deve ser adicionado.
- Informações adicionais: preencher com alguma outra informação necessária sobre a composição do requisito aspectual.

Tomando como base o exemplo da Figura 35, todas as informações referentes à verificação de débito devem ficar disponíveis em uma tabela de composição de nível 1, conforme exemplo da Tabela 7.

Tabela 7 – Exemplo da tabela de composição de nível 1 preenchida de Sousa (2004).

REQUISITO ASPECTUAL: AR#1 VERIFICAÇÃO DE DÉBITO				
Caso de Uso Afetado	Condição (Opcional)	Regra de Composição	Ponto de Junção	Informações Adicionais
UC#5: Fazer Matrícula	-	Empacotamento	3	Se semDébito então execute (ponto de junção) Senão exibe (msg Erro)

No exemplo da Tabela 7 a regra de composição definida para o caso de uso *Fazer Matrícula* foi empacotamento, pois a execução do ponto de junção dependerá do resultado obtido na verificação de débito para que a matrícula seja executada.

O passo de resolução de conflitos entre requisitos aspectuais deve ser executado quando diferentes requisitos aspectuais precisarem ser adicionados no mesmo ponto de junção de um mesmo caso de uso.

A ordem de execução dos requisitos aspectuais deve seguir a ordem de precedência dos seus respectivos operadores de composição. Entretanto, caso dois ou mais requisitos aspectuais, com os mesmos operadores de composição, precisarem ser adicionados no mesmo ponto de junção, a ordem de precedência deve ser decidida pelo desenvolvedor.

Caso exista um conflito de um requisito aspectual em um mesmo passo de um caso de uso, a ordem de precedência desse requisito aspectual em relação aos requisitos aspectuais com que este conflita deve ser especificada da seguinte forma: $P[\text{numPassoConflito}] - [\text{siglaOperador}] = [\text{ordemPrecedência}]$, sendo que:

- numPassoConflito deve ser substituído pelo número do passo do caso de uso onde existe o conflito.

- siglaOperador deve ser substituído por SA, caso ocorra o conflito com o operador sobreposição.antes, SD, caso ocorra o conflito com o operador de sobreposição.depois, S, caso ocorra o conflito com o operador de substituição e E caso ocorra o conflito com o operador de empacotamento.
- ordemPrecedência deve ser substituído pela ordem de precedência do requisito aspectual.

A Tabela 8 apresenta um exemplo de tabela de conflitos a ser elaborada para todo o sistema de software.

Tabela 8 – Tabela de conflitos entre requisitos aspectuais de Sousa (2004).

	AR#01	AR#02	AR#n
UC#01			
UC#02	P3-E=2°	P3-E=1°	...
UC#n			

No exemplo da Tabela 8, o caso de uso 2 (UC#2) possui o mesmo operador de composição de empacotamento do passo 3, por isso, foi necessária a definição da ordem de precedência entre os requisitos aspectuais 1 e 2 (AR#1 e AR#2).

3.3.6 Modelo de Engenharia de Requisitos Orientado a Interesses

O modelo de engenharia de requisitos orientado a interesses tem o objetivo de identificar interesses transversais funcionais e não-funcionais de maneira uniforme, isto é, a natureza do interesse, seja ela funcional ou não-funcional não interfere na forma de tratamento dos interesses deste modelo (RASHID; MOREIRA; ARAÚJO, 2003), (MOREIRA; ARAÚJO; RASHID, 2005), (MOREIRA; RASHID; ARAÚJO, 2005).

A Figura 36 apresenta o modelo de engenharia de requisitos orientado a interesses.

Figura 36 – Modelo de engenharia de requisitos baseada no tratamento uniforme de interesses adaptado de Moreira, Araújo e Rashid (2005).

A atividade identificar e especificar interesses realiza a identificação de interesses através de mecanismos como pontos de vista, casos de uso e *softgoals*; a especificação dos interesses identificados é feita utilizando XML (*eXtensible Markup Language*). Um exemplo de especificação de interesse é apresentado na Figura 37.

```

<?xml version="1.0"?>
<Interesse nome="Mobilidade">
  <Requisito id="1">
    O sistema será acessado em movimento.
    <Requisito id="1.1">
      O sistema será acessado entre uma área limitada.
    </Requisito>
  </Requisito>
</Interesse>
  
```

Figura 37 – Especificação do interesse de mobilidade em XML.

A atividade para identificar relacionamentos de interesses de alta granularidade tem o objetivo de relacionar interesses entre si através de interdependências. Estes relacionamentos são identificados usando técnicas como análise de domínio, etnografia, processamento de linguagem natural, entre outras.

A atividade para especificar projeções dos interesses usando regras de composição tem a finalidade de especificar como um requisito pertencente a um interesse influencia o comportamento de um conjunto de requisitos em vários outros interesses. Nesta atividade devem ser realizadas também as análises de *trade-off* entre interesses.

As Tabelas 9(a) e 9(b) apresentam as ações e operações possíveis a serem aplicadas durante a especificação da composição, que também deve ser expressa em XML.

Tabela 9 – Descrição das ações e operações de restrição adaptada de Moreira, Araújo e Rashid (2005).

Ação de Restrição	
Tipo	Descrição
Enforce	Usada para impor uma condição adicional sobre um conjunto de requisitos de um interesse.
Ensure	Usada para garantir que uma condição deve existir para um conjunto de requisitos de um interesse realmente existe.
Provide	Usada para especificar funções adicionais a serem incorporadas por um conjunto de requisitos de um interesse.
Applied	Usada para descrever regras que alteram a saída de um conjunto de requisitos de um interesse.
Exclude	Usada para excluir alguns interesses ou requisitos se a condição <i>todos</i> é especificada.
Affect	Usada para especificar um conjunto de requisitos de um interesse que altera o estado de outro interesse.

(a)

Operação de Restrição	
Tipo	Descrição
During	Descreve um intervalo temporal durante o qual um conjunto de requisitos está sendo satisfeito.
Between	Descreve um intervalo temporal entre a satisfação de dois requisitos. O intervalo inicia quando o primeiro requisito é satisfeito e termina quando o segundo começa a ser satisfeito.
On	Descreve o ponto do intervalo temporal depois que um conjunto de requisitos foi satisfeito.
For	Descreve que funções adicionais complementam os requisitos dos interesses.
With	Descreve que uma condição é válida para dois conjuntos de requisitos relacionados.
In	Descreve que uma condição é válida para um conjunto de requisitos que foram satisfeitos.
AND, OR, XOR	Conjunção, disjunção e OU-exclusivo.

(b)

O conjunto de atividades para manipular conflitos deve ser executado na sequência.

A atividade construir a tabela de contribuição tem a finalidade de identificar a contribuição (positiva ou negativa) que um interesse pode exercer sobre

outros interesses. A definição de contribuição é baseada nas definições de correlações de Chung et al. (2000).

A atividade identificar projeções refletidas através de desdobramentos tem o objetivo reduzir a quantidade de análises de contribuições entre interesses retirando a parte simétrica da tabela de interesses x interesses. Um exemplo é apresentado na Tabela 10.

Tabela 10 – Exemplo de simetria na tabela de interesses x interesses adaptado de Moreira, Araújo e Rashid (2005).

I_1	I_1				
I_2	+	I_2			
I_3	-	+	I_3		
...				...	
I_n		-			I_n

A atividade atribuir pesos para interesses conflitantes tem o objetivo de determinar o grau de importância que um interesse exerce sobre o outro. Pesos são atribuídos aos interesses conforme escalas. As escalas possuem os seguintes graus de importância: muito importante (entre 0,8 e 1,0), importante (entre 0,5 e 0,8), médio (entre 0,3 e 0,5), não tão importante (entre 0,1 e 0,3) e não considerado (entre 0 e 0,1).

A atividade resolver conflitos deve ser feita baseada nos pesos e negociada entre as partes interessadas. Uma vez que todos os conflitos tenham sido resolvidos, a especificação é revisada e uma nova composição é realizada a fim de identificar outros possíveis conflitos.

Finalmente, a atividade especificar dimensões dos interesses deve identificar a influência dos interesses em fases posteriores do desenvolvimento. Especificação, projeto, evolução, arquitetura e implementação são as possíveis fases que sofrem influências dos interesses.

Os trabalhos tais como: Vieira, Brito e Moreira (2006) e Brito et al. (2007) evoluíram a proposta descrita nesta seção e propuseram o uso do método de análise multi-critérios, ou MCDM (*Multiple Criteria Decision Making*) e o processo hierárquico analítico, ou AHP (*Analytical Hierarchical Process*) para o gerenciamento de conflitos a fim de evitar julgamentos baseados em decisões humanas.

3.3.7 Processo da Engenharia de Requisitos Orientado a Aspectos AOSD-Europe

A AOSD-Europe é um grupo de pesquisa europeu que tem o objetivo de integrar pesquisa, treinamento e atividades relacionadas ao desenvolvimento de software orientado a aspectos e inovações nas áreas de análise e projeto orientado a aspectos, métodos formais, linguagens e aplicações de técnicas de desenvolvimento de software orientado a aspectos (AOSD-Europe, 2010).

Como parte do trabalho deste grupo, foi feita uma proposta inicial do processo de engenharia de requisitos orientada a aspectos com os seguintes objetivos:

- Consolidar o trabalho desenvolvido na área de engenharia de requisitos orientada a aspectos;
- Integrar as lições aprendidas das abordagens existentes;
- Ampliar a maturidade dos modelos propostos;
- Reduzir a duplicação do trabalho das pesquisas na área para que pontos mais avançados pudessem ser estudados.

Este processo está representado na Figura 38 e apresenta dois conjuntos de atividades, de identificação e análise de interesses.

Figura 38 – Processo de engenharia de requisitos orientado a aspectos adaptado de Chitchyan et al. (2006).

O conjunto de atividades de identificação de interesses tem foco principal na aquisição e estruturação inicial de requisitos. O conjunto de atividades de

análise de interesses tem foco principal na análise de requisitos e resolução de inconsistências.

O conjunto de atividades de identificação de interesses consiste de: elicitação, identificação, representação e refinamento de interesses.

Durante a atividade de elicitação de interesses, o engenheiro de requisitos estabelece os pontos relevantes sobre os pontos de vistas das partes interessadas, isto é, os interesses relacionados ao sistema de software a ser desenvolvido. Esta atividade pode ser executada através de técnicas de discussões com as partes interessadas, entrevistas, observação etnográfica, entre outras. A saída desta atividade é a lista inicial de interesses das partes interessadas.

Durante a atividade de identificação de interesses, os interesses identificados são representados utilizando-se métodos adequados tais como o *NFR Framework* (CHUNG et al., 2000), abordagem baseada em temas (CLARKE; BANIASSAD, 2005), entre outras. Exemplos de representações de interesses são grafos de interdependência de *softgoals*, pontos de vista, temas, casos de uso, entre outros.

Após a atividade de refinamento de requisitos, os interesses refinados podem alimentar novamente as atividades de elicitação, identificação e representação. Quando os interesses estiverem suficientemente detalhados, passa-se para o conjunto de atividades de análise.

Este conjunto consiste de: composição de interesses, resolução de *trade-off*, mapeamento de requisitos e refinamento de requisitos.

A atividade de composição de interesses tem a finalidade de identificar conflitos entre interesses com a mesma importância, que precisam ser compostos no mesmo ponto (BRITO et al., 2007).

A atividade de resolução de *trade-off* procura solução para os conflitos identificados. A partir deste ponto, pode-se voltar à atividade de identificação de interesses e novas iterações de refinamento, elicitação, identificação e representação podem ser iniciadas.

É importante ressaltar que a atividade de refinamento de requisitos é incluída em ambos os conjuntos de atividades, de identificação e de análise, porque o refinamento de requisitos é a atividade principal do processo de engenharia de requisitos, quando se detecta precisamente o que deve ser

executado pelo sistema de software a ser desenvolvido (CHITCHYAN et al., 2006).

Complementando a descrição deste processo, duas técnicas são propostas, a Early-AIM (*Early Aspect Identification and Mining*) e CoCA (*Composition-Centric Approach*). A técnica Early-AIM é proposta para as atividades de elicitaco, identificao e refinamento de requisitos com o auxlio de uma ferramenta chamada EA-Miner, que fornece suporte automtico para identificar e separar interesses funcionais e no-funcionais no nvel de requisitos. A tcnica CoCA proposta para as atividades de refinamento, composio, *trade-off* e mapeamento de requisitos e faz uso de uma linguagem de descrio de requisitos para representao de interesses e posterior composio.

3.3.8 Outras Abordagens

Outros trabalhos relevantes da rea de engenharia de requisitos orientada a aspectos esto relacionados nesta seo. So eles: desenvolvimento de software orientado a aspectos com casos de uso (JACOBSON; NG, 2004), (JACOBSON, 2003a, 2003b), abordagem baseada em temas (CLARKE; BANIASSAD, 2005), identificao de aspectos candidatos com a abordagem I* (ALENCAR et al., 2006), estratgia orientada a aspectos para modelagem de requisitos (SILVA, 2006), (SILVA; LEITE, 2005a, 2005b) e a notaco de requisitos de usurio orientado a aspectos (MUSSBACHER, 2008).

Estas abordagens esto relacionadas nesta seo pois, apesar de no possirem semelhanas significativas com este trabalho, so reconhecidas e referenciadas no meio acadmico.

O desenvolvimento de software orientado a aspectos com casos de uso (JACOBSON; NG, 2004), (JACOBSON, 2003a, 2003b), considera que os casos de uso so interesses transversais por natureza, pois a sua realizao envolve diversas classes. Esta abordagem tambm define o conceito de fatia de caso de uso, ou *use case slice*. Uma fatia de caso de uso contm todas as especificidades relacionadas a um caso de uso. Os interesses no-funcionais so representados por um tipo especial de caso de uso chamado caso de uso de infra-estrutura. Nesta abordagem dois tipos distintos de separao de

interesses podem ser aplicados, o primeiro através de casos de uso do tipo *peer*, que são casos de uso independentes, isto é, que não fazem referências a outros casos de uso, e o segundo tipo através de extensões de casos de uso. A composição de interesses é executada através de operadores do tipo *after*, *before* e *around* aplicadas às extensões de casos de uso. São definidas as diretrizes para manipular sobreposições de interesses através de unidades chamadas *non-specific use case slices*, que agrupam comportamento comum de diversos casos de uso.

A abordagem baseada em temas define um tema como o encapsulamento de um interesse (CLARKE; BANIASSAD, 2005). No nível de requisitos, um tema é considerado um subconjunto de responsabilidades descritas por um conjunto de requisitos. No nível de projeto, os temas incluem as estruturas e comportamentos necessários para satisfazer as responsabilidades definidas pelos requisitos. Esta abordagem propõe três atividades principais: análise, projeto e composição. A análise tem o objetivo de identificar requisitos e temas. Esta atividade envolve o mapeamento de requisitos em interesses. A representação Tema/Doc permite a visualização dos relacionamentos entre interesses. Estes relacionamentos expõem os interesses entrelaçados e permitem que aspectos sejam identificados. A atividade de projeto utiliza o Tema/UML, baseado na linguagem UML, onde são identificadas classes e métodos potenciais. Aspectos podem surgir durante o detalhamento desta atividade. A atividade de composição define como os modelos Tema/UML são combinados, isto é, como os temas estão inter-relacionados.

A identificação de aspectos candidatos com a abordagem I* (ALENCAR et al., 2006) tem o objetivo de identificar aspectos candidatos utilizando o *framework* I*. Este *framework* permite descrever os relacionamentos entre a organização e os diversos agentes do sistema de software a ser desenvolvido, bem como prover o entendimento das razões lógicas a respeito das decisões tomadas. Os interesses transversais são detectados no *framework* I*, quando um elemento é requerido ou afeta vários outros elementos do modelo, modificando seu comportamento original. O resultado da identificação de aspectos candidatos é validado pela aplicação de um conjunto de heurísticas para derivar o modelo de casos de uso a partir dos modelos I* e, assim, identificar interesses transversais potenciais nos modelos UML resultantes. Os

interesses transversais são identificados no modelo de caso de uso quando um caso de uso é incluído por mais de um caso de uso, ou se existe um caso de uso que seja estendido por mais de um caso de uso.

A estratégia orientada a aspectos para modelagem de requisitos (SILVA, 2006), (SILVA; LEITE, 2005a, 2005b) propõe um metamodelo para integração de características transversais. A estratégia é composta por três atividades, separar, compor e visualizar. A atividade de separação conta com o uso do V-graph, a linguagem LMROA (Linguagem de Modelagem de Requisitos Orientada a Aspectos) e diretrizes para que os requisitos sejam modelados. O relacionamento transversal entre requisitos pode ser identificado e registrado em um novo tipo de relacionamento, adicionado como uma extensão do modelo de requisitos. A atividade de composição tem a finalidade de combinar os requisitos modelados, sendo que o modelo resultante é a entrada para que sejam extraídas informações necessárias para a atividade de visualização.

A notação de requisitos de usuário orientado a aspectos (MUSSBACHER, 2008) propõe o uso da notação orientada a aspectos para requisitos do usuário, também conhecida como AoURN (*Aspec-Oriented User Requirements Notation*), que integra os mapas de caso de uso orientados a aspectos (MUSSBACHER; AMYOT; WEISS, 2007), também conhecidos como AoUCM (*Aspect-Oriented Use Case Maps*) e a linguagem de requisitos orientada a goals e aspectos, a AoGRL (*Aspect-Oriented and Goal-Oriented Requirements Language*). Esta proposta integra conceitos do NFR *Framework* (CHUNG et al., 2000), casos de uso, e conceitos da orientação a aspectos em uma linguagem similar à do *framework I**.

3.4 Considerações Finais

Existem vários termos na literatura referentes ao uso do termo aspecto durante as fases iniciais do desenvolvimento. Exemplos destes termos são: requisitos aspectuais, aspecto candidato e aspecto antecipado. Neste trabalho é utilizado o termo aspecto, por ser considerado mais abrangente.

O termo requisito aspectual não é utilizado por não estar consolidado no meio acadêmico (NIU; EASTERBROOK; YU, 2007). Os termos aspecto

candidato e aspecto antecipado também não são utilizados, pois se entende que não há necessidade de identificar aspectos que podem ou não ser confirmados nas fases posteriores do desenvolvimento.

No decorrer do capítulo 3 foram apresentadas diversas abordagens e processos relacionados à engenharia de requisitos orientada a aspectos.

O modelo de engenharia de requisitos orientado a aspectos com UML (RASHID et al., 2002), (ARAÚJO et al., 2002), o modelo de atributos de qualidade transversais para a engenharia de requisitos (BRITO; MOREIRA; ARAÚJO, 2002), (MOREIRA, ARAÚJO; BRITO, 2002) e a abordagem direcionada a casos de uso aspectuais (ARAÚJO; MOREIRA, 2003) foram os trabalhos pioneiros na área de engenharia de requisitos orientada a aspectos com propostas que propõem a integração de métodos orientados a aspectos ao modelo de casos de uso.

A evolução dos trabalhos se deu com e a integração do NFR *Framework* no modelo de engenharia de requisitos (BRITO; MOREIRA, 2004), (BRITO; MOREIRA, 2003) e a abordagem direcionada a casos de uso para o desenvolvimento e software orientado a aspectos (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003) que integram o uso do NFR *Framework* para tratamento de interesses não-funcionais.

O modelo de engenharia de requisitos orientado a interesses (RASHID; MOREIRA; ARAÚJO, 2003), (MOREIRA; ARAÚJO; RASHID, 2005), (MOREIRA; RASHID; ARAÚJO, 2005) e o processo da engenharia de requisitos orientada a aspectos AOSD-Europe (CHITCHYAN et al., 2006) apresentam propostas não atreladas ao uso de métodos específicos, como casos de uso e NFR *Framework*, para tratamento de interesses transversais, o que torna estas abordagens mais genéricas e flexíveis.

As abordagens e processos relacionados à engenharia de requisitos orientada a aspectos apresentados neste capítulo deram subsídios para a definição do PARNAFOA. As justificativas e escolhas para o uso das abordagens e processos propostos na literatura, que foram integrados ao PARNAFOA, são apresentadas no capítulo 4.

4. Descrição do Processo de Análise de Requisitos Não-Funcionais Orientado a Aspectos – PARNAFOA

“A chave para produzir uma mudança duradoura é a disciplina para fazer as coisas passo a passo.” – Jim Collins

Este capítulo tem o objetivo de apresentar as principais características do PARNAFOA, a discussão em relação aos trabalhos apresentados no capítulo 3 e a descrição do processo propriamente dita.

4.1 Características Principais do PARNAFOA

O PARNAFOA tem o objetivo de integrar interesses transversais e não-transversais não-funcionais ao modelo de casos de uso de sistemas de software. Isso é obtido através de um conjunto de atividades para a identificação, separação e composição de interesses não-funcionais, sejam eles transversais ou não, e a incorporação dos resultados no modelo de casos de uso de um sistema de software. Obtém-se, com isso, um modelo de casos de uso que contém interesses funcionais e não-funcionais integrados, sendo que, os interesses não-funcionais são identificados e tratados desde as fases iniciais do desenvolvimento.

O PARNAFOA utiliza *softgoals* e casos de uso para representar interesses não-funcionais e funcionais, assim como diversos trabalhos da literatura (CHITCHYAN et al., 2006), (ARAÚJO; MOREIRA, 2003), (BRITO; MOREIRA, 2003), (BRITO; MOREIRA, 2004), (SOUSA, 2004), (SOUSA; SILVA; CASTRO, 2003), (MOREIRA; ARAÚJO; RASHID, 2005).

As abordagens, modelos e processos apresentados no capítulo 3 contribuíram para a definição da estrutura de sub-processos do PARNAFOA. Em especial destaca-se o processo da engenharia de requisitos orientada a aspectos AOSD-Europe (CHITCHYAN et al., 2006) que apresenta uma proposta consolidada a partir de vários trabalhos da área.

Estes trabalhos consideram a incorporação de aspectos nas fases iniciais do desenvolvimento através do modelo de casos de uso. A decisão a respeito do tratamento de aspectos como novos casos de uso, neste trabalho, foi baseada no modelo de engenharia de requisitos orientado a aspectos com

UML (RASHID et al., 2002), (ARAÚJO et al., 2002), no modelo de atributos de qualidade transversais para a engenharia de requisitos (BRITO; MOREIRA; ARAÚJO, 2002), (MOREIRA; ARAÚJO; BRITO, 2002) e na abordagem direcionada a casos de uso aspectuais (ARAÚJO; MOREIRA, 2003).

A decisão da utilização do relacionamento de inclusão entre novos casos de uso a serem incorporados no modelo de casos de uso foi motivada pelo modelo de atributos de qualidade transversais para a engenharia de requisitos (BRITO; MOREIRA; ARAÚJO, 2002), (MOREIRA; ARAÚJO; BRITO, 2002). A proposta deste trabalho é não introduzir novos tipos de relacionamento no modelo de casos de uso. Demais abordagens propõem o uso de novos tipos de relacionamentos como sobreposição, substituição, empacotamento, restrição e *crosscuts*, que não foram utilizados neste trabalho.

Os dados obtidos na modelagem dos requisitos funcionais e não-funcionais ficam presentes no modelo de casos de uso e nos SIGs, respectivamente. Desta forma, o uso de formulários para descrição de interesses, conforme propostas do modelo de engenharia de requisitos orientado a aspectos com UML (RASHID et al., 2002), (ARAÚJO et al., 2002), modelo de atributos de qualidade transversais para a engenharia de requisitos (BRITO; MOREIRA; ARAÚJO, 2002), (MOREIRA; ARAÚJO; BRITO, 2002), abordagem direcionada a casos de uso aspectuais (ARAÚJO; MOREIRA, 2003) e integração do NFR *Framework* no modelo de engenharia de requisitos (BRITO; MOREIRA, 2004), (BRITO; MOREIRA, 2003) não foram utilizados neste trabalho para evitar redundância de informações.

Informações sobre relacionamentos positivos e negativos entre interesses estão presentes nos SIGs na forma de correlações positivas e negativas. Tabelas de contribuições, positivas e negativas, entre interesses provenientes das propostas de integração do NFR *Framework* no modelo de engenharia de requisitos (BRITO; MOREIRA, 2004), (BRITO; MOREIRA, 2003) e, do modelo de engenharia de requisitos orientado a interesses (RASHID; MOREIRA; ARAÚJO, 2003), (MOREIRA; ARAÚJO; RASHID, 2005), (MOREIRA; RASHID; ARAÚJO, 2005), não foram utilizadas para que a informação não fosse duplicada.

O PARNAFOA utiliza o NFR *Framework* desde a identificação dos NFRs até a definição de metas. Isto não ocorre na maioria das abordagens e modelos

apresentados, a qual utiliza o NFR *Framework* somente para a decomposição de tipos de RNFs. A exceção é feita na abordagem direcionada a casos de uso para o desenvolvimento de software orientado a aspectos (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003), a qual utiliza o NFR *Framework* para detalhar e analisar os requisitos não-funcionais.

A escolha das regras de composição do PARNAFOA foi baseada na abordagem direcionada a casos de uso para o desenvolvimento de software orientado a aspectos (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003), por serem de fácil manipulação e entendimento. As definições da integração do NFR *Framework* no modelo de engenharia de requisitos (BRITO; MOREIRA, 2004), (BRITO; MOREIRA, 2003) e do modelo de engenharia de requisitos orientado a interesses (RASHID; MOREIRA; ARAÚJO, 2003), (MOREIRA; ARAÚJO; RASHID, 2005), (MOREIRA; RASHID; ARAÚJO, 2005) não são utilizados neste trabalho. Apesar de tornarem o processo de composição mais preciso, estes trabalhos tornam a atividade de composição de interesses demasiadamente complexa pela inclusão de uma quantidade excessiva de regras e operadores.

O PARNAFOA considera restrições de projeto e restrições operacionais identificados durante a construção dos SIGs. Restrições de projeto estão fora do escopo da abordagem direcionada a casos de uso para o desenvolvimento de software orientado a aspectos (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003).

Para a resolução de conflitos com a definição de regras de precedência, a proposta apresentada na abordagem direcionada a casos de uso para o desenvolvimento de software orientado a aspectos (SOUSA, 2004), (SOUSA et al., 2004), (SOUSA; SILVA; CASTRO, 2003) foi considerada apropriada. A proposta do modelo de engenharia de requisitos orientado a interesses (RASHID; MOREIRA; ARAÚJO, 2003), (MOREIRA; ARAÚJO; RASHID, 2005), (MOREIRA; RASHID; ARAÚJO, 2005) traz benefícios na priorização de interesses durante a resolução de conflitos para a composição, porém, sua utilidade seria maior se aplicada inicialmente na priorização de interesses. Durante a composição de interesses, as prioridades e regras de precedência devem ser definidas a partir do entendimento do sistema de software e seus requisitos. A utilização de métodos de análise para a priorização podem

auxiliar, mas não são suficientes para decidir o que deve ser priorizado durante a composição. Esta análise deve contar com o apoio de especialistas.

O modelo de engenharia de requisitos orientado a interesses (RASHID; MOREIRA; ARAÚJO, 2003), (MOREIRA; ARAÚJO; RASHID, 2005), (MOREIRA; RASHID; ARAÚJO, 2005) utiliza XML na especificação e composição de interesses, com o objetivo de torná-las mais flexíveis, porém, traz maior complexidade no entendimento e manipulação das especificações pelas partes interessadas.

Com relação às demais abordagens, a proposta de (JACOBSON; NG, 2004), (JACOBSON, 2003a, 2003b) não dá ênfase ao tratamento de interesses não-funcionais durante as fases iniciais de desenvolvimento.

A proposta de temas para especificar aspectos (CLARKE; BANIASSAD, 2005) requer o uso de ferramentas de apoio que auxiliem na representação dos temas, pois em sistemas de grande porte, as representações podem tornar-se demasiadamente complexas. A falta de suporte ferramental pode inviabilizar sua aplicação dependendo do porte do projeto.

O uso de uma linguagem definida com base em conceitos da programação orientada a aspectos e XML, proposta por Silva (2006) e Silva e Leite (2005a, 2005b) são fatores que agregam maior complexidade para o processo de identificação e composição de aspectos na fase de requisitos.

Os trabalhos de Mussbacher (2008) e Mussbacher, Amyot e Weiss (2007) apresentam uma linguagem que enfatiza a importância das partes interessadas na definição dos sistemas de software. Adaptações da representação do NFR *Framework* são propostas, bem como o uso das representações do *framework* I*. Neste trabalho, optou-se pelo uso da representação original do NFR *Framework*, o que inviabiliza a adoção desta proposta.

A definição do PARNAFOA conta com o detalhamento de métodos relacionados ao tratamento de requisitos não-funcionais, descritos no capítulo 2 e abordagens orientadas a aspectos, descritas no capítulo 3. Este tratamento foi necessário, para que os métodos utilizados apresentassem o mesmo nível de detalhamento para sua inclusão no processo.

4.2 Visão Geral do PARNAFOA

De acordo com a OMG-BPMN (2009), um processo é um conjunto de atividades seqüenciadas. Um sub-processo representa um processo incluído dentro de outro processo. Uma atividade define o trabalho que uma organização executa dentro de um processo ou sub-processo.

O PARNAFOA foi descrito conforme definições de Bandor (2007), que estabelece os principais elementos a serem considerados na descrição de um processo. Um processo deve conter definições sobre papéis e responsabilidades para execução do trabalho, ferramentas a serem utilizadas, procedimentos e métodos que possam definir como as atividades devem ser executadas.

A representação do PARNAFOA é feita através da *Business Management Process Notation* (BPMN). Esta notação pode ser compreendida por analistas, que criam processos, técnicos de desenvolvimento, responsáveis pela implementação da tecnologia que provê suporte ao processo e pelas pessoas do negócio, que gerenciam e monitoram o processo (OMG-BPMN, 2009). Um resumo de sua notação encontra-se no Apêndice A.

Para executar adequadamente as atividades do PARNAFOA, os participantes deste processo devem ter perfil recomendado a um engenheiro de requisitos, com habilidades para capacidade analítica, de estruturação e de escrita da documentação de requisitos. Ainda, boa comunicação, empatia, moderação de discussões e resolução de conflitos são características importantes para o relacionamento com partes interessadas (PAECH, 2008).

De igual importância são as habilidades adquiridas com a experiência. Como um processo de engenharia de requisitos é complexo, heurísticas conhecidas como melhores práticas (KOTONYA; SOMMERVILLE, 1998) e padrões de requisitos (HAGGE; LAPPE, 2005) são importantes.

Para complementar a lista de habilidades necessárias, um engenheiro de requisitos deve executar adequadamente as atividades de requisitos definidas com base no modelo CMMi (CHRISSIS; KONRAD; SHRUM, 2007) que, de maneira geral, compreende as atividades de elicitação, análise, validação, especificação e gerenciamento de requisitos.

Além da habilidade requerida por um engenheiro de requisitos, para que um profissional possa utilizar o PARNAFOA, deve conhecer as técnicas que foram utilizadas na definição do processo, ou seja, modelagem de casos de uso, NFR *Framework* e conceitos relacionados aos métodos de engenharia de requisitos orientada a aspectos.

Esta seção apresenta a visão macroscópica do processo, através da descrição sucinta dos seus sub-processos. Nas seções seguintes é apresentada a descrição dos sub-processos e os respectivos artefatos de entrada e saída. As descrições detalhadas das atividades e dos artefatos gerados pelo PARNAFOA encontram-se nos Apêndices B e C, respectivamente. A visão geral do PARNAFOA é apresentada na Figura 39.

Figura 39 – Visão geral do PARNAFOA.

Durante o sub-processo para obter os requisitos sistema de software (1), os requisitos do sistema de software recebidos como dados de entrada são avaliados e, conforme necessário, podem ser revisados para adequação do nível de detalhamento e do formato requeridos pelo PARNAFOA.

Durante o sub-processo para relacionar interesses funcionais e não-funcionais (2) do sistema de software, os interesses funcionais e não-funcionais são identificados; os interesses não-funcionais são classificados e associados aos interesses funcionais no modelo de casos de uso.

Durante o sub-processo para identificar aspectos (3), os requisitos não-funcionais, que compõem os interesses não-funcionais, são avaliados e os aspectos são identificados.

Durante o sub-processo para decompor requisitos não-funcionais (4), os requisitos não-funcionais são detalhados e analisados através de SIGs. Este sub-processo deve ser executado repetidas vezes até que todos os requisitos não-funcionais estejam decompostos.

Durante o sub-processo para elaborar o modelo composto (5) o resultado proveniente da decomposição dos RNFs são incorporados ao modelo de casos de uso do sistema de software. Análises de conflito durante a composição dos requisitos não-funcionais são executadas neste sub-processo.

4.3 Sub-Processo 1: Obter os Requisitos do Sistema de Software

Este sub-processo tem a finalidade de gerar os documentos necessários para descrever os requisitos do sistema de software a ser desenvolvido. Para isso, o elaborador de requisitos gera um conjunto de documentos de entrada. A Figura 40 apresenta o diagrama do sub-processo obter os requisitos do sistema de software.

Figura 40 – Sub-processo obter os requisitos do sistema de software (1).

O diagrama prevê a elaboração dos documentos de requisitos pelo elaborador de requisitos, utilizando quaisquer técnicas de elicitação de requisitos. O elaborador de requisitos pode ser o cliente ou o responsável pela elaboração dos requisitos do sistema de software.

Os principais documentos de entrada são constituídos pelo modelo de casos de uso (composto pelo diagrama e especificação dos casos de uso), especificação de requisitos não funcionais com suas respectivas prioridades, complementados pelo glossário e documentos existentes, como: *workload* do sistema, o *workload* organizacional, fluxo de informações, especificações de requisitos funcionais, prioridades organizacionais, entre outros. A descrição dos artefatos de entrada se encontra no Apêndice B.

O engenheiro de requisitos deve executar a atividade verificar documentação de entrada (1.1) a fim de analisar se os documentos recebidos apresentam o conteúdo adequado, através dos padrões estabelecidos no Apêndice B. Em função dos resultados da verificação, os documentos podem ser aceitos ou rejeitados. Em caso de rejeição, o engenheiro de requisitos solicita a adequação ao elaborador de requisitos, para que os documentos possam ser reapresentados conforme padrões estabelecidos.

4.4 Sub-Processo 2: Relacionar Interesses Funcionais e Não-Funcionais

Este sub-processo tem como objetivo identificar interesses funcionais e não-funcionais e associá-los no diagrama de casos de uso do sistema de software. Neste sub-processo o método proposto por Supakkul e Chung (2004) descrito na seção 2.3.2 é detalhado para que seja incorporado ao PARNAFOA. Este método define um conjunto de regras para associar elementos do diagrama de casos de uso e *softgoals*, e analisar os impactos dos *softgoals* através dos elementos do diagrama de casos de uso.

As atividades deste sub-processo são apresentadas na Figura 41.

Figura 41 – Sub-processo relacionar interesses funcionais e não-funcionais (2).

Os interesses não-funcionais são separados na atividade para identificar interesses funcionais e não-funcionais (2.1). Um conjunto de RNFs, tais como desempenho, segurança e confiabilidade é considerado um interesse não-funcional.

O catálogo de tipos de RNFs do NFR *Framework* (CHUNG et al., 2000) é utilizado para classificar interesses não-funcionais. Porém, como este catálogo é restrito aos tipos de requisitos de acurácia, segurança e desempenho, são utilizadas a norma ISO/IEC 9126 (ISO/IEC 9126, 2001), com as características e sub-características de qualidade, e a lista de RNFs proposta por Chung et al. (2000) para estender a lista de tipos de RNFs disponíveis. O uso de uma lista mais abrangente de tipos de requisitos não-funcionais, que facilite a classificação dos interesses não-funcionais, é uma das contribuições deste trabalho.

Os interesses não-funcionais identificados para o sistema de software são representados como *softgoals*. Desta forma, um *softgoal* representa um interesse não-funcional, que por sua vez, pode ser composto de um ou mais RNFs.

Os interesses funcionais, por sua vez, são obtidos através do modelo de casos de uso. Cada um dos casos de uso representa um interesse funcional do sistema de software a ser desenvolvido.

A saída da atividade para identificar interesses funcionais e não-funcionais (2.1) são os *softgoals*, a lista de interesses funcionais e não-funcionais, e a

tabela de requisitos não-funcionais (RNF) com os interesses não-funcionais associados a cada um dos RNFs.

Durante a atividade para identificar interesses funcionais e não-funcionais, os *softgoals* possuem somente a definição de tipo de RNF. A lista de interesses possui a relação de interesses do sistema de software e a classificação de acordo com sua natureza, que pode ser funcional ou não-funcional. A tabela de RNF possui os RNFs relacionados aos interesses não-funcionais e a definição da origem da classificação do interesse, que pode ser proveniente do catálogo de tipo de RNFs, da lista de Chung et al. (2000) ou das características e sub-características de qualidade da norma ISO/IEC 9126 (ISO/IEC 9126, 2001).

A atividade para integrar interesses funcionais e não-funcionais (2.2) considera os dados provenientes da atividade 2.1 e utiliza o método de Supakkul e Chung (2004). No PARNAFOA, as heurísticas para a associação de um *softgoal* a um elemento do diagrama de casos de uso do método original foram detalhadas. Para o detalhamento, foi considerada a classificação de requisitos não-funcionais proposta por Kotonya e Sommerville (1998), apresentada na seção 2.1 deste trabalho. O detalhamento das heurísticas para associação de um *softgoal* a um elemento do diagrama de casos de uso é uma das contribuições deste trabalho.

A classificação de Kotonya e Sommerville (1998) foi escolhida pois faz a diferenciação entre RNFs externos, de produto e de processo. Esta distinção é necessária para que os RNFs externos sejam associados aos elementos externos do sistema de software (atores), e para que os requisitos de produto e processo possam ser associados aos demais elementos como: casos de uso, relação atores e casos de uso e limite do sistema.

As heurísticas definidas para associar um *softgoal* a um elemento do diagrama de casos de uso são listadas a seguir:

- O ponto de associação entre um *softgoal* e um ator deve ocorrer quando existirem requisitos não-funcionais externos ou restrições operacionais aplicáveis ao ator;
- O ponto de associação entre um *softgoal* e um caso de uso deve ocorrer quando existirem requisitos de produto aplicáveis ao caso de uso;
- O ponto de associação entre um *softgoal* e o relacionamento ator-caso de uso deve ocorrer quando existirem requisitos de acesso ao sistema

de software, comunicação, interface entre o ator e o sistema de software, ou troca de informação entre o sistema de software e os atores;

- O ponto de associação entre um *softgoal* e o limite do sistema de software deve ocorrer quando existirem requisitos externos, de processo, ou de produto considerados para o sistema de software como um todo;

Os SIGs iniciais referentes ao sistema de software são gerados no momento em que os interesses funcionais (casos de uso) e não-funcionais (*softgoals*) são associados entre si no diagrama integrado de casos de uso. O diagrama integrado terá a representação conforme descrito na seção 2.3.2.

A estrutura de um SIG inicial do diagrama integrado é compatível com a nomenclatura tipo [tópico], onde tipo representa um tipo de requisito não-funcional e o tópico representa o contexto ou qualificação para o tipo (CHUNG et al., 2000). A saída deste sub-processo deve ser o diagrama integrado que contém os SIGs iniciais associados aos elementos do diagrama de casos de uso.

Os documentos existentes e o glossário, fornecidos como entrada, são utilizados para consulta, conforme necessidade.

4.5 Sub-Processo 3: Identificar Aspectos

Este sub-processo tem a finalidade de identificar interesses transversais (aspectos) e não-transversais a partir dos RNFs do sistema de software. Cada RNF que compõe um interesse não-funcional deve ser analisado. Neste sub-processo é utilizado o método de Supakkul e Chung (2004), descrito na seção 2.3.2, para analisar as propagações de um requisito não-funcional nos elementos do modelo de casos de uso.

A representação do sub-processo se encontra na Figura 42.

Figura 42 – Sub-processo identificar aspectos (3).

Na atividade de identificação de propagações (3.1) é utilizado o método de Supakkul e Chung (2004). Para utilização deste método no PARNAFOA suas regras de propagação foram detalhadas considerando a proposta original de Supakkul e Chung (2004). Este detalhamento tem a finalidade de auxiliar a detecção de aspectos. O detalhamento das regras de propagação, propostas originalmente por Supakkul e Chung (2004), é uma das contribuições deste trabalho. As novas regras de propagação são apresentadas a seguir:

- Regra de propagação através do ator: as relações de especialização e generalização entre atores devem ser avaliadas a fim de identificar possíveis propagações de um RNF associado a um ator.
- Regra de propagação através do caso de uso: As relações de inclusão, extensão e generalização entre os casos de uso devem ser avaliadas a fim de identificar possíveis propagações de um RNF associado a um caso de uso:
 - Uma relação de inclusão pode não propagar um RNF, como foi estabelecido por Supakkul e Chung (2004). Um exemplo desta situação é referente ao RNF de *safety*, que em muitas situações, costuma ser específico a uma determinada função relacionada a um caso de uso, não propagando seus RNFs.
 - Uma relação de extensão pode ou não propagar um requisito não-funcional, pois isso dependerá do tipo do RNF em questão. Um exemplo desta situação é referente ao RNF desempenho, que pode propagar o RNF através de um caso de uso de extensão, caso este participe da execução de uma função que

tenha uma restrição especificada de tempo. Porém, se na mesma situação, um caso de uso de extensão tiver o objetivo de calcular médias de valores para que os dados sejam armazenados como dados históricos, este RNF não será propagado através do caso de uso de extensão.

- A propagação de um RNF em relações de generalização mostrou-se sempre válida, porém, somente serão encontradas em representações UML anteriores à versão 2 (OMG-UML, 2009). Isso ocorre porque o caso de uso abstrato não possui a seqüência completa do fluxo de eventos de um caso de uso concreto; isso faz com que o caso de uso abstrato propague o RNF a ele associado.
- Regra de propagação através da relação ator-caso de uso: devem ser avaliadas as propagações entre atores generalizados ou especializados, quando ocorrerem, e entre relacionamentos entre seus respectivos casos de uso, incluídos, estendidos ou generalizados quando existir um RNF associado a uma relação ator-caso de uso.
- Regra de propagação através do limite de sistema: um RNF quando associado ao limite do sistema, pode ser propagado para o sistema de software como um todo, para um ou mais casos de uso, para um ou mais atores do sistema de software ou para uma ou mais relações atores-casos de uso.
- Deve existir conhecimento sobre os requisitos funcionais do sistema de software. Propagações podem ocorrer mesmo sem que exista relacionamento entre casos de uso, atores ou relação ator-caso de uso em que um RNF estiver associado. Um exemplo desta situação está relacionado ao RNF de acurácia. Se for estabelecido que uma determinada informação deve possuir acurácia, e esta informação for manipulada por diversos casos de uso, a propagação deste RNF irá ocorrer mesmo sem que os casos de uso estejam relacionados.

A análise das propagações dos RNFs gera a tabela de RNF com propagações. Nesta tabela deve ser identificado o ponto de associação relacionado a cada um dos RNFs no diagrama integrado e a propagação

propriamente dita, na qual a lista dos elementos afetados pelo requisito não-funcional deve ser elaborada.

A partir desta análise deve ser executada a atividade de identificação de aspectos (3.2). Esta atividade conta com as seguintes heurísticas:

- Quando um requisito não-funcional estiver associado a mais de um elemento do modelo de caso de uso, mesmo que não ocorra propagação, este requisito é considerado um aspecto.
- Quando um requisito não-funcional for propagado para mais de um elemento do diagrama de caso de uso, este requisito é considerado um aspecto.
- Quando um requisito não-funcional estiver associado ao limite do sistema, este requisito é considerado um aspecto, pois poderá afetar diversos casos de uso ou mesmo diversos artefatos do sistema de software em fases posteriores do desenvolvimento.

A definição das heurísticas para que aspectos possam ser identificados, a partir da análise dos pontos de associação e das propagações, é uma contribuição deste trabalho.

A saída deste sub-processo é a tabela de RNF com aspectos incluindo a definição de aspecto para cada um dos RNFs do sistema de software em questão.

4.6 Sub-Processo 4: Decompor Requisitos Não-Funcionais

Este sub-processo tem a finalidade de decompor e analisar requisitos não-funcionais através de SIGs. Neste sub-processo é utilizado o NFR *Framework* (CHUNG et al., 2000) para decompor os SIGs iniciais presentes no diagrama integrado. Detalhamentos do NFR *Framework* relacionados à decomposição de tipo, tópico e de operacionalizações são definidos nas seções 4.6.1 e 4.6.2 respectivamente. Estes detalhamentos propostos para o NFR *Framework* são contribuições deste trabalho.

A representação deste sub-processo é apresentada na Figura 43.

Figura 43 – Sub-processo decompor requisitos não-funcionais (4).

As atividades 4.2 a 4.9 devem ser executadas conforme descrições da seção 2.2.7. O mapeamento entre os passos do NFR *Framework* definidos na seção 2.2.7 e o sub-processo do PARNAFOA é apresentado na Tabela 11.

Tabela 11 – Mapeamento entre os passos do NFR *Framework* e o sub-processo 4 do PARNAFOA.

Atividades do NFR <i>Framework</i> (seção 2.2.7)	Atividades do PARNAFOA (sub-processo 4)
Passo 4: Decomposição dos <i>softgoals</i> dos SIGs iniciais.	Atividade 4.2: Decompor <i>softgoals</i> .
Passo 5: Definição de prioridades dos <i>softgoals</i> .	Atividade 4.3: Definir prioridades.
Passo 6: Identificação de possíveis operacionalizações.	Atividade 4.4: Identificar operacionalizações.
Passo 7: Definição das interdependências implícitas entre <i>softgoals</i> .	Atividade 4.5: Identificar correlações.
Passo 8: Registro de decisões de projeto.	Atividade 4.6: Registrar decisões de projeto.
Passo 9: Seleção de alternativas de projeto.	Atividade 4.7: Selecionar alternativas de projeto.
Passo 10: Avaliação do impacto das decisões.	Atividade 4.8: Aplicar o procedimento de avaliação.

Os passos iniciais do NFR *Framework* a respeito da aquisição do conhecimento do domínio, aquisição e catalogação sobre os RNFs e

identificação dos RNFs do sistema de software não são consideradas no sub-processo 4 pois são atividades que foram executadas no sub-processo (1), para obter os requisitos do sistema de software.

A ordem de execução das atividades 4.2 a 4.7, e as atividades que devem ou não ser executadas, podem variar, dependendo do tipo de sistema de software. As atividades para identificar correlações e definir prioridades podem ser omitidas, caso não existam correlações a serem registradas nos SIGs e caso não exista necessidade de identificar prioridades explicitamente. Em outra situação, a atividade para definir prioridade pode ser executada antes da atividade para decompor *softgoals*, caso o RNF seja crítico para o sistema de software.

A atividade para selecionar um SIG inicial a ser detalhado (4.1) estabelece o SIG a ser detalhado a partir das prioridades definidas na especificação de RNFs. A priorização dos SIGs faz com que seja possível desenvolvê-los iterativamente (JACOBSON; BOOCH; RUMBAUGH, 1999). A saída da atividade 4.1 será o SIG inicial a ser detalhado.

As atividades 4.2 a 4.7 recebem, como entrada, a tabela de RNF com aspectos, para que as propagações possam ser representadas nos SIGs e a especificação dos requisitos não-funcionais, caso seja necessário consultá-la.

Cada uma das atividades de 4.2 a 4.7 detalha o SIG incorporando novas informações.

A atividade para decompor *softgoals* (4.2) inclui a decomposição de tipo e tópico dos requisitos não-funcionais, gerando como saída o SIG decomposto.

A atividade para definir prioridades (4.3) inclui prioridades no SIG, que podem estar relacionadas ao tipo ou tópico do requisito não-funcional. A saída desta atividade é o SIG com prioridades.

A atividade para identificar operacionalizações (4.4) inclui no SIG as estratégias de implementação a serem consideradas para o requisito não-funcional. A saída desta atividade é o SIG com operacionalizações.

A atividade para identificar correlações (4.5) identifica conflito ou harmonia entre requisitos não-funcionais através de interdependências implícitas. A saída desta atividade é o SIG com correlações.

A atividade para registrar as decisões de projeto (4.6) identifica argumentos dos desenvolvedores e análises de *tradeoffs* entre *softgoals*. A saída desta atividade é o SIG com decisões.

A atividade para selecionar alternativas de projeto (4.7) inclui as definições sobre os *softgoals* que serão aceitos ou rejeitados. A saída desta atividade é o SIG com alternativas, com os *softgoals* de operacionalização selecionados.

A atividade para aplicar o procedimento de avaliação (4.8) executa os passos do procedimento de avaliação e registra os rótulos nos *softgoals*. A saída desta atividade é o SIG avaliado.

A atividade para definir relacionamentos de metas de operacionalizações (4.9) registra no SIG as metas e os relacionamentos de metas de operacionalizações. A saída desta atividade é o SIG com o seu detalhamento finalizado. Durante esta atividade, caso mais de uma meta tenha sido definida para um mesmo RNF, recomenda-se o desdobramento deste requisito para cada uma das metas. A tabela de RNF deve ser revisada a partir do sub-processo 1 (obter os requisitos do sistema de software).

4.6.1 Detalhamento da Atividade para Decompor *Softgoals*

Na atividade para decompor *softgoals* (4.2) do PARNAFOA são propostos dois tipos de detalhamentos para o SIG. O primeiro detalhamento refere-se à decomposição de tipo de RNF e o segundo refere-se à decomposição de tópico de RNF.

Na decomposição de tipo de RNF, foram considerados o catálogo de tipos de RNFs e a norma ISO/IEC 9126 (ISO/IEC 9126, 2001). Como o catálogo de tipos de RNFs do NFR *Framework* é restrito aos RNFs de acurácia, segurança e desempenho, as características e sub-características da norma foram utilizadas, neste trabalho, para ampliar os tipos e sub-tipos de RNFs propostos por Chung et al. (2000).

Os tópicos representam, neste trabalho, os elementos do diagrama de casos de uso. Portanto, na decomposição de tópico de RNFs, foram criados métodos com a finalidade de representar, no SIG, a propagação de um RNF nos elementos do diagrama de casos de uso.

Foram definidos neste trabalho os métodos de decomposição de limite de sistema, de casos de uso, de ator e de relação ator-caso de uso. A Figura 44 apresenta os métodos de decomposição de tópico de requisitos não-funcionais para o PARNAFOA.

Figura 44 – Métodos de decomposição de tópicos para o PARNAFOA.

A Figura 44 (a) representa o método de decomposição de limite de sistema. Um RNF associado ao limite do sistema pode ser propagado para: casos de uso, atores ou relações atores-casos de uso.

A Figura 44 (b) representa o método de decomposição de caso de uso. Um RNF pode ser propagado para: casos de uso incluídos, estendidos, especializados ou generalizados.

A Figura 44 (c) representa o método de decomposição de ator. Um RNF pode ser propagado para: atores especializados ou generalizados.

A Figura 44 (d) representa o método de decomposição da relação ator-caso de uso. Um RNF pode ser propagado para: ator especializado, ator generalizado, caso de uso incluído, caso de uso estendido, caso de uso generalizado ou caso de uso especializado.

Dependendo da propagação de um RNF, mais de um método de decomposição pode ser utilizado em um mesmo SIG.

Um exemplo do método de decomposição de limite de sistema pode ser observado na Figura 45.

Figura 45 – Método de decomposição de limite de sistema.

A Figura 45 (a) representa o método de decomposição de limite de sistema. A Figura 45 (b) apresenta o diagrama integrado que associa o *softgoal* ao limite de sistema. Uma análise deve ser feita para avaliar a propagação do requisito não-funcional. Um ou mais casos de uso podem ser afetados. O SIG inicial é apresentado na Figura 45 (c) e, supondo que ocorra uma propagação em ambos os casos de uso do sistema de software, o SIG resultante, na Figura 45 (d) apresentará os tópicos do SIG preenchidos com os casos de uso 14 e 15.

A Figura 46 apresenta um exemplo do método de decomposição para casos de uso.

Figura 46 – Método de decomposição de caso de uso.

A Figura 46 (a) representa o método de decomposição de caso de uso. A Figura 46 (b) mostra o diagrama integrado que associa o *softgoal* a um caso de uso. Uma análise deve ser feita para avaliar a propagação do RNF, que pode afetar um ou mais casos de uso. O SIG inicial é apresentado na Figura 46 (c), e supondo que ocorra uma propagação no caso de uso 7 incluído e no caso de uso especializado 4, o SIG resultante, na Figura 46 (d) apresentar os tópicos do SIG preenchidos com os casos de uso 7 e 4.

Um exemplo de método de decomposição de ator é apresentado na Figura 47.

Figura 47 – Método de decomposição de ator.

A Figura 47 (a) representa o método de decomposição de ator. A Figura 47 (b) mostra o diagrama integrado que associa o *softgoal* a um ator. Uma análise deve ser feita para avaliar a propagação do RNF que pode afetar um ou mais atores. O SIG inicial é apresentado na Figura 47 (c) e, supondo que ocorra a propagação para os atores especializados 3 e 4 e para o ator generalizado 1, o SIG resultante, na Figura 47 (d) apresentará os tópicos do SIG preenchidos com os atores 1, 3 e 4.

A Figura 48 apresenta um exemplo de método de decomposição de relação ator-caso de uso.

Figura 48 – Método de decomposição relação ator-caso de uso.

A Figura 48 (a) representa o método de decomposição da relação ator-caso de uso. A Figura 48 (b) mostra o diagrama integrado que associa o *softgoal* a uma relação ator-caso de uso. Uma análise deve ser feita para avaliar a propagação do RNF que pode afetar um ou mais atores e casos de uso. O SIG inicial é apresentado na Figura 48 (c) e, supondo que ocorra a propagação para os atores generalizado 6 e especializado 8 e para o caso de uso incluído 13, o SIG resultante, na Figura 48 (d) apresenta os tópicos do SIG preenchidos com os atores 6 e 8, e com o caso de uso 13.

4.6.2 Detalhamento da Atividade para Identificar Operacionalizações

Na atividade para identificar operacionalizações (4.4) foi feita uma proposta de inclusão de novas operacionalizações no catálogo de operacionalização do

NFR *Framework* que é restrito aos requisitos de acurácia, segurança e desempenho.

Esta proposta consiste da incorporação das táticas definidas por Bass, Clements e Kazman (2003) como sugestões de operacionalizações. Táticas são decisões de projeto para introduzir atributos de qualidade ao projeto de arquitetura de software. Analisando-se o contexto, pode-se considerar uma tática como um *softgoal* de operacionalização, uma vez que ambas representam estratégias de implementação a serem selecionadas para sistemas de software.

Assim como o NFR *Framework* realiza análises de conflito e harmonia (correlações negativas e positivas), análises equivalentes são feitas por Bass, Clements e Kazman (2003) para escolha de determinadas táticas.

Desta forma, considerou-se que as táticas possam ser utilizadas para enriquecer o catálogo de operacionalizações.

A Tabela 12 apresenta as principais operacionalizações de Chung et al. (2000) e táticas de Bass, Clements e Kazman (2003).

Tabela 12 – Operacionalizações e táticas utilizadas na decomposição de operacionalização.

Requisito Não-Funcional	Estratégia	Operacionalização	Tática
Acurácia	Manipulação de exceção	X	
	Atribuição de recurso	X	
	Validação	X	
	Reforço de Justificativa	X	
	Verificação	X	
	Ponto de Verificação	X	
	Melhor Fluxo de Informação	X	
	Autorização	X	
	Certificação	X	
	Assistência de Rastreamento	X	
	Verificação Cruzada	X	
	Confirmação	X	
	Verificação de Consistência	X	
	Auditoria	X	
	Reforço de Autenticação	X	
Segurança	Perturbação	X	
	Filtro de Vírus	X	
	Autorização de Acesso	X	
	Auditoria	X	
	Alarme	X	
	Encriptação	X	
	Postagem Rápida	X	
	Autenticar Usuários		X
	Autorizar Usuários		X
	Manter a Confidencialidade dos Dados		X
	Manter a Integridade		X
	Limitar a Exposição		X
	Limitar o Acesso		X

Requisito Não-Funcional	Estratégia	Operacionalização	Tática
	Detecção de Intrusão		X
	Identificação		X
	Restauração		X
Desempenho	Correção Antecipada	X	
	Determinação Estática de Valor	X	
	Indexação	X	
	Descompressão	X	
	Correção Tardia	X	
	Determinação Dinâmica de Valor	X	
	Compressão	X	
	Redução de Reorganização em Tempo de Execução	X	
	Executar Primeiro	X	
	Executar mais Cedo	X	
	Executar mais Tarde	X	
	Executar por Último	X	
	Poucos Atributos por Tupla	X	
	Vários Atributos por Tupla	X	
	Replicação de Atributo Derivado	X	
	Acessar Muitos Atributos por Tupla	X	
	Acessar muitas Tuplas por Relação	X	
	Herança de Código Estático	X	
	Herança de Código Dinâmico	X	
	Planejamento / Programação	X	
	Verificação Seletiva	X	
	Verificação Cíclica	X	
	Gatilho	X	
	Aumentar a Eficiência Computacional		X
	Reduzir a Sobrecarga Computacional		X
	Gerenciar a Taxa de Eventos		X
	Controlar a Frequência de Amostragem		X
	Introduzir Concorrência		X
	Manter Múltiplas Cópias		X
	Aumentar a Disponibilidade de Recursos		X
	Política de Planejamento / Programação		X
Disponibilidade	Ping		X
	Temporizador de Homem Morto		X
	Exceções		X
	Votação		X
	Redundância Ativa		X
	Redundância Passiva		X
	Reserva		X
	Operação em Sombra		X
	Resincronização de Estado		X
	Ponto de Verificação / Retomada		X
	Remoção do Serviço		X
	Transações		X
	Monitor de Processo		X
Modificabilidade	Coerência Semântica		X
	Antecipar Mudanças Previstas		X
	Generalizar Módulos		X
	Limitar Opções Possíveis		X
	Abstrair Serviços Comuns		X
	Esconder Informação		X
	Manter a Interface Existente		X
	Restringir Caminhos de Comunicação		X
	Usar um Intermediário		X
	Registro em Tempo de Execução		X
	Arquivos de Configuração		X
	Polimorfismo		X
	Substituição de Componentes		X
	Aderência a Protocolos Definidos		X

Requisito Não-Funcional	Estratégia	Operacionalização	Tática
Testabilidade	Armazenar / Reproduzir		X
	Separar Interface da Implementação		X
	Acesso Especializado a Rotinas / Interfaces		X
	Monitores Embutidos		X
Usabilidade	Separar Interface do Usuário		X
	Suporte à Iniciativa do Usuário		X
	Suporte à Iniciativa do Sistema		X

4.7 Sub-Processo 5: Elaborar o Modelo Composto

Este sub-processo tem o objetivo de incorporar o resultado da decomposição dos RNFs, através dos SIGs, ao modelo de casos de uso. Sua representação é apresentada na Figura 49.

Figura 49 – Sub-processo elaborar modelo composto (5).

A atividade para identificar metas operacionais e restrições de projeto (5.1) tem a finalidade de classificar as metas identificadas nos SIGs em metas operacionais ou restrições de projeto.

As metas operacionais são novas funções que devem ser incorporadas ao sistema de software. As restrições de projeto são impactam nas atividades do desenvolvimento e contribuem para que uma meta operacional possa ser satisfeita; são exemplos de restrições de projeto a infra-estrutura de hardware, restrições de implementação, documentação ou gerenciamento de projeto.

As metas operacionais devem ser incorporadas ao modelo de casos de uso do sistema de software através de novos casos de uso. A identificação de restrições de projeto deve sugerir restrições a serem impostas ao sistema de

software. Estas informações devem ser atualizadas na tabela de RNF com metas operacionais e restrições de projeto.

Nesta tabela deve ser identificado:

- Número seqüencial atribuído ao SIG que originou a meta;
- Texto da meta;
- Classificação da meta em: meta operacional ou restrição de projeto;
- Origem da meta, que pode ser: NFR (proveniente do NFR *Framework*), tática (proveniente da proposta de Bass, Clements e Kazman (2003)) ou definida pelo engenheiro de requisitos (caso não existam catálogos disponíveis relacionados ao tipo de RNF a ser detalhado);
- Nome do caso de uso a ser incluído no modelo de casos de uso, quando se tratar de meta operacional ou a descrição da restrição de projeto a ser imposta ao sistema de software, quando se tratar de restrição de projeto.

Recomenda-se que cada meta operacional seja definida em apenas um caso de uso pois, se mais de uma meta operacional for definida em um único caso de uso são necessários controles adicionais para identificá-las.

A atividade para definir pontos de composição (5.2) utiliza regras para a composição através dos operadores de sobreposição, substituição e empacotamento, definidos no modelo de engenharia de requisitos orientado a aspectos com UML (ARAÚJO et al., 2002), apresentado na seção 3.3.1.

A definição da tabela de RNF com composição foi baseada na proposta da abordagem direcionada a casos de uso para o desenvolvimento de software orientado a aspectos (SOUSA, 2004), descrita na seção 3.3.5.

Esta tabela possui:

- Identificação do caso de uso que recebe a composição;
- Passo do seu fluxo básico ou alternativo do caso de uso em que a composição é feita;
- Regra de composição a ser aplicada;
- Condição da composição, quando se tratar de substituição ou empacotamento. No caso da substituição, deve ser definida a ação que substitui o fluxo básico ou alternativo do caso de uso que recebe a

composição. No caso do empacotamento, deve ser definida a situação em que a inclusão deve ocorrer.

Somente metas operacionais devem ser compostas no modelo de casos de uso, por representarem funções a serem adicionadas no sistema de software. Restrições de projeto somente terão sua descrição definida na tabela de RNF com meta operacional ou restrição de projeto.

A partir do momento em que os pontos de composição estão definidos e a tabela de RNF com composição está preenchida, novas propagações, que antes não foram detectados podem ser identificadas. Uma confirmação das propagações deve ser feita na atividade confirmar propagação (5.3), e caso exista necessidade de atualização das propagações, esta atividade deve ser remetida para o sub-processo de identificação de aspectos, onde a tabela de RNF com propagações e o SIG, devem ser atualizados e o processo reiniciado a partir deste ponto, como mostra o *link* B na Figura 49. A atividade 5.3 pode não ser executada caso não exista necessidade de revisão das propagações.

A atividade para resolver de conflitos (5.4) deve ser iniciada a partir da tabela de RNF com composição. A análise de conflitos tem a finalidade de identificar pontos de composição que coincidam no mesmo passo de um cenário de um caso de uso, com a mesma regra de composição. Estes conflitos são identificados e a tabela de RNF com prioridades é preenchida.

Esta tabela possui a definição da prioridade de composição e, o menor valor da atribuído na coluna de prioridade da composição representa a prioridade mais alta.

A atividade para atualizar o modelo do sistema de software (5.5) com as informações da composição encerra este sub-processo. Novos casos de uso são adicionados no diagrama de casos de uso através do relacionamento de inclusão. Esta definição é baseada na proposta do modelo de atributos de qualidade transversais para a engenharia de requisitos, descrita na seção 3.3.2 deste trabalho.

Um estereótipo <<aspecto>> deve ser incluído no novo caso de uso, caso o requisito não-funcional representado seja um aspecto.

É importante salientar que requisitos não-funcionais que não são identificados como aspectos são também tratados por este sub-processo. Os

operadores de composição, bem como a inclusão de novos casos de uso são também contemplados. Neste caso estereótipos não são utilizados.

Caso ocorra o inter-relacionamento entre requisitos não-funcionais, isto é, caso exista necessidade de relacionar os novos casos de uso entre si, a definição das especificações dos novos casos de uso deve ser concluída para que a tabela de RNF com composição seja preenchida. Esta situação é apresentada na atividade 5.5, como mostra o *link C* da Figura 49.

A saída deste sub-processo consiste do modelo de casos de uso atualizado, a tabela de RNF com prioridades e o diagrama integrado. O modelo de casos de uso recebido como dado de entrada no PARNAFOA não é alterado. A informação sobre o passo do fluxo básico ou alternativo em que ocorre a composição dos novos casos de uso fica disponível somente na planilha de RNF com prioridades.

4.8 Considerações Finais

A definição inicial do PARNAFOA foi revisada até chegar nesta versão que foi apresentada. Informações para que a revisão pudesse ser realizada foram coletadas à medida que o processo foi aplicado e avaliado em cinco sistemas de software com características diversas. Foi percebida também a evolução na integração dos métodos e adequação dos detalhes dos métodos originais, utilizados no PARNAFOA.

A definição do PARNAFOA considerou abordagens e processos propostos na literatura, bem como a integração e detalhamento de métodos relacionados aos RNFs e aspectos. As abordagens e processos integrados ao PARNAFOA e o detalhamento dos métodos utilizados neste processo foram contribuições deste trabalho.

5. Aplicações do PARNAFOA

“Saber não é suficiente; devemos aplicar. Querer não é suficiente; devemos fazer.” – Johann von Goethe

O PARNAFOA foi aplicado em cinco sistemas de software com domínio, porte e complexidade diversos. São eles: sistema de gerenciamento de hotel (SGH), sistema de bóias meteorológicas (SBM), sistema olhos no céu (ONC), sistema de controle automático de trens (ATC – *Automatic Train Control*) e sistema de gerenciamento de crise (SGC). Este capítulo tem o objetivo de apresentar os pontos relevantes destas aplicações.

A estrutura de apresentação das aplicações segue os sub-processos, conforme a descrição do PARNAFOA no capítulo 4.

Os diagramas de casos de uso e os SIGs desenvolvidos para as cinco aplicações foram elaborados através da ferramenta StarUML (LEE et al., 2005), que foi obtida durante a interação com o pesquisador Sam Supakkul.

A descrição detalhada dos artefatos utilizados e dos procedimentos para a execução das atividades encontra-se nos Apêndices B e C, respectivamente.

Os termos *security* e *safety* foram mantidos em inglês, pois a tradução de ambos em português é segurança, o que não é adequado.

As aplicações do PARNAFOA foram analisadas através da resposta a um conjunto de perguntas que procuraram ressaltar pontos importantes:

1. Quantos RNFs existem na aplicação?
2. Quantos casos de uso existem na aplicação?
3. Quantos atores existem na aplicação?
4. Quantos *softgoals* foram identificados na aplicação?
5. Houve necessidade de adequação dos dados de entrada para a aplicação?
6. Quanto tempo foi gasto na execução do PARNAFOA para a aplicação?
7. Qual a contribuição de cada aplicação para a melhoria do PARNAFOA?
8. Quais tipos de propagação ocorreram em cada aplicação?
9. Quais tipos de pontos de associação ocorreram em cada aplicação?
10. Ocorreram propagações em casos de uso que possuem relacionamento de inclusão?

11. Ocorreram propagações em casos de uso que possuem relacionamento de extensão?
12. Ocorreram propagações em casos de uso que possuem relacionamento de generalização?
13. Quantos aspectos relacionados aos RNFs foram detectados na aplicação?
14. Houve necessidade de desmembramento dos RNFs da aplicação?
15. Houve necessidade de revisão das propagações dos RNFs na aplicação?
16. Quantas restrições de projeto foram definidas para a aplicação?
17. Quantas metas operacionais foram definidas para a aplicação?

A avaliação de cada uma das aplicações, com as respostas a estas questões, está apresentada na seção 5.6 deste capítulo.

5.1 Sistema de Gerenciamento de Hotel

O sistema de gerenciamento de hotel (SGH) é baseado em um exemplo de Jacobson e Ng (2004). Este exemplo foi escolhido por ser de domínio conhecido e ter característica de um sistema de informação.

As principais funções do SGH são: reserva de quartos, efetivação do *check in* e *check out* de clientes, definição de taxas e ofertas promocionais durante períodos pré-definidos pela gerência do hotel, controle de pontos de fidelidade dos clientes para serem utilizados no momento do pagamento, controle de lista de espera quando todos os quartos estiverem reservados ou ocupados e manipulação de diferentes tipos de clientes (individuais, corporativos e membros). Para simplificar, a reserva pode ser feita apenas pelo cliente.

5.1.1 Sub-processo 1: Obter os requisitos do sistema de software

O sub-processo 1 é composto pela atividade de elaborar os requisitos e pela atividade de verificar a documentação de entrada (1.1).

Foi utilizada a descrição do sistema de gerenciamento de hotel de Jacobson e Ng (2004), que foi considerado como documento adicional; esta descrição

inclui o diagrama de casos de uso, partes das especificações dos casos de uso e especificação de requisitos não-funcionais. Após verificar a documentação de entrada (1.1), foi executada a atividade de elaborar os requisitos. Esta atividade consistiu do levantamento do glossário e revisão do modelo de casos de uso para completar os casos de uso faltantes.

O glossário do SGH é apresentado na Tabela 13.

Tabela 13 – Glossário (SGH).

Termo	Definição
Reserva	Período planejado para um quarto para permanência de um hóspede no hotel.
Check in	Chegada do cliente chega no hotel. O período da sua permanência tem início. Um quarto é alocado para um cliente. O quarto fica com o estado ocupado.
Check out	Final da permanência do cliente no hotel. O quarto passa para o estado disponível. O cliente deve fazer o pagamento referente aos seus gastos. Créditos de fidelidade podem ser utilizados para reduzir os gastos.
Créditos de Fidelidade	Pontos acumulados pelo cliente a cada permanência no hotel. Estes pontos podem ser utilizados para reduzir os gastos na próxima estadia no hotel.

O diagrama de casos de uso do SGH, após revisão, é apresentado na Figura 50.

Figura 50 – Diagrama de casos de uso (SGH).

Os atores do SGH são os seguintes:

- Cliente: é responsável por reservar o quarto.
- Membro: é um tipo de cliente que pode ser individual ou corporativo.
- Equipe do Hotel: é responsável por registrar os membros, manipular lista de espera, realizar *check in* e *check out* de clientes.

- Gerência do Hotel: é responsável por estabelecer as taxas dos quartos e as taxas promocionais.

O caso de uso *Reservar Quarto* recebe os dados da reserva, procura um quarto compatível no período desejado e marca como reservado. No caso de indisponibilidade do quarto, a reserva é inserida na lista de espera.

O caso de uso *Manipular Lista de Espera* realiza a inserção de uma reserva na lista de espera por um quarto.

O caso de uso *Registrar Membro* realiza a inserção, consulta, atualização e exclusão de dados do cliente no sistema.

O caso de uso *Check in* recebe os dados necessários para alocar o cliente no quarto e modifica o estado do quarto para ocupado.

O caso de uso *Check out* recebe os dados do quarto a ser liberado atualiza o estado do quarto para disponível.

O caso de uso *Ganhar e Resgatar Créditos* calcula os créditos do cliente após o *check out* e emite a nota de pagamento das despesas.

O caso de uso *Estabelecer as Taxas do Quarto* define ou atualiza as taxas dos quartos.

O caso de uso *Estabelecer as Taxas de Promoção* define ou atualiza as taxas promocionais relacionadas aos quartos.

A Tabela 14 apresenta um exemplo da especificação do caso de uso *Reservar Quarto*. A descrição dos seus campos encontra-se no Apêndice B. As demais especificações do SGH encontram-se em Bombonatti (2010).

Tabela 14 – Especificação do caso de uso Reservar Quarto (SGH).

Nome do Caso de Uso	Reservar Quarto
Descrição Breve	Este caso de uso procura um quarto compatível no período solicitado e marca como reservado.
Atores	Cliente.
Evento Inicial	O cliente seleciona o comando de reserva de quarto.
Fluxo Básico	<ol style="list-style-type: none"> 1. O cliente seleciona o comando de reserva de quarto. 2. O sistema exibe os tipos de quarto e as taxas. 3. O cliente verifica a taxa de cada um dos quartos. 4. O cliente solicita uma reserva para o quarto selecionado. 5. O sistema busca um quarto disponível e decrementa o número de quartos disponíveis do tipo selecionado. {Atualizar Disponibilidade do Quarto} 6. O sistema cria uma nova reserva com os detalhes fornecidos. 7. O sistema exibe o número de confirmação da reserva e as instruções para o cliente. 8. O caso de uso termina.

Fluxos Alternativos	5A. Submissão duplicada: se no passo 5 do fluxo básico existir uma reserva idêntica no sistema (mesmo nome, e-mail, e mesmas datas de entrada e saída), o sistema exibe a reserva existente e pergunta para o cliente se ele quer proceder com esta reserva. 5A1. Se o cliente quiser continuar, o sistema procede com a reserva. 5A2. Se o cliente indicar que a nova reserva é uma duplicação, o caso de uso termina.
Sub-fluxos	3S. Verificar custo do quarto 3S1. O cliente seleciona o tipo de quarto desejado e indica o período de permanência. 3S2. O sistema computa o custo para o período indicado.
Requisitos Especiais	O sistema deve manipular cinco reservas concorrentes. Cada reserva não deve demorar mais de 20 segundos.
Pré-condições	O cliente está na página do sistema do hotel.
Pós-condições	Uma nova reserva é criada quando a reserva é feita com sucesso, e o número de quartos disponíveis para as datas especificadas foi decrementado. Se a reserva não for realizada com sucesso, não há mudanças no banco de dados.
Pontos de Extensão	E1. O ponto de extensão {Atualizar Disponibilidade do Quarto} ocorre no passo 5 do fluxo básico.
Pontos de Inclusão	Nenhum.

A especificação de requisitos não-funcionais do SGH é descrita na Tabela 15.

Tabela 15 – Especificação de requisitos não-funcionais (SGH).

Id	Requisito	Prioridade
1	O sistema deve ser acessível via internet.	1
2	O sistema deve armazenar todos os registros em um banco de dados relacional.	2
3	Para fins de auditoria, todas as transações no sistema devem ser registradas.	4
4	Somente pessoas autorizadas podem desempenhar as suas respectivas funções.	3
5	O sistema deve rastrear as preferências do cliente e utilizá-las como <i>default</i> .	5
6	Todas as recuperações de dados não devem exceder 2 segundos.	2
7	O sistema deve manipular cinco reservas concorrentes.	2
8	Cada reserva não deve exceder 20 segundos.	2

Os RNFs, bem como suas respectivas prioridades, foram recebidos como dados de entrada. O menor valor indica a maior prioridade. A prioridade mais alta foi estabelecida para o requisito referente à acessibilidade via internet (id 1), pois todas as reservas devem ser executadas pelo cliente, via internet. A segunda prioridade foi estabelecida para os RNFs de desempenho, que são: registros que devem ser armazenados em um banco de dados relacional (id 2), tempo limite para recuperações de dados (id 6), manipulação de reservas concorrentes (id 7) e tempo limite para realizar uma reserva (id 8).

5.1.2 Sub-processo 2: Relacionar interesses funcionais e não-funcionais

O sub-processo 2, para relacionar interesses funcionais e não-funcionais, é composto das atividades para identificar interesses funcionais e não-funcionais (2.1) e integrar interesses funcionais e não-funcionais (2.2).

Na atividade para identificar os interesses funcionais e não-funcionais (2.1), a meta é alcançada a partir da busca de informação no modelo de casos de uso e da especificação de requisitos não-funcionais, recebidos como dados de entrada.

Como resultado desta atividade foram gerados os *softgoals* (Figura 51), a lista de interesses (Tabela 16) e a tabela de RNF (Tabela 17).

Figura 51 – *Softgoals* (SGH).

Os *softgoals* representam os interesses não-funcionais do SGH, que por sua vez podem representar um ou mais requisitos não-funcionais do mesmo tipo. A lista de interesses do SGH é apresentada na Tabela 16.

Tabela 16 – Tabela de interesses (SGH).

Interesse	Natureza
Reservar Quarto	Funcional
Manipular Lista de Espera	
Registrar Membros	
Check in	
Check out	
Ganhar e Resgatar Créditos	
Estabelecer as Taxas dos Quartos	
Estabelecer Taxas Promocionais	
Acessibilidade	Não-Funcional
Desempenho	
Security	
Usabilidade	
Informativo	

Cada um dos interesses possui uma natureza, que pode ser funcional ou não-funcional. Os interesses funcionais são os casos de uso do SGH e os interesses não-funcionais são os *softgoals* referentes ao SGH.

Os requisitos não-funcionais devem ser classificados de acordo com os interesses não-funcionais definidos para o SGH. Esta classificação é

apresentada na Tabela 17, que é a Tabela 15, acrescida de colunas referentes ao interesse e a origem do interesse.

Tabela 17 – Tabela de RNF (SGH).

Id	Requisito	Prioridade	Interesse	Origem do Interesse
1	O sistema deve ser acessível via internet.	1	Acessibilidade	Lista Chung
2	O sistema deve armazenar todos os registros em um banco de dados relacional.	2	Desempenho	NFR
3	Para fins de auditoria, todas as transações no sistema devem ser registradas.	4	Informativo	Lista Chung
4	Somente pessoas autorizadas podem desempenhar as suas respectivas funções.	3	Security	NFR
5	O sistema deve rastrear as preferências do cliente e utilizá-las como <i>default</i> .	5	Usabilidade	ISO/IEC 9126
6	Todas as recuperações de dados não devem exceder 2 segundos.	2	Desempenho	NFR
7	O sistema deve manipular cinco reservas concorrentes.	2	Desempenho	NFR
8	Cada reserva não deve exceder 20 segundos.	2	Desempenho	NFR

Os interesses não-funcionais possuem uma origem, isto é, a fonte onde foi encontrada a classificação mais adequada para cada um deles. Esta classificação pode ser proveniente do catálogo do NFR (*NFR Framework*), da norma ISO/IEC 9126 (ISO/IEC 9126, 2001) ou da lista proposta por Chung et al. (2000).

O interesse de acessibilidade é composto por apenas um requisito não-funcional (id 1), enquanto que o interesse de desempenho é composto por quatro requisitos não-funcionais de desempenho (id 2, id 6, id 7 e id 8).

A atividade para integrar interesses funcionais e não-funcionais (2.2) utilizou as informações geradas na atividade 2.1 para elaborar o diagrama integrado, que relaciona interesses funcionais e não-funcionais. Este diagrama é apresentado na Figura 52.

Figura 52 – Diagrama integrado (SGH).

Este diagrama representa associações de *softgoals* com o limite do sistema, relação ator-caso de uso e caso de uso. Estas associações são estabelecidas de acordo com as definições da seção 4.4.

Os *softgoals* de acessibilidade, desempenho e informativo estão associados ao limite de sistema por representarem RNFs gerais, aplicáveis ao sistema como um todo. O *softgoal* de *security* está associado à relação ator-caso de uso por representar de um RNF relacionado à comunicação entre os atores e o SGH. Os *softgoals* de desempenho e usabilidade estão associados ao caso de uso *Reservar Quarto*, por representarem RNFs relacionados especificamente à reserva de quartos.

O *softgoal* de desempenho está associado ao limite de sistema e ao caso de uso *Reservar Quarto*, pois os RNFs (id 2 e id 6) referem-se ao sistema, e os RNFs (id 7 e id 8) referem-se ao caso de uso *Reservar Quarto*.

5.1.3 Sub-processo 3: Identificar aspectos

O sub-processo 3, para identificar aspectos, é composto das atividades de identificar propagações (3.1) e identificar aspectos (3.2).

Na atividade para identificar propagações (3.1), o diagrama integrado e os SIGs iniciais são recebidos como dados de entrada e as propagações relacionadas a cada um dos RNFs, que compõem um interesse não-funcional, devem ser analisadas.

As heurísticas para analisar as propagações de um RNF nos elementos do diagrama de casos de uso estão definidas na seção 4.5.

O resultado desta análise está apresentado na Tabela 18, que é a Tabela 17, acrescida das colunas ponto de associação e propagação.

Tabela 18 – Tabela de RNF com propagação (SGH).

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
1	O sistema deve ser acessível via internet.	1	Acessibilidade	Lista Chung	Acessibilidade [SGH]	Sistema
2	O sistema deve armazenar todos os registros em um banco de dados relacional.	2	Desempenho	NFR	Desempenho [SGH]	Sistema
3	Para fins de auditoria, todas as transações no sistema devem ser registradas.	4	Informativo	Lista Chung	Informativo [SGH]	Sistema
4	Somente pessoas autorizadas podem desempenhar as suas respectivas funções.	3	<i>Security</i>	NFR	Security [Equipe do Hotel-Registrar Membros, Equipe do Hotel-Check in, Equipe do Hotel-Check out, Cliente-Reservar Quarto, Gerenciamento do Hotel-Estabelecer Taxas dos Quartos]	Nenhuma
5	O sistema deve rastrear as preferências do cliente e utilizá-las como <i>default</i> .	5	Usabilidade	ISO/IEC 9126	Usabilidade [Reservar Quarto]	Nenhuma
6	Todas as recuperações de dados não devem exceder 2 segundos.	2	Desempenho	NFR	Desempenho [SGH]	Sistema
7	O sistema deve manipular cinco reservas concorrentes.	2	Desempenho	NFR	Desempenho [Reservar Quarto]	Caso de uso Manipular Lista de Espera

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
8	Cada reserva não deve exceder 20 segundos.	2	Desempenho	NFR	Desempenho [Reservar Quarto]	Nenhuma

A coluna ponto de associação identifica o SIG inicial relacionado ao requisito não-funcional representado no diagrama integrado da Figura 52.

Para os RNFs pertencentes ao interesse de desempenho, foram identificados pontos de associação com o limite de sistema (id 2 e id 6) e com o caso de uso *Reservar Quarto* (id 7 e id 8). Para os pontos de associação de limite do sistema, a propagação se dá no sistema como um todo. No caso do RNF (id 7), uma propagação ocorreu no caso de uso *Manipular Lista de Espera*, pois para que uma reserva seja concluída, a utilização da lista de espera pode ser necessária. A propagação do RNF de desempenho (id 8) não ocorreu no caso de uso *Manipular Lista de Espera*, pois se considerou que somente o caso de uso *Reservar Quarto* é responsável pela reserva em até 20 segundos.

A atividade para identificar aspectos (3.2) utiliza as heurísticas definidas na seção 4.5 deste trabalho. A Tabela 19 apresenta a tabela de RNF com aspectos, que é a tabela de RNF com propagação acrescida da coluna aspecto.

Tabela 19 – Tabela de RNF com aspectos (SGH).

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação	Aspecto
1	O sistema deve ser acessível via internet.	1	Acessibilidade	Lista Chung	Acessibilidade [SGH]	Sistema	Sim
2	O sistema deve armazenar todos os registros em um banco de dados relacional.	2	Desempenho	NFR	Desempenho [SGH]	Sistema	Sim
3	Para fins de auditoria, todas as transações no sistema devem ser registradas.	4	Informativo	Lista Chung	Informativo [SGH]	Sistema	Sim
4	Somente pessoas autorizadas podem desempenhar suas funções.	3	Security	NFR	Segurança [Equipe do Hotel-Registrar Membros, Equipe do Hotel-Check in, Equipe do Hotel-Check out, Cliente-Reservar Quarto,	Nenhuma	Sim

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação	Aspecto
					Gerenciamento do Hotel- Estabelecer Taxas dos Quartos]		
5	O sistema deve rastrear as preferências do cliente e utilizá-las como <i>default</i> .	5	Usabilidade	ISO/IEC 9126	Usabilidade [Reservar Quarto]	Nenhuma	Não
6	Todas as recuperações de dados não devem exceder 2 segundos.	2	Desempenho	NFR	Desempenho [SGH]	Sistema	Sim
7	O sistema deve manipular cinco reservas concorrentes.	2	Desempenho	NFR	Desempenho [Reservar Quarto]	Caso de uso Manipular Lista de Espera	Sim
8	Cada reserva não deve exceder 20 segundos.	2	Desempenho	NFR	Desempenho [Reservar Quarto]	Nenhuma	Não

Os RNFs que não foram considerados aspectos (id 5 e id 8) não possuem nenhuma propagação e também não possuem associação com mais de um elemento do diagrama de casos de uso. Os requisitos id 5 e id 8 estão associados somente ao caso de uso *Reservar Quarto*.

Os RNFs associados ao limite do sistema, como: id 1, id 2, id 3 e id 6 são considerados aspectos pois podem afetar diversos casos de uso ou mesmo diversos artefatos do sistema de software em fases posteriores do desenvolvimento.

O requisito não-funcional de *security* (id 4) está associado a diversos elementos do modelo de casos de uso, e por isso, é um aspecto, mesmo sem possuir nenhuma propagação.

O requisito não-funcional de desempenho (id 7) foi propagado para o caso de uso *Manipular Lista de Espera* e, desta forma, é considerado um aspecto.

5.1.4 Sub-processo 4: Decompor requisitos não-funcionais

Este sub-processo é composto pelas atividades para selecionar SIG inicial a ser detalhado (4.1), decompor *softgoals* (4.2), definir prioridades (4.3), identificar operacionalizações (4.4), identificar correlações (4.5), registrar decisões de projeto (4.6), selecionar alternativas de projeto (4.7), aplicar o procedimento de avaliação (4.8) e definir relacionamentos de metas de

operacionalizações (4.9). Todas estas atividades têm como principal objetivo elaborar os SIGs com base nos SIGs iniciais.

A atividade para selecionar SIG inicial a ser detalhado (4.1) recebe como entrada o diagrama integrado, os SIGs iniciais e a especificação de requisitos não-funcionais. A seleção do SIG inicial para o detalhamento é feita considerando as prioridades dos RNFs contidas na especificação de requisitos não-funcionais (Tabela 15).

Para ilustrar a atividade para decompor RNFs, foram selecionados os RNFs de acessibilidade e desempenho, que possuem, respectivamente, as prioridades 1 e 2. Os SIGs apresentados foram construídos através das atividades do sub-processo 4 (decompor requisitos não-funcionais).

O SIG que representa o RNF de acessibilidade está apresentado na Figura 53.

Figura 53 – SIG1 de acessibilidade (SGH).

Este SIG resultou em uma única meta que direciona o desenvolvimento do SGH para a plataforma *web*. O RNF de acessibilidade não faz parte do catálogo de decomposição de RNFs, por isso, este SIG foi elaborado sem a utilização de catálogos de decomposição de tipo e de operacionalização.

O SIG que representa os RNFs de desempenho (id 2 e id 6) associados ao limite do sistema é apresentado na Figura 54. A escolha pela decomposição de dois RNFs no mesmo SIG foi tomada pelo engenheiro de requisitos pois, além de ambos os RNFs estarem associados ao limite do sistema, estão também relacionados à manipulação de dados do sistema de software.

Figura 54 – SIG 2 de desempenho relacionado ao limite do sistema (SGH).

Este SIG é inicialmente decomposto utilizando o método de decomposição de subtipo, que resulta no *softgoal* de espaço. Posteriormente é utilizado o método de decomposição de gerenciamento de entidade. Este método fica localizado na camada de transações (número 3) da estrutura que organiza os métodos de decomposição dos RNFs de desempenho para sistemas de informação. A estrutura de camadas do catálogo de desempenho do RNF *Framework* tem a finalidade de organizar a informação sobre interações entre características dos modelos de dados, técnicas de implementação e características de desempenho relacionadas às linguagens de especificação. São seis camadas que organizam a informação sobre o RNF de desempenho: entidades e classes, atributos, transações, hierarquias de dados, restrições de integridade e processo (CHUNG et al., 2000). O método de gerenciamento de entidade tem a finalidade de manipular dados a partir de operações como: armazenar, recuperar, atualizar e remover. Neste caso, o SIG 2 trata do armazenamento de dados para o SGH.

As metas definidas para o SIG 2 foram o uso de banco de dados relacional estruturado e vários atributos por tupla. O *softgoal* de operacionalização relacionado ao banco de dados relacional não faz parte do catálogo do NFR

Framework e, por isso, foi definido pelo engenheiro de requisitos. O *softgoal* relacionado ao armazenamento de vários atributos por tupla pertence ao catálogo de operacionalizações do RNF de desempenho do NFR *Framework*.

Dois SIGs relacionados ao desempenho do SGH, associados ao caso de uso *Reservar Quarto* foram criados. A escolha pela criação de SIGs separados foi feita pelo engenheiro de requisitos, para que o SIG fique mais claro, pois apesar dos requisitos estarem representados pelo mesmo ponto de associação no diagrama integrado, eles podem tratar de assuntos diferentes.

O SIG que representa o requisito não-funcional de desempenho (id 7) associado ao caso de uso *Reservar Quarto* é apresentado na Figura 55.

Figura 55 – SIG 3 de desempenho relacionado ao caso de uso Reservar Quarto (SGH).

Este SIG faz uso do método de decomposição de caso de uso definido na seção 4.6.1. A aplicação deste método tem a finalidade de apresentar a propagação do RNF (id 7) no caso de uso *Manipular Lista de Espera*. Dois *softgoals* de operacionalização foram definidos, porém, somente o *softgoal* de operacionalização para introduzir concorrência foi aceito para o SGH. Ambas as operacionalizações fazem parte das táticas apresentadas no item 4.6.2 deste trabalho. O *softgoal* introduzir concorrência sugere a implementação através do uso de *threads* e, com isso, diversas reservas concorrentes poderão ser manipuladas pelo SGH.

O SIG que representa o requisito não-funcional de desempenho (id 8) associado ao caso de uso *Reservar Quarto* é apresentado na Figura 56.

Figura 56 – SIG 4 de desempenho relacionado ao caso de uso Reservar Quarto (SGH).

Neste SIG, o método de decomposição de subtipo é utilizado para decompor o *softgoal* de desempenho em um *softgoal* de tempo. O *softgoal* de tempo é decomposto utilizando o método de transações individuais, com a finalidade de identificar a operação de reserva a ser executada pelo caso de uso *Reservar Quarto*. Um *softgoal* de operacionalização foi definido a partir das táticas de Bass, Clements e Kazman (2003), para aumentar a disponibilidade de recursos a fim de executar reservas em até 20 segundos.

Os demais SIGs do SGH encontram-se em Bombonatti (2010).

5.1.5 Sub-processo 5: Elaborar modelo composto

Este sub-processo é composto pelas atividades para identificar metas operacionais e restrições de projeto (5.1), definir pontos de composição (5.2), confirmar propagação (5.3), resolver conflitos (5.4) e atualizar o modelo do sistema de software (5.5).

A atividade para identificar metas operacionais e restrições de projeto (5.1) recebe como dados e entrada os SIGs e a tabela de RNF com aspectos. Nesta atividade deve ser elaborada a tabela de RNF com metas operacionais e

restrições de projeto, que é a tabela de aspectos (Tabela 19) acrescida das colunas: id SIG, meta, origem da meta, meta operacional ou restrição de projeto e descrição. Esta tabela tem a finalidade de registrar a classificação do tipo de meta referente a cada SIG e a ação a ser tomada caso a meta seja classificada como meta operacional ou restrição de projeto.

A tabela de RNF com metas operacionais e restrições de projeto é apresentada na Tabela 20.

Tabela 20 – Tabela de RNF com metas operacionais e restrições de projeto (SGH).

Id	Id SIG	Meta	Origem da Meta	Meta Operacional ou Restrição de Projeto	Descrição
1	SIG1	O sistema deve ser desenvolvido em plataforma web.	Definido pelo engenheiro de requisitos	Restrição de Projeto	Definir arquitetura do projeto baseada em plataforma web.
2	SIG2	Utilizar um banco de dados relacional estruturado.	Definido pelo engenheiro de requisitos	Restrição de Projeto	Definição da camada de persistência e modelo de dados baseado em um banco de dados relacional.
3	SIG6	Armazenar log das operações para fins de auditoria.	Definido pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Armazenar Log.
4	SIG5	A cada novo acesso de usuário solicitar identificação, autenticação e validação de acesso.	NFR	Meta Operacional	Novo caso de uso Autenticar Usuário.
5	SIG7	Recuperar as informações de log a fim de definir as preferências de usuário.	Definido pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Definir Preferências.
6	SIG2	Utilizar vários atributos por tupla a fim de obter recuperações de dados em até 2 segundos.	NFR	Restrição de Projeto	Definição da camada de persistência e modelo de dados.
7	SIG3	Utilizar controle do processo de software por <i>threads</i> para manipular reservas concorrentes.	Tática	Restrição de Projeto	Codificação de software com <i>threads</i> .
8	SIG4	Obter os melhores recursos computacionais disponíveis a fim de processar as reservas em até 20 segundos.	Tática	Restrição de Projeto	Definição da arquitetura computacional com os melhores recursos computacionais disponíveis.

Na representação da tabela de RNF com metas operacionais e restrições (Tabela 20) foram apresentadas as colunas referentes ao conteúdo da tabela de RNF com metas operacionais e restrições de projeto e o identificador dos requisitos, para auxiliar o entendimento desta última atividade.

Nesta tabela podem ser observadas as restrições de projeto a serem impostas ao SGH e os novos casos de uso a serem adicionados no modelo de casos de uso. Estes casos de uso são identificados somente para metas operacionais, que requerem a inclusão de novas funções no modelo do sistema de software.

Para o RNF (id 1) foi considerada uma restrição de projeto, pois a arquitetura do SGH deve ser definida com base na plataforma web. Para o RNF (id 3) foi considerada meta operacional, pois uma nova função deve ser adicionada no SGH, através do caso de uso *Armazenar Log*, para que o registro das operações seja realizada pelo sistema, para fins de auditoria.

A atividade para definir pontos de composição (5.2) tem a finalidade de identificar os pontos no fluxo de seqüência de eventos dos casos de uso, onde as metas operacionais devem ser inseridas. A Tabela 21 apresenta a tabela de RNF com composição, que é a tabela de RNF com metas operacionais e restrições de projeto (Tabela 20), acrescida das colunas: caso de uso, ponto de composição, regra de composição e condições da composição.

Tabela 21 – Tabela de RNF com composição (SGH).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição
3	Reservar Quarto	6	sobreposição.depois	Nenhuma
	Registrar Membros	3, 1A1, 1B1, 1C1	sobreposição.depois	Nenhuma
	Check in	2	sobreposição.depois	Nenhuma
	Check out	1	sobreposição.depois	Nenhuma
	Estabelecer Taxas dos Quartos	2	sobreposição.depois	Nenhuma
	Manipular Lista de Espera	1	sobreposição.depois	Nenhuma
	Ganhar e Resgatar Créditos	2	sobreposição.depois	Nenhuma
	Estabelecer Taxas Promocionais	3	sobreposição.depois	Nenhuma
4	Reservar Quarto	1	sobreposição.antes	Nenhuma
	Registrar Membros	1	sobreposição.antes	Nenhuma
	Check in	1	sobreposição.antes	Nenhuma
	Check out	1	sobreposição.antes	Nenhuma
	Estabelecer Taxas dos Quartos	1	sobreposição.antes	Nenhuma
5	Armazenar Log	1	sobreposição.depois	Nenhuma

As linhas apresentadas na tabela de RNF com composição tratam-se apenas dos RNFs que possuem metas operacionais como resultado da decomposição através dos SIGs. As restrições de projeto não necessitam de definições relacionadas à composição, pois não adicionam novas funções ao sistema.

Nesta tabela são identificados os casos de uso que serão afetados pela meta operacional, o ponto na seqüência de eventos do caso de uso afetado, o tipo de regra de composição e, caso necessário, a condição da composição.

As regras de composições definidas para o SGH são do tipo sobreposição, pois não se fez necessário substituir ou mesmo empacotar nenhum fluxo de eventos dos casos de uso do SGH, por isso, não é necessário o preenchimento da coluna condição da composição.

É possível verificar na Tabela 21 que os requisitos não-funcionais id 3 e id 4 possuem impacto em diversos casos de uso e o requisito não-funcional id 5, impacta apenas o caso de uso *Armazenar Log*.

Um exemplo, da atividade de definir pontos de composição, relacionado ao requisito não-funcional (id 3) é a composição do caso e uso *Armazenar Log*, definido para o requisito (id 3) na tabela de RNF com meta operacional e restrição de projetos, ao caso de uso *Reservar Quarto*, depois da execução do passo 6 do seu fluxo básico.

A atividade para confirmar as propagações (5.3) tem a finalidade de analisar e consolidar as propagações inicialmente identificadas, pois a definição dos pontos de composição pode dar uma visão mais clara sobre onde os novos casos de uso devem ser adicionados.

Inicialmente, não existia propagação prevista no requisito não-funcional id 8 (Tabela 20); no entanto, detectou-se que este requisito propaga-se para o sistema como um todo, pois a utilização de melhores recursos computacionais afeta não somente o caso de uso *Reservar Quarto*, mas o SGH. Com esta constatação, este requisito é considerado um aspecto. A Figura 57 apresenta o novo SIG após esta revisão.

Figura 57 – SIG 4 atualizado (SGH).

A atividade para a resolver conflitos (5.4) tem a finalidade de verificar se os casos de uso afetados possuem a mesma regra de composição a ser aplicada no mesmo ponto de composição. No SGH não foram detectados conflitos entre casos de uso. Apesar da inexistência de conflitos, a tabela de RNF com conflitos deve ser preenchida com a informação da inexistência de conflitos, conforme regras de preenchimento definidas no Apêndice B. A tabela de RNF com prioridades é a tabela de RNF com composição (Tabela 21) acrescida da coluna prioridades da composição.

A atividade para atualizar o modelo do sistema de software (5.5) tem a finalidade de construir um modelo de casos de uso atualizado com os novos casos de uso incluídos. A Figura 58 apresenta o diagrama de casos de uso atualizado do SGH.

Figura 58 – Diagrama de casos de uso atualizado (SGH).

Os casos de uso destacados são os novos casos de uso incluídos no diagrama. Os casos de uso *Armazenar Log* e *Autenticar Usuário* são aspectos do SGH, pois são incluídos por mais de um caso de uso. Estes casos de uso possuem o estereótipo <<aspecto>>. O caso de uso *Definir Preferência* que não é aspecto SGH, não possui o estereótipo <<aspecto>>. A informação sobre os casos de uso que são ou não considerados aspectos deve ser consultada na coluna aspecto (Tabela 19).

Como pode ser observado na Figura 58, ocorreu o inter-relacionamento entre os novos casos de uso *Definir Preferências* e *Armazenar Log*. A existência dessa relação se deve ao fato de que o caso de uso *Definir Preferências* utiliza as informações de log para que as preferências do usuário possam ser atualizadas em seu próximo acesso.

As especificações dos novos casos de uso incluídos também devem ser elaboradas. Um exemplo de especificação do caso de uso *Autenticar Usuário* pode ser observado na Tabela 22.

Tabela 22 – Exemplo da especificação do caso de uso Autenticar Usuário (SGH).

Nome do Caso de Uso	Autenticar Usuário
Descrição Breve	Identificar, autenticar e validar acesso de usuários ao sistema.
Atores	Nenhum

Evento Inicial	Os casos de uso Reservar Quarto, Registrar Membros, Check in, Check out e Estabelecer Taxas dos Quartos solicitam autenticação do usuário no sistema antes da execução de qualquer operação.
Fluxo Básico	<ol style="list-style-type: none"> 1. O usuário digita seu nome e sua senha. 2. O sistema identifica, autentica e valida seu acesso. 3. O sistema verifica suas permissões. 4. O caso de uso termina.
Fluxos Alternativos	2A. Usuário não identificado. 2A1. O sistema emite uma mensagem para usuário não identificado. 2A2. O caso de uso termina.
Sub-Fluxos	Nenhum.
Requisitos Especiais	4
Pré-condições	O usuário está com a janela de pedido de login e senha.
Pós-condições	Usuário logado no sistema.
Pontos de Extensão	Nenhum.
Pontos de Inclusão	Nenhum.

O campo requisitos especiais é preenchido com o identificador do RNF que deu origem à nova especificação. Demais especificações dos casos de uso incluídos no modelo atualizado de casos de uso do SGH encontram-se em Bombonatti (2010).

5.2 Sistema de Bóias Meteorológicas

O sistema de bóias meteorológicas (SBM) utilizado neste trabalho é baseado no exemplo de Booch (1986). Este sistema foi escolhido pois se trata de um sistema de software embarcado de tempo real.

Este sistema de software coleta dados para navegação e tempo do ambiente e transmite para ser usado no tráfego marítimo. As bóias estão distribuídas nos pontos estratégicos do mar e coletam temperatura da água e do ar, velocidade do vento e dados da sua posição, através de sensores de diversos tipos. Cada bóia tem um número diferente de sensores de vento e de temperatura, e pode ser modificada para aceitar outros tipos de sensores no futuro. Cada bóia é também equipada com um rádio transmissor, para envio de informações, e um receptor, para receber as solicitações dos navios. A bóia possui uma chave que, quando acionada, inicia a transmissão da mensagem de pedido de socorro pelo naufrago. A transmissão desta mensagem pode ser

desligada a partir de uma mensagem enviada pelo navio. A bóia possui também uma luz vermelha, que é ativada pelos navios, durante a operação de busca de náufragos, e desligada no final do resgate.

5.2.1 Sub-processo 1: Obter os requisitos do sistema de software

Os documentos de entrada consistiram da descrição do sistema de Booch (2004) e o modelo de casos de uso de Melnikoff (2009). Após a atividade de verificar a documentação de entrada (1.1), foi executada a atividade para elaborar os requisitos, que consistiu do levantamento do glossário, da elaboração da especificação de RNFs e da revisão do modelo de casos de uso.

O glossário do SBM é apresentado na Tabela 23.

Tabela 23 – Glossário (SBM).

Termo	Definição
Bóias Meteorológicas	É um equipamento que coleta e envia dados do ambiente, como temperatura da água e do ar, velocidade do vento e dados da sua posição, para auxiliar no tráfego marítimo.
Mensagem periódica	Mensagem enviada periodicamente pela bóia, em um intervalo pré-definido de tempo.
Mensagem não-periódica	Mensagem enviada pela bóia através da solicitação de navios.

O diagrama de casos de uso do SBM é apresentado na Figura 59.

Figura 59 – Diagrama de casos de uso (SBM).

Os atores do SBM são os seguintes:

- Sensor: é responsável pela coleta das informações do ambiente (temperatura da água e do ar, velocidade do vento e dados da posição da bóia).
- Relógio: é responsável pelo acionamento da varredura dos sensores.
- Naufrago: é responsável por acionar o pedido de socorro nas bóias.
- Navio: é responsável pela ativação/desativação da luz vermelha, recebimento/cancelamento das mensagens de ajuda, recebimento das mensagens periódicas da bóia e solicitação e recebimento dos dados das últimas 24 horas.

O caso de uso *Ler Sensores* tem a finalidade de executar as leituras dos dados provenientes dos sensores.

O caso de uso *Calcular Médias* executa o cálculo da média dos valores de temperatura e velocidade do vento, após dez coletas consecutivas.

O caso de uso *Transmitir Mensagem* envia mensagens da bóia aos navios.

O caso de uso *Enviar Mensagem SOS* transmite mensagens de pedido de ajuda do naufrágio para o navio.

O caso de uso *Desligar Mensagem SOS* comanda o encerramento da transmissão de mensagens de pedido de ajuda.

O caso de uso *Comandar Luz Vermelha* ativa a luz vermelha da bóia.

O caso de uso *Desativar Luz Vermelha* desliga a luz vermelha da bóia.

A Tabela 24 apresenta uma especificação do caso de uso *Ler Sensores*. A descrição dos seus campos encontra-se no Apêndice B. As demais especificações do SBM encontram-se em Bombonatti (2010).

Tabela 24 – Especificação do caso de uso Ler Sensores (SBM).

Nome do Caso de Uso	Ler Sensores
Descrição Breve	Este caso de uso tem o objetivo de coletar e armazenar informações provenientes dos sensores da bóia.
Atores	Relógio Sensor
Evento Inicial	Sinal do início da varredura.
Fluxo Básico	<ol style="list-style-type: none"> 1. O sistema lê a informação dos sensores da bóia e transforma em uma unidade de engenharia conveniente. <i>{Média}</i> 2. O sistema armazena os dados obtidos juntamente com a data e hora da leitura. 3. O caso de uso termina

Fluxos Alternativos	Nenhum.
Sub-fluxos	Nenhum.
Requisitos Especiais	Nenhum.
Pré-condição	Coleta dos dados pronta para ser iniciada.
Pós-condição	Dados dos sensores da bóia armazenados.
Pontos de Extensão	E1. O ponto de extensão {Média} ocorre no passo 1 do fluxo básico. A informação é lida e a média é calculada depois da coleta de 10 valores consecutivos.
Pontos de Inclusão	Nenhum.

A especificação de requisitos não-funcionais do SBM é descrita na Tabela 25.

Tabela 25 – Especificação de requisitos não-funcionais (SBM).

Id	Requisito	Prioridade
1	O sistema deve coletar os dados da velocidade do vento a cada 30 segundos.	1
2	O sistema deve coletar os dados da temperatura da água e do ar, e da localização a cada 10 segundos.	1
3	O sistema deve transmitir aos navios os dados de velocidade do vento, temperatura e localização a cada 60 segundos.	1
4	O sistema deve transmitir os dados das últimas 24 horas para um navio quando solicitado. Esta operação é menos prioritária que as transmissões periódicas.	1
5	O sistema pode ser modificado para incluir outros sensores no futuro.	2
6	O sistema deve verificar a acurácia das informações provenientes dos sensores.	3
7	O navio deve ter permissão para solicitar dados por razões de segurança.	4

Na Tabela 25, o menor valor indica a maior prioridade. A prioridade mais alta foi estabelecida para o requisito referente à leitura e transmissão de dados (id 1, id 2, id 3 e id 4). Demais prioridades foram estabelecidas para os requisitos de modificação do sistema para inclusão de novos sensores (id 5), verificação da acurácia da informação (id 3) e acesso permitido para solicitação de dados (id 4).

5.2.2 Sub-processo 2: Relacionar interesses funcionais e não-funcionais

O sub-processo 2, para relacionar interesses funcionais e não-funcionais, é composto das atividades para identificar interesses funcionais e não-funcionais (2.1) e integrar interesses funcionais e não-funcionais (2.2).

Na atividade para identificar os interesses funcionais e não-funcionais (2.1), foram utilizados o modelo de casos de uso e a especificação de requisitos não-funcionais, recebidos como dados de entrada para esta atividade.

Os *softgoals* resultantes são apresentados na Figura 60.

Figura 60 – *Softgoals* (SBM).

Os *softgoals* representam os interesses não-funcionais do SBM, que por sua vez podem representar um ou mais requisitos não-funcionais do mesmo tipo. A lista de interesses do SBM é apresentada na Tabela 26.

Tabela 26 – Tabela de interesses (SBM).

Interesse	Natureza
Ler Sensores	Funcional
Calcular Médias	
Comandar Luz Vermelha	
Desativar Luz Vermelha	
Transmitir Mensagem	
Enviar Mensagem SOS	
Desligar Mensagem SOS	
Desempenho	Não-Funcional
Modificabilidade	
Acurácia	
Security	

Os RNFs devem ser classificados de acordo com os interesses não-funcionais definidos para o SBM. Esta classificação é apresentada na Tabela 27 que é a Tabela 25, acrescida de colunas referentes ao interesse e a origem do interesse.

Tabela 27 – Tabela de RNF (SBM).

Id	Requisito	Prioridade	Interesse	Origem do Tipo de Interesse
1	O sistema deve coletar os dados da velocidade do vento a cada 30 segundos.	1	Desempenho	NFR
2	O sistema deve coletar os dados da temperatura da água e do ar, e da localização a cada 10 segundos.	1	Desempenho	NFR
3	O sistema deve transmitir aos navios os dados de velocidade do vento, temperatura e localização a cada 60 segundos.	1	Desempenho	NFR

Id	Requisito	Prioridade	Interesse	Origem do Tipo de Interesse
4	O sistema deve transmitir os dados das últimas 24 horas para um navio quando solicitado. Esta operação é menos prioritária que as transmissões periódicas.	1	Desempenho	NFR
5	O sistema pode ser modificado para incluir outros sensores no futuro.	2	Modificabilidade	ISO/IEC 9126
6	O sistema deve verificar a acurácia das informações provenientes dos sensores.	3	Acurácia	NFR
7	O navio deve ter permissão para solicitar dados por razões de segurança.	4	Security	NFR

O interesse de desempenho é composto pelos requisitos RNFs (id 1, id 2, id 3 e id 4). O interesse de modificabilidade é composto de apenas um requisito não-funcional (id 5); o mesmo ocorre com os interesses de acurácia (id 6) e *security* (id 7). A origem do interesse de desempenho, acurácia e *security* é o NFR *Framework*, enquanto que a origem do interesse de modificabilidade é a norma ISO/IEC 9126 (ISO/IEC 9126, 2001).

A atividade para integrar interesses funcionais e não-funcionais (2.2) utilizou as informações geradas na atividade 2.1 para que o diagrama integrado fosse elaborado. Este diagrama é apresentado na Figura 61.

Figura 61 – Diagrama integrado (SBM).

Neste diagrama pode-se notar associações entre *softgoals* e o limite do sistema, relação ator-caso de uso e caso de uso. Estas associações são estabelecidas de acordo com as definições da seção 4.4.

O *softgoal* de modificabilidade está associado ao limite de sistema por representar um RNF geral, aplicável ao sistema como um todo. Os *softgoals* de *security* e acurácia estão associados à relação ator-caso de uso: ambos representam RNFs relacionados à comunicação entre os atores e o SBM. O *softgoal* de desempenho está associado aos casos de uso *Ler Sensores* e *Transmitir Mensagens*, por terem sido considerados RNFs relacionados especificamente à leitura e transmissão de dados.

O *softgoal* de desempenho está associado a dois casos de uso, *Ler Sensores* e *Transmitir Mensagens*, pois os RNFs (id 1 e id 2) se referem ao intervalo de tempo para leitura de dados e os RNFs (id 3 e id 4) referem-se ao intervalo de tempo para transmissão de dados.

5.2.3 Sub-processo 3: Identificar aspectos

O sub-processo 3, para identificar aspectos, é composto das atividades para identificar propagações (3.1) e identificar aspectos (3.2).

Na atividade para identificar propagações (3.1), o diagrama integrado e os SIGs iniciais são recebidos como dados de entrada e as propagações relacionadas a cada um dos RNFs, que compõe um interesse não-funcional, devem ser analisadas.

As heurísticas para analisar as propagações de um requisito não-funcional nos elementos do diagrama de casos de uso estão definidas na seção 4.5 deste trabalho.

O resultado desta análise pode ser observado na Tabela 28 que é a Tabela 27, acrescida de colunas ponto de associação e propagação.

Tabela 28 – Tabela de RNF com propagação (SBM).

Id	Requisito	Prioridade	Interesse	Origem do Tipo de Interesse	Ponto de Associação	Propagação
1	O sistema deve coletar os dados da velocidade do vento a cada 30 segundos.	1	Desempenho	NFR	Desempenho [Ler Sensores]	Nenhuma
2	O sistema deve coletar os dados da temperatura da água e do ar, e da localização a cada 10 segundos.	1	Desempenho	NFR	Desempenho [Ler Sensores]	Nenhuma
3	O sistema deve transmitir aos navios os dados de velocidade do vento, temperatura e localização a cada 60 segundos.	1	Desempenho	NFR	Desempenho [Transmitir Mensagem]	Nenhuma
4	O sistema deve transmitir os dados das últimas 24 horas para um navio quando solicitado. Esta operação é menos prioritária que as transmissões periódicas.	1	Desempenho	NFR	Desempenho [Transmitir Mensagem]	Nenhuma
5	O sistema pode ser modificado para incluir outros sensores no futuro.	2	Modificabilidade	ISO/IEC 9126	Modificabilidade de [SBM]	Sistema
6	O sistema deve verificar a acurácia das informações provenientes dos sensores.	3	Acurácia	NFR	Acurácia [Sensor-Ler Sensores]	Nenhuma
7	O navio deve ter permissão para solicitar dados por razões de segurança.	4	Security	NFR	Security [Navio-Transmitir Mensagem]	Nenhuma

Os requisitos não-funcionais de desempenho (id 1, id 2, id 3 e id 4) associados aos casos de uso *Ler Sensores* e *Transmitir Mensagens* não possuem nenhuma propagação, por serem específicos às funções de leitura e transmissão de dados, respectivamente.

O requisito não-funcional de modificabilidade (id 5) é propagado para o sistema como um todo, pois o sistema deve permitir modificações futuras para inclusão de novos sensores.

Os requisitos não-funcionais de acurácia (id 6) e de *security* (id 7) também não são propagados para nenhum outro elemento do diagrama de casos de uso. O requisito não-funcional de acurácia trata especificamente das informações trocadas entre o sensor e sistema através do caso de uso *Ler*

Sensores e, o requisito não-funcional de *security* é específico à permissão de acesso do navio para que informações possam ser transmitidas a ele.

A atividade para identificar aspectos utiliza as heurísticas definidas na seção 4.5 deste trabalho. A Tabela 29 apresenta tabela de RNF com aspectos, que é a tabela de RNF com propagações (Tabela 28) acrescida da coluna aspecto.

Tabela 29 – Tabela de RNF com aspectos (SBM).

Id	Requisito	Prioridade	Interesse	Origem do Tipo de Interesse	Ponto de Associação	Propagação	Aspecto
1	O sistema deve coletar os dados da velocidade do vento a cada 30 segundos.	1	Desempenho	NFR	Desempenho [Ler Sensores]	Nenhuma	Não
2	O sistema deve coletar os dados da temperatura da água e do ar, e da localização a cada 10 segundos.	1	Desempenho	NFR	Desempenho [Ler Sensores]	Nenhuma	Não
3	O sistema deve transmitir aos navios os dados de velocidade do vento, temperatura e localização a cada 60 segundos.	1	Desempenho	NFR	Desempenho [Transmitir Mensagem]	Nenhuma	Não
4	O sistema deve transmitir os dados das últimas 24 horas para um navio quando solicitado. Esta operação é menos prioritária que as transmissões periódicas.	1	Desempenho	NFR	Desempenho [Transmitir Mensagem]	Nenhuma	Não
5	O sistema pode ser modificado para incluir outros sensores no futuro.	2	Modificabilidade	ISO/IEC 9126	Modificabilidade de [SBM]	Sistema	Sim
6	O sistema deve verificar a acurácia das informações provenientes dos sensores.	3	Acurácia	NFR	Acurácia [Sensor-Ler Sensores]	Nenhuma	Não
7	O navio deve ter permissão para solicitar dados por razões de segurança.	4	<i>Security</i>	NFR	Security [Navio-Transmitir Mensagem]	Nenhuma	Não

Somente o RNF de modificabilidade foi considerado aspecto, pois possui propagação através do limite do sistema, que pode afetar diversos casos de

uso ou mesmo diversos artefatos do sistema de software em fases posteriores do desenvolvimento.

Os demais RNFs não são considerados aspectos, pois não possuem nenhuma propagação e são associados a um elemento do diagrama de casos de uso.

5.2.4 Sub-processo 4: Decompor requisitos não-funcionais

Este sub-processo é composto pelas atividades para selecionar SIG inicial a ser detalhado (4.1), decompor *softgoals* (4.2), definir prioridades (4.3), identificar operacionalizações (4.4), identificar correlações (4.5), registrar decisões de projeto (4.6), selecionar alternativas de projeto (4.7), aplicar o procedimento de avaliação (4.8) e definir relacionamentos de metas de operacionalizações (4.9). Todas estas atividades têm como principal objetivo elaborar os SIGs com base nos SIGs iniciais definidos no diagrama integrado.

A atividade para selecionar SIG inicial a ser detalhado (4.1) recebe como entrada o diagrama integrado, os SIGs iniciais e a especificação de requisitos não-funcionais. A seleção do SIG inicial para o detalhamento é feita considerando as prioridades dos RNFs contidas na especificação de requisitos não-funcionais (Tabela 25).

Para ilustrar a atividade para decompor RNFs, foram selecionados os RNFs de desempenho, modificabilidade e acurácia, que possuem, respectivamente, as prioridades 1, 2 e 3.

O SIG que representa o RNF de desempenho associado ao caso de uso *Ler Sensores* é apresentado na Figura 62.

Figura 62 – SIG 1 de desempenho associado ao caso de uso Ler Sensores (SBM).

Os RNFs de desempenho (id 1 e id 2) associados ao caso de uso *Ler Sensores* são representados no mesmo SIG, pois ambos estão relacionados à leitura de dados dos sensores.

Inicialmente o *softgoal* de desempenho foi decomposto no *softgoal* de tempo através do método de decomposição de subtipo. Neste SIG foi utilizada a camada número 2, para tratamento de atributos, do NFR *Framework*. A definição da camada a ser utilizada para a decomposição do requisito não-funcional de desempenho no NFR *Framework* define o conjunto de métodos de decomposição a ser utilizado. Neste caso foi selecionada a camada de tratamento de atributos, pois os atributos de velocidade do vento, temperatura e localização devem ser manipulados pelo sensor.

O *softgoal* de operacionalização utilizado para organizar a coleta de informações dos sensores foi a indexação, proveniente do catálogo de operacionalização do RNF de desempenho do NFR *Framework*. Esta operacionalização define que os dados devem ser lidos e armazenados com um identificador para que seja possível identificar o momento da leitura.

A utilização da operacionalização de indexação tem um impacto negativo no requisito não-funcional de espaço, de acordo com o catálogo de correlações do

NFR *Framework*, pois a informação referente ao índice também deve ser armazenada. Neste caso, o engenheiro de requisitos considerou que apesar da existência da correlação negativa entre os RNFs de tempo e espaço, o espaço de armazenamento dos índices não terá impacto negativo para o SBM.

Desta forma, o RNF de desempenho associado ao caso de uso *Ler Sensores* é considerado suficientemente satisfeito.

O SIG referente ao RNF de desempenho associado ao caso de uso *Transmitir Mensagem* é apresentado na Figura 63.

Figura 63 –SIG 2 de desempenho associado ao caso de uso Transmitir Mensagem (SBM).

Este SIG utiliza a camada 4 da estrutura em camadas para tratamento dos RNFs de desempenho de sistemas de informação. Esta camada tem a finalidade de tratar a hierarquia de dados. No exemplo da Figura 63, um conjunto de dados deve ser manipulado durante as transmissões periódicas e não-periódicas.

O método de decomposição de subtipo decompõe o *softgoal* de desempenho no *softgoal* de tempo. A fim de explorar os diferentes tipos de operações para a transmissão de mensagens, o *softgoal* de tempo foi decomposto utilizando o método de operações comuns, para expressar os tipos de transmissões: periódicas e não-periódicas.

A transmissão de mensagens periódicas é prioritária em relação às mensagens não-periódicas, como é apresentado no SIG da Figura 63. Os *softgoals* de afirmação mostram que as mensagens não-periódicas devem ser enviadas através de uma solicitação dos navios, e as mensagens periódicas, a cada 60 segundos. Outro *softgoal* de afirmação considerado é o que estabelece que as mensagens de pedido de ajuda devem ser prioritárias em relação às mensagens periódicas.

Os *softgoals* de operacionalização identificados para este SIG dizem respeito à ordem de transmissão das mensagens. A operacionalização a ser executada primeiro refere-se às mensagens periódicas, que devem ser enviadas antes das mensagens não-periódicas, que possuem o *softgoal* de operacionalização executar mais tarde.

A execução mais tarde contribui negativamente para o *softgoal* de tempo, de acordo com o catálogo de correlações do NFR *Framework*; entretanto, esta questão não foi considerada crítica para que o *softgoal* de desempenho pudesse ser suficientemente satisfeito, apesar do rótulo fracamente satisfeito apresentado pelo *softgoal* de tempo.

O SIG referente a modificabilidade é apresentado na Figura 64.

Figura 64 – SIG 3 de modificabilidade (SBM).

O SIG de modificabilidade não pertence aos catálogos do NFR *Framework*, por isso, este SIG foi elaborado pelo engenheiro de requisitos.

Os *softgoals* de operacionalização deste SIG utilizam as táticas coerência semântica e antecipar mudanças previstas (BASS; CLEMENTS; KAZMAN, 2003). A coerência semântica define que módulos devem trabalhar junto, sem que exista dependência excessiva entre eles, isto é, devem apresentar alta coesão e baixo acoplamento. Antecipar mudanças previstas define que, se existe um conjunto de mudanças já esperadas, a atribuição de responsabilidade para os diversos módulos deve considerar estas expectativas, a fim de que diretrizes de reuso possam ser aplicadas. A meta deste SIG define que a arquitetura de software deve considerar estes princípios.

O SIG de acurácia é apresentado na Figura 65.

Figura 65 – SIG 4 de acurácia (SBM).

Este SIG utiliza o catálogo de decomposição do RNF de acurácia do NFR *Framework*. O método utilizado para refinar o *softgoal* de acurácia foi a conservação. Este método define que, para estabelecer a acurácia de um conjunto de informações presentes no sistema, deve-se:

- Estabelecer a acurácia no momento em que a informação é recebida pelo sistema a partir de um agente externo e;
- Manipular corretamente a informação internamente ao sistema.

Em outras palavras, a informação recebida pelo sistema possui acurácia, se a informação for recebida com acurácia, e o processamento interno do sistema conserva a acurácia da informação. No caso do SBM, o sensor foi considerado um agente externo.

Desta forma, o SBM deve garantir o recebimento da informação com acurácia da fonte, através da melhoria do fluxo de informação e, garantir que o fluxo de chegada da informação esteja correto, através da confirmação da acurácia na leitura dos sensores. Neste SIG, duas metas foram identificadas para o mesmo RNF; por isso, a recomendação é que o requisito original seja desmembrado em dois, a fim de proporcionar maior clareza à especificação. O requisito (id 6A) foi adicionado à especificação com o seguinte texto: “O sistema deve receber a informação da leitura dos sensores sem ruído”.

O SIG referente ao requisito não-funcional de *security* do SGH encontra-se em Bombonatti (2010).

5.2.5 Sub-processo 5: Elaborar modelo composto

Este sub-processo é composto pelas atividades para identificar metas operacionais e restrições de projeto (5.1), definir pontos de composição (5.2), confirmar propagação (5.3), resolver conflitos (5.4) e atualizar o modelo do sistema de software (5.5).

A atividade para identificar metas operacionais e restrições de projeto (5.1) recebe como dados e entrada os SIGs e a tabela de RNF com aspectos. Nesta atividade deve ser elaborada a tabela de RNF com metas operacionais e restrições de projeto, que é a tabela de aspectos (Tabela 29) acrescida das colunas: id SIG, meta, origem da meta, meta operacional ou restrição de projeto e descrição. Esta tabela tem a finalidade de registrar a classificação do tipo de meta referente a cada SIG e a ação a ser tomada caso a meta seja classificada como meta operacional ou restrição de projeto.

A tabela de RNF com metas operacionais e restrições de projeto é apresentada na Tabela 30.

Tabela 30 – Tabela de RNF com metas operacionais e restrições de projeto (SBM).

Id	Id SIG	Meta	Origem da Meta	Meta Operacional ou Restrição de Projeto	Descrição
1	SIG1	A velocidade do vento deve ser coletada e indexada no momento do armazenamento a cada 30 segundos.	NFR	Meta Operacional	Novo caso de Uso Coletar Mensagem 30 Segundos.
2	SIG1	A temperatura e a localização devem ser coletadas e indexadas no momento do armazenamento a cada 10 segundos.	NFR	Meta Operacional	Novo caso de Uso Coletar Mensagem 10 Segundos.
3	SIG2	Mensagens periódicas devem ser enviadas a cada 60 segundos. Estas mensagens são prioritárias em relação às transmissões não periódicas.	NFR	Meta Operacional	Novo caso de uso Transmitir Mensagem 60 Segundos.
4	SIG2	Mensagens não periódicas devem ser enviadas conforme solicitação dos navios com dados das últimas 24 horas.	NFR	Meta Operacional	Novo caso de uso Transmitir Mensagem 24 Horas.
5	SIG3	Escolher uma arquitetura de software que permita utilizar princípios da coerência semântica (alta coesão, baixo acoplamento e diretrizes de reuso).	Tática	Restrição de Projeto	Definição da arquitetura de software a ser utilizada.
6	SIG4	O sistema deve solicitar uma confirmação de solicitação da fonte.	NFR	Meta Operacional	Novo caso de uso Confirmar Leitura
6A	SIG4	Melhorar o canal de comunicação para garantir o recebimento da informação sem ruído.	NFR	Restrição de Projeto	Melhoria do canal de comunicação.
7	SIG5	O sistema irá solicitar dos navios identificação, autenticação e autorização de acesso a fim de receber informações meteorológicas.	NFR	Meta operacional	Novo caso de uso Validar Acesso

Nesta tabela podem ser observadas as restrições de projeto a serem impostas ao SBM e os novos casos de uso a serem adicionados no modelo de casos de uso.

O RNF (id 1) foi considerado meta operacional, pois adiciona uma nova função ao SBM, através do caso de uso *Coletar Mensagem 30 segundos*. O requisito não-funcional (id 5) foi considerado restrição de projeto pois

estabelece que a arquitetura de software deve ser implementada de modo a permitir o uso dos princípios da coerência semântica.

A atividade para definir pontos de composição (5.2) tem a finalidade de identificar os pontos no fluxo de sequência de eventos dos casos de uso, onde as metas operacionais devem ser inseridas. A Tabela 31 apresenta a tabela de RNF com composição, que é a tabela de RNF com metas operacionais e restrições de projeto, acrescida das colunas: caso de uso, ponto de composição, regra de composição e condições da composição.

Tabela 31 – Tabela de RNF com composição (SBM).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição
1	Ler Sensores	1	sobreposição.antes	Nenhuma
2	Ler Sensores	1	sobreposição.antes	Nenhuma
3	Transmitir Mensagem	1	sobreposição.antes	Nenhuma
4	Transmitir Mensagem	2	sobreposição.depois	Nenhuma
6	Ler Sensores	1	sobreposição.depois	Nenhuma
7	Transmitir Mensagem	1	sobreposição.antes	Nenhuma

Todas as regras de composições definidas para o SBM são do tipo sobreposição, por isso, não é necessário o preenchimento da coluna condição da composição.

É possível verificar na Tabela 31 que todos os requisitos não-funcionais impactam apenas um caso de uso.

Um exemplo de composição relacionado ao RNF (id 1), é a inclusão do caso de uso *Coletar Mensagem 30 segundos*, definido na tabela de RNF com meta operacional e restrições de projeto (Tabela 30) para o RNF (id 1), antes da execução do primeiro passo do fluxo básico do caso de uso *Ler Sensores*.

A atividade para confirmar propagações (5.3) não detectou nenhuma necessidade de revisão nas propagações anteriormente definidas no SBM.

Na atividade resolver conflitos (5.4) foram detectados conflitos nos casos de uso *Ler Sensores* e *Transmitir de Mensagem*. Os conflitos podem ser detectados quando um filtro é aplicado no campo caso de uso da tabela de RNF com composição. A Tabela de RNF com prioridades (Tabela 32), que é a

tabela de RNF com composição acrescida da coluna prioridade da composição, apresenta os conflitos e a definição de prioridades para o caso de uso *Leitura de Sensores*.

Tabela 32 – Tabela de RNF com prioridades – caso de uso Leitura de Sensores (SBM).

Id	Interesse	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição	Prioridade da Composição
1	Desempenho	Leitura de Sensores	1	sobreposição.antes	Nenhuma	2
2	Desempenho	Leitura de Sensores	1	sobreposição.antes	Nenhuma	1
6	Acurácia	Leitura de Sensores	1	sobreposição.depois	Nenhuma	Nenhuma

Na tabela do caso de uso *Leitura de Sensores* a composição do RNF (id 2) prevalece em relação ao RNF (id 1), que recebem as prioridades 1 e 2, respectivamente. O RNF (id 2) é composto com prioridade 1, pois a leitura de 10 segundos deve ocorrer antes da leitura de 30 segundos.

O RNF de acurácia não participa da priorização, pois possui uma regra de composição diferente (sobreposição.depois) dos demais requisitos (sobreposição.antes), por isso, não há conflito.

A Tabela 33 apresenta a tabela de RNF com prioridades para o caso de uso *Transmitir de Mensagem*.

Tabela 33 – Tabela de RNF com prioridades – caso de uso Transmitir Mensagem (SBM).

Id	Interesse	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição	Prioridade da Composição
3	Desempenho	Transmitir Mensagem	1	sobreposição.antes	Nenhuma	2
4	Desempenho	Transmitir Mensagem	2	sobreposição.depois	Nenhuma	Nenhuma
7	Security	Transmitir Mensagem	1	sobreposição.antes	Nenhuma	1

Na tabela do caso de uso *Transmitir Mensagem*, a composição do RNF (id 7) prevalece em relação ao RNF (id 3), pois as mensagens somente são transmitidas caso o navio possua permissão de acesso para tal operação.

O RNF de desempenho (id 4) não participa da priorização, pois possui uma regra de composição diferente (sobreposição.depois) dos demais requisitos.

A atividade para atualizar o modelo do sistema de software (5.5) tem a finalidade de construir um modelo de casos de uso atualizado com os novos casos de uso incluídos. A Figura 66 apresenta o diagrama de casos de uso atualizado do SBM.

Figura 66 – Diagrama de casos de uso atualizado (SBM).

Os casos de uso destacados são os novos casos de uso incluídos no diagrama. Os casos de uso *Confirmar Leitura*, *Coletar Mensagem 30 segundos*, *Coletar Mensagem 10 segundos*, *Transmitir Mensagem 60 segundos*, *Validar Acesso* e *Transmitir Mensagem 24 horas* não são aspectos e, por isso, não possuem o estereótipo **<<aspecto>>**. A informação sobre os casos de uso que são ou não considerados aspectos deve ser consultada na tabela de RNF com prioridades, na coluna aspecto.

As especificações dos novos casos de uso incluídos também devem ser elaboradas. Um exemplo de especificação do caso de uso *Coletar Mensagem 30 segundos* é apresentado na Tabela 34.

Tabela 34 – Exemplo da especificação do caso de uso Coletar Mensagem 30 segundos (SBM).

Nome do Caso de Uso	Coletar Mensagem 30 segundos
Descrição Breve	Este caso de uso tem o objetivo de solicitar a coleta de informações a cada 30 segundos.
Atores	Relógio
Evento Inicial	Sinal de início de varredura de 30 segundos.

Fluxo Básico	1. O sistema solicita a coleta da informação de velocidade do vento. 2. O caso de uso termina.
Fluxos Alternativos	Nenhum.
Sub-fluxos	Nenhum.
Requisitos Especiais	1
Pré-condição	Informação disponível para leitura no sensor.
Pós-condição	Solicitação de coleta de informação de velocidade do vento enviada.
Pontos de Extensão	Nenhum.
Pontos de Inclusão	Nenhum.

O campo requisitos especiais deve ser preenchido com o identificador do requisito não-funcional que deu origem a esta nova especificação.

Demais especificações dos casos de uso incluídos no modelo final do SBM encontram-se em Bombonatti (2010).

5.3 Sistema Olhos no Céu

O sistema olhos no céu (ONC) é um sistema de gerenciamento de tráfego que possui integração de tecnologias de supervisão, comunicação e mapeamento digital (KOTSAKIS; KETSELIDIS, 2001). Esta aplicação foi selecionada pois possui interface de comunicação com diversos sistemas.

O objetivo do ONC é prover suporte para serviços de gerenciamento de frota de veículos, para informação customizada de movimentação, e para situações de crise, como: terremotos ou outros desastres naturais. O sistema ONC combina tecnologias terrestres e aéreas, que incluem comunicação aéreo-terrestre, supervisão em tempo real e técnicas de coleta de dados de tráfego.

O sistema ONC auxilia os usuários na visualização de informações de tráfego atualizadas pela integração de diversas fontes (imagens, mapas, FCD) em um sistema de informações geográficas conhecido como GIS. Adicionalmente, o ONC provê serviços de comunicação e supervisão com áreas urbanas específicas em casos quando a infra-estrutura de comunicação existente está congestionada ou desarranjada.

5.3.1 Sub-processo 1: Obter os requisitos do sistema de software

Os documentos de entrada consistiram da descrição do sistema de Kotsakis e Ketselidis (2001). Após a atividade de verificar a documentação de entrada (1.1), foi executada a atividade para elaborar os requisitos, que consistiu do levantamento do glossário, da elaboração da especificação de RNFs e da revisão do modelo de casos de uso.

O glossário do ONC é apresentado na Tabela 35.

Tabela 35 – Glossário (ONC).

Termo	Definição
Autoridade Pública	Ministro, autoridade de tráfego, serviço de proteção civil.
Companhia de táxi	Operadores de viagem, escritório central de companhia de táxi.
<i>Floating Car Data (FCD)</i>	É um método para determinar a velocidade do tráfego nas estradas. Este método é baseado na coleção de dados de localização, velocidade, direção de viagem e tempo para direção de veículos. Estes dados constituem fontes essenciais para a informação de tráfego e para a maior parte dos sistemas de transporte inteligente (ITS – <i>Intelligent Transportation Systems</i>). Isto significa que cada veículo apropriadamente equipado atua como um sensor para a estrada. Com base nestas informações, congestionamentos podem ser identificados, tempos de viagens podem ser calculadas e relatórios de tráfego podem ser instantaneamente gerados.
<i>Geographic Information System (GIS)</i>	Um sistema de informação geográfica que integra hardware, software, e dados, para capturar, gerenciar, analisar e exibir informações geográficas.
Informações pré-viagem	Informações pré-viagem incluem qualquer informação necessária antes da partida, como páginas web de interesse, condições do tráfego, tempo até o destino.
Motorista	Indivíduos, motoristas em geral.
Operador de Rede	Usuário de informações de tráfego para redistribuição aos assinantes.
Zeppelin	Um zeppelin é equipado com câmeras especiais que capturam imagens da área urbana. Estas imagens são armazenadas em um banco de dados e podem ser visualizadas em um GIS ou podem ficar disponíveis em mídias. Estas imagens podem melhorar a acurácia da estimativa do fluxo de tráfego em uma dada área.
<i>Global System for Mobile Communications (GSM)</i>	O sistema global para comunicações móveis é uma tecnologia móvel e o padrão mais popular para celulares. O seu sinal e os canais de voz são digitais.

A Figura 67 apresenta o diagrama de casos de uso.

Figura 67 – Diagrama de casos de uso (ONC).

Os atores do ONC são os seguintes:

- **Autoridade Pública:** é responsável por monitorar as condições de tráfego. Pode ser uma autoridade de tráfego ou um responsável pelo serviço de proteção civil.
- **Motorista:** é responsável por solicitar informações de tráfego e as condições das vias.
- **Operador de Rede:** é responsável por redistribuir informações de tráfego.
- **Companhia de Taxi:** é responsável por monitorar e comunicar as condições de tráfego para a frota de carros.

O caso de uso *Transferir Dados e Voz* permite estabelecer comunicação de voz entre os atores do sistema a fim de viabilizar a transferência de informações entre eles.

O caso de uso *Monitorar Céu* fornece imagens de tráfego de veículos de uma área solicitada.

O caso de uso *Monitorar e Informar Tráfego* envia informações sobre as condições de tráfego de veículos.

O caso de uso *Gerenciar Frota* tem a finalidade de visualizar dados da frota de veículos, exibir dados da frota, trocar informações, obter informações sobre a melhor rota e rastrear um determinado veículo.

O caso de uso *Informar Dados Pré-Viagem* tem o objetivo de informar o usuário sobre o caminho mais curto a ser percorrido para uma viagem.

A Tabela 36 apresenta uma especificação do caso de uso *Monitorar e Informar Tráfego*. A descrição dos seus campos encontra-se no Apêndice B. As demais especificações do ONC encontram-se em Bombonatti (2010).

Tabela 36 – Especificação do caso de uso Monitorar e Informar Tráfego (ONC).

Nome do Caso de Uso	Monitorar e Informar Tráfego
Descrição Breve	O objetivo deste caso de uso é informar condições de estradas e congestionamentos para o centro de monitoramento de tráfego e para os motoristas, através da combinação de algoritmos de deslocamentos de carros e imagens em tempo real.
Atores	Autoridade pública Motoristas Companhia de táxi
Evento Inicial	Solicitação de visualização de imagem de tráfego.
Fluxo Básico	<ol style="list-style-type: none"> 1. O ator solicita informação sobre uma área de interesse. 2. O ator solicita a opção FCD. 3. O sistema calcula e responde à solicitação com os dados de tráfego. 4. O ator visualiza as imagens de tráfego do FCD em um GIS. 5. O ator pode enviar a informação para outros motoristas ou arquivar para uso futuro. 6. O caso de uso termina.
Fluxos Alternativos	<p>2A. O ator solicita a opção para imagens em tempo real.</p> <p>2A1. O sistema obtém as imagens e responde à solicitação com as imagens solicitadas.</p> <p>2A2. O ator pode visualizar, armazenar ou enviar as imagens de tempo real da área de interesse.</p> <p>2B. FCD não disponível para a área solicitada.</p> <p>2B1. O sistema compara as imagens em tempo real ao FCD. O caso de uso <i>Monitorar Céu</i> é incluído para enviar as imagens coletadas a fim de que sejam comparadas ao FCD com o intuito de aumentar a confiança e acurácia das informações de condições de tráfego.</p> <p>2B2. O ator visualiza as imagens de tráfego.</p>
Sub-fluxos	Nenhum.
Requisitos Especiais	Nenhum.
Pré-condição	Disponibilidade de um sinal GPS e cobertura GSM.
Pós-condição	Nenhuma.
Pontos de Extensão	Nenhum.
Pontos de Inclusão	I1. O ponto de inclusão do caso de uso <i>Monitorar Céu</i> ocorre no passo 2B1 do sub-fluxo.

A especificação de requisitos não-funcionais do ONC é apresentada na Tabela 37.

Tabela 37 – Especificação de requisitos não-funcionais (ONC).

Id	Requisito	Prioridade
1	O sistema deve prover acurácia do posicionamento dos veículos.	1
2	O sistema deve ter capacidade de comunicação do canal para transmissão de dados e voz.	2
3	O sistema deve prover qualidade das imagens das áreas solicitadas.	2

Na Tabela 37, a prioridade mais alta foi estabelecida para o requisito relacionado a acurácia do posicionamento dos veículos (id 1). A segunda prioridade foi estabelecida para a transmissão de dados, voz e imagem (id 2 e id 3).

5.3.2 Sub-processo 2: Relacionar interesses funcionais e não-funcionais

O sub-processo 2, para relacionar interesses funcionais e não-funcionais, é composto das atividades para identificar interesses funcionais e não-funcionais (2.1) e integrar interesses funcionais e não-funcionais (2.2).

Na atividade para identificar os interesses funcionais e não-funcionais (2.1) foram utilizados o modelo de casos de uso e a especificação de requisitos não-funcionais, recebidos como dados de entrada para esta atividade.

Como resultado, foram gerados os *softgoals* apresentados na Figura 68.

Figura 68 – *Softgoals* (ONC).

Os *softgoals* representam os interesses não-funcionais do ONC que, por sua vez, podem representar um ou mais RNFs do mesmo tipo. A lista de interesses do ONC é apresentada na Tabela 38.

Tabela 38 – Tabela de interesses (ONC).

Interesse	Natureza
Transferir Dados e Voz	Funcional
Monitorar Céu	
Monitorar e Informar Tráfego	
Gerenciar Frota	

Interesse	Natureza
Informar Dados Pré-Viagem	
Acurácia	Não-Funcional
Desempenho	

Os RNFs devem ser classificados de acordo com os interesses não-funcionais definidos para o ONC. Esta classificação é apresentada na Tabela 39 que é a Tabela 37, acrescida de colunas referentes ao interesse e a origem do interesse.

Tabela 39 – Tabela de RNF (ONC).

Id	Requisito	Prioridade	Interesse	Origem do Tipo de Requisito
1	O sistema deve prover acurácia do posicionamento.	1	Acurácia	NFR
2	O sistema deve ter capacidade de comunicação do canal para transmissão de dados e voz.	2	Desempenho	NFR
3	O sistema deve prover qualidade das imagens das áreas solicitadas.	2	Desempenho	NFR

O interesse de acurácia é composto pelo RNF (id 1). O interesse de desempenho é composto dos RNFs (id 2 e id 3). A origem dos interesses de acurácia e desempenho é o catálogo do NFR *Framework*.

A atividade para integrar interesses funcionais e não-funcionais (2.2) utilizou as informações geradas na atividade 2.1 para que o diagrama integrado fosse elaborado. Este diagrama é apresentado na Figura 69.

Figura 69 – Diagrama integrado (ONC).

Neste diagrama, é possível observar as associações entre *softgoals* e os casos de uso *Transferir Dados e Voz*, *Monitorar Céu* e *Gerenciar Frota*. Estas associações são estabelecidas de acordo com as definições da seção 4.4.

O *softgoal* de acurácia está associado ao caso de uso *Gerenciar Frota*, pois o RNF (id 1) define que a obtenção dos dados de posicionamento dos veículos deve ser executada com precisão. O *softgoal* de desempenho está associado ao caso de uso *Transferir Dados e Voz*, pois o RNF (id 2) estabelece que a capacidade de transmissão de dados e voz deve ser realizada adequadamente, e ao caso de uso *Monitorar Céu*, em que o RNF (id 3) estabelece que as imagens das áreas solicitadas devem ter qualidade adequada.

5.3.3 Sub-processo 3: Identificar aspectos

O sub-processo 3, para identificar aspectos, é composto das atividades para identificar propagações (3.1) e identificar aspectos (3.2).

Na atividade para identificar propagações (3.1), o diagrama integrado e os SIGs iniciais são recebidos como dados de entrada. Nesta atividade as propagações relacionadas a cada um dos RNFs, que compõe um interesse não-funcional, devem ser analisadas.

As heurísticas para analisar as propagações de um RNF nos elementos do diagrama de casos de uso estão definidas na seção 4.5 deste trabalho.

O resultado desta análise pode ser observado na Tabela 40; esta tabela é a Tabela 39, acrescida de colunas ponto de associação e propagação.

Tabela 40 – Tabela de RNF com propagação (ONC).

Id	Requisito	Prioridade	Interesse	Origem do Tipo de Requisito	Ponto de Associação	Propagação
1	O sistema deve prover acurácia do posicionamento dos veículos.	1	Acurácia	NFR	Acurácia [Gerenciar Frota]	Monitorar e Informar Tráfego Monitorar Céu Informar Dados Pré-Viagem
2	O sistema deve ter capacidade de comunicação do canal para transmissão de dados e voz.	2	Desempenho	NFR	Desempenho [Transferir Dados e Voz]	Nenhuma
3	O sistema deve prover qualidade das imagens das áreas solicitadas.	2	Desempenho	NFR	Desempenho [Monitorar Céu]	Nenhuma

O RNF de acurácia (id 1) é propagado para os casos de uso *Monitorar e Informar Tráfego*, *Monitorar Céu* e *Informar Dados Pré-Viagem* pois todos manipulam a informação sobre o posicionamento dos veículos.

Os RNFs de desempenho (id 2 e id 3) associados aos casos de uso *Transmitir Dados e Voz* e *Monitorar Céu* não possuem nenhuma propagação, por serem específicos da capacidade de comunicação do canal para transmissão de dados e voz e à qualidade das imagens das áreas solicitadas.

A atividade para identificar aspectos (3.2) utiliza as heurísticas definidas na seção 4.5 deste trabalho. A Tabela 41 apresenta a tabela de RNF com aspectos, que é a tabela de propagações acrescida da coluna aspecto.

Tabela 41 – Tabela de RNF com aspectos (ONC).

Id	Requisito	Prioridade	Interesse	Origem do Tipo de Requisito	Ponto de Associação	Propagação	Aspecto
1	O sistema deve prover acurácia do posicionamento dos veículos.	1	Acurácia	NFR	Acurácia [Gerenciar Frota]	Monitorar e Informar Tráfego Monitorar Céu Informar Dados Pré-Viagem	Sim

Id	Requisito	Prioridade	Interesse	Origem do Tipo de Requisito	Ponto de Associação	Propagação	Aspecto
2	O sistema deve ter capacidade de comunicação do canal para transmissão de dados e voz.	2	Desempenho	NFR	Desempenho [Transferir Dados e Voz]	Nenhuma	Não
3	O sistema deve prover qualidade das imagens das áreas solicitadas.	2	Desempenho	NFR	Desempenho [Monitorar Céu]	Nenhuma	Não

Somente o RNF de acurácia foi considerado aspecto, pois possui propagação através de três casos de uso, que manipulam o posicionamento dos veículos.

Os RNFs de desempenho não possuem propagações e estão relacionados a apenas um elemento do diagrama de casos de uso e, por isso, não são considerados aspectos do ONC.

5.3.4 Sub-processo 4: Decompor requisitos não-funcionais

Este sub-processo é composto pelas atividades para selecionar SIG inicial a ser detalhado (4.1), decompor *softgoals* (4.2), definir prioridades (4.3), identificar operacionalizações (4.4), identificar correlações (4.5), registrar decisões de projeto (4.6), selecionar alternativas de projeto (4.7), aplicar o procedimento de avaliação (4.8) e definir relacionamentos de metas de operacionalizações (4.9). Todas estas atividades têm como principal objetivo elaborar os SIGs com base nos SIGs iniciais definidos no diagrama integrado.

A atividade para selecionar SIG inicial a ser detalhado (4.1) recebe como entrada o diagrama integrado, os SIGs iniciais e a especificação de requisitos não-funcionais. A seleção do SIG inicial para o detalhamento é feita considerando as prioridades dos RNFs contidas na especificação de requisitos não-funcionais (Tabela 37).

Para ilustrar a atividade para decompor requisitos não-funcionais serão apresentados os RNFs de acurácia e desempenho, que possuem, respectivamente, as prioridades 1 e 2.

O SIG que representa o RNF de acurácia associado ao caso de uso *Gerenciar Frota* é apresentado na Figura 70.

Figura 70 – SIG1 de acurácia (ONC).

Durante a construção deste SIG, foram utilizados os métodos de decomposição de limite de sistema e de casos de uso, propostos na seção 4.6.1. Como a propagação do RNF de acurácia ocorreu em casos de uso que possuem o relacionamento de inclusão entre si (*Gerenciar Frota*, *Monitorar e*

Informar Tráfego e Monitorar Céu), e o caso de uso *Informar Dados Pré-Viagem*, que não possui relacionamento com nenhum outro caso de uso, foi necessário iniciar a decomposição do softgoal de acurácia através do sistema ONC. O ONC foi decomposto em seus casos de uso *Gerenciar Frota* e *Informar Dados Pré-Viagem* e, posteriormente, os relacionamentos de inclusão entre os casos de uso *Gerenciar Frota*, *Monitorar* e *Informar Tráfego e Monitorar Céu* puderam ser expressos no SIG.

Este SIG também utiliza o catálogo de decomposição do RNF de acurácia do NFR *Framework*. O método utilizado para refinar o *softgoal* de acurácia foi o de conservação.

Os *softgoals* de operacionalização relativos a atribuir recurso e melhor fluxo de informação foram identificados para o ONC. O *softgoal* atribuir recurso tem a finalidade de utilizar o sistema GPS como fonte precisa da informação sobre o posicionamento. O *softgoal* melhor fluxo de informação foi decomposto, de acordo com o catálogo do NFR *Framework*, até a definição da meta que diz respeito à utilização de um equipamento compatível com o sinal GSM, a fim de permitir o recebimento de informações de tráfego pelos motoristas. Esta compatibilidade torna-se necessária pois os dados de tráfego são transmitidos aos motoristas através de um canal *wireless* GSM-SMS.

Neste SIG, duas metas foram identificadas para o mesmo requisito não-funcional e, por isso, a recomendação é que o requisito original seja desmembrado em dois, a fim de proporcionar maior clareza à especificação. O requisito (id 1A) foi adicionado à especificação com o seguinte texto: “O sistema deve prover acurácia do posicionamento através de equipamentos compatíveis com o sinal a ser recebido”.

O SIG relacionado ao *softgoal* de desempenho, associado ao caso de uso *Transferir Dados*, é apresentado na Figura 71.

Figura 71 – SIG 2 de desempenho (ONC).

Neste SIG foi identificado somente um *softgoal* de operacionalização para aumentar a disponibilidade de recursos. O objetivo é que a capacidade de transmissão de voz e dados possa ser melhorada.

O SIG relacionado ao *softgoal* de desempenho, associado ao caso de uso *Monitorar Céu*, é apresentado na Figura 72.

Figura 72 – SIG 3 de desempenho (ONC).

Similarmente ao SIG 2 o *softgoal* aumentar a disponibilidade de recursos foi utilizado para que as imagens de tráfego possam ter a qualidade adequada.

Em ambos os SIGs, das Figuras 71 e 72, detectaram-se duas correlações negativas entre os RNFs de desempenho e de custos. Nesta situação, o engenheiro de requisitos ressalta que a adição de recursos computacionais mais poderosos, para melhorar a comunicação de dados, voz e a qualidade da imagem, refletem diretamente nos custos adicionais.

5.3.5 Sub-processo 5: Elaborar modelo composto

Este sub-processo é composto pelas atividades para identificar metas operacionais e restrições de projeto (5.1), definir pontos de composição (5.2), confirmar propagação (5.3), resolver conflitos (5.4) e atualizar o modelo do sistema de software (5.5).

A atividade para identificar metas operacionais e restrições de projeto (5.1) recebe como dados de entrada os SIGs e a tabela de RNF com aspectos. Nesta atividade deve ser elaborada a tabela de RNF com metas operacionais e restrições de projeto; que é a tabela de aspectos (Tabela 41) acrescida das colunas: id SIG, meta, origem da meta, meta operacional ou restrição de projeto e descrição. Esta tabela tem a finalidade de registrar a classificação do tipo de meta referente a cada SIG e a ação a ser tomada caso a meta seja classificada como meta operacional ou restrição de projeto.

A tabela de RNF com metas operacionais e restrições de projeto é apresentada na Tabela 42.

Tabela 42 – Tabela de RNF com metas operacionais e restrições de projeto (ONC).

Id	Id SIG	Meta	Origem da Meta	Meta Operacional ou Restrição de Projeto	Descrição
1	SIG1	Uso do sinal de GPS como fonte precisa de posicionamento.	NFR	Meta Operacional	Novo caso de uso Verificar Posicionamento
1A	SIG1	Uso de equipamento compatível com GSM a fim de permitir o recebimento de informações visuais do tráfego.	NFR	Restrição de Projeto	Uso obrigatório de equipamento para recebimento de imagens compatível com GSM.
2	SIG2	Aumentar os recursos computacionais a fim de aumentar a capacidade de transmissão de voz e dados.	Tática	Restrição de Projeto	Definição de arquitetura de hardware com recursos computacionais compatíveis com a capacidade de comunicação desejada.
3	SIG3	Aumentar os recursos computacionais a fim de aumentar a qualidade da imagem.	Tática	Restrição de Projeto	Definição de arquitetura de hardware com recursos computacionais compatíveis com a qualidade de imagem desejada.

Na representação da tabela de RNF com metas operacionais e restrições (Tabela 42) foram apresentadas as colunas referentes ao conteúdo da tabela de RNF com metas operacionais e restrições de projeto e o identificador dos requisitos, para auxiliar o entendimento desta última atividade.

Nesta tabela podem ser observadas as restrições de projeto a serem impostas ao ONC e o novo caso de uso, *Verificar Posicionamento*, a ser adicionado no modelo de casos de uso.

O RNF (id 1) foi considerado meta operacional, pois irá adicionar, através do caso de uso *Verificar Posicionamento*, uma nova função no ONC, para que o sinal GPS seja utilizado como fonte precisa de informação. O RNF (id 2) foi considerado restrição de projeto, pois a meta aumentar os recursos computacionais interfere diretamente na definição da arquitetura de hardware do ONC.

A atividade para definir pontos de composição (5.2) tem a finalidade de identificar os pontos no fluxo de seqüência de eventos dos casos de uso, onde as metas operacionais devem ser inseridas. A Tabela 43 apresenta a tabela de RNF com composição, que é a tabela de RNF com metas operacionais e restrições de projeto (Tabela 43), acrescida das colunas: caso de uso, ponto de composição, regra de composição e condições da composição.

Tabela 43 – Tabela de RNF com composição (ONC).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição	Prioridade da Composição
1	Gerenciar Frota	2	sobreposição.antes	Nenhuma	Nenhuma
	Monitorar e Informar Tráfego	3	sobreposição.antes	Nenhuma	Nenhuma
	Monitorar Céu	2	sobreposição.antes	Nenhuma	Nenhuma
	Informar Dados Pré-Viagem	5	sobreposição.antes	Nenhuma	Nenhuma

Todas as regras de composições definidas para o ONC são do tipo sobreposição, por isso, não é necessário o preenchimento da coluna condição da composição.

Um exemplo de composição relacionado ao RNF (id 1), é a inclusão do caso de uso *Verificar Posicionamento*, definido na tabela de RNF com meta operacional e restrições de projeto (Tabela 42) para o RNF (id 1), antes da execução do segundo passo do fluxo básico do caso de uso *Gerenciar Frota*.

A atividade para confirmar propagações (5.3) não detectou nenhuma necessidade de revisão nas propagações iniciais do ONC.

Na atividade para resolver conflitos (5.4) não foram detectados conflitos entre casos de uso do ONC.

Na atividade para atualização do modelo do sistema de software (5.5), um novo diagrama de casos de uso deve ser gerado, com os novos elementos incluídos. A Figura 73 apresenta o diagrama de casos de uso atualizado.

Figura 73 – Diagrama de casos de uso atualizado (ONC).

O caso de uso destacado é o novo caso de uso incluído no diagrama. O caso de uso *Verificar Posicionamento* possui o estereótipo <<aspecto>>, pois é incluído por diversos casos de uso do ONC.

A especificação do novo caso de uso incluído foi elaborada e é apresentada na Tabela 44.

Tabela 44 – Especificação do caso de uso Verificar Posicionamento (ONC).

Nome do Caso de Uso	Verificar Posicionamento
Descrição Breve	O objetivo deste caso de uso é verificar o posicionamento do veículo.
Atores	GPS
Evento Inicial	Solicitação de posicionamento de veículo pelos casos de uso Monitorar Céu, Monitorar e Informar Tráfego, Gerenciar Frota, Informar Dados Pré-Viagem.
Fluxo Básico	<ol style="list-style-type: none"> 1. O sistema solicita cálculo do posicionamento ao sistema GPS. 2. O sistema recebe informação do posicionamento. 3. O caso de uso termina.
Fluxos Alternativos	Nenhum.
Sub-fluxos	Nenhum.
Requisitos Especiais	1
Pré-condição	Sinal GPS ativo.
Pós-condição	Posicionamento detectado.
Pontos de Extensão	Nenhum.
Pontos de Inclusão	Nenhum.

O campo requisitos especiais deve ser preenchido com o identificador do RNF que deu origem à nova especificação.

5.4 Sistema de Controle Automático de Trens

O sistema de controle automático de trens, conhecido como ATC (*Automatic Train Control*) tem a finalidade de controlar a movimentação de um trem de forma segura (*safety*). Esta aplicação foi selecionada pois trata-se de um sistema crítico de tempo real.

A descrição do ATC foi feita com base nos documentos do projeto de um sistema deste tipo, porém o produto foi descaracterizado, por ser um produto comercial. Por isso, os documentos não estão disponíveis na referência.

O ATC permite a operação do trem em três modos de operação: automático, semi-automático e manual.

No modo automático, o ATC realiza todas as operações relativas à operação do trem. As intervenções do operador ocorrem através de requisições para atuação no freio de emergência, para a seleção do nível de restrição e para comando de baixa aderência. No modo semi-automático o ATC se

restringe às funções de tratamento de velocidade, controle de tração e frenagem, supervisão de portas e comunicação com a estação. No modo manual, todas as funções de operação automática são desativadas, com exceção da comunicação com a estação.

Na aplicação do PARNAFOA foi considerado o modo de operação automático que foi considerado significativo para a avaliação.

5.4.1 Sub-processo 1: Obter os requisitos do sistema de software

Os dados de entrada foram estruturados a partir de documentos de um sistema ATC, eliminando-se os dados que o caracterizassem. Após a atividade de verificar a documentação de entrada (1.1), foi executada a atividade para elaborar os requisitos, que consistiu do levantamento do glossário e da elaboração da especificação de RNFs, do modelo de casos de uso e do diagrama de estados do trem.

O glossário do ATC é apresentado na Tabela 45.

Tabela 45 – Glossário (ATC).

Termo	Descrição
Código de Velocidade	Código que corresponde ao valor máximo da velocidade permitida no trecho.
Sinal de Via	Sinal recebido do equipamento da via contendo o código de velocidade fornecido pela via.
Velocidade de Referência	Velocidade de referência para o controle de velocidade do ATO.
Velocidade Máxima Permitida	Velocidade calculada pelo ATC do trem em função do modo de operação, código de velocidade recebido e níveis de restrição impostos.
Velocidade Decodificada	Valor da velocidade correspondente ao código de velocidade.
Velocidade Real	Velocidade do trem medida através dos tacômetros.
Primeiro Freio de Emergência	Freio utilizado quando a velocidade real é igual à velocidade máxima permitida.
Máximo Freio de Serviço	Freio utilizado quando a velocidade real é menor que a velocidade máxima permitida.
Máximo Freio de Emergência	Freio utilizado quando a velocidade real é maior que a velocidade máxima permitida.
Baixa aderência	Restrição aplicada pelo operador durante condições de chuva. A velocidade máxima permitida é reduzida.
Nível de restrição	Restrição nas velocidades limites permitidas ao trem por perdas na capacidade de frenagem.
Velocidade de partida	Valor mínimo de velocidade para partida do trem.

A Figura 74 apresenta o diagrama de casos de uso.

Figura 74 – Diagrama de casos de uso (ATC).

Os atores do ATC são os seguintes:

- Relógio: é responsável pelo fornecimento dos pulsos a cada milissegundo.
- Acelerômetro: é um equipamento de via responsável pela medição da aceleração do trem.
- ATO (*Automatic Train Operacion*) do Trem: é responsável pela execução dos comandos de abertura e fechamento de portas.
- Tacômetro: é um equipamento de via responsável pela medição dos dados para cálculo da velocidade do trem.
- Sistema de Portas: é responsável pela abertura e fechamento das portas do trem.
- ATO da Estação: é responsável pelo envio dos sinais de parada programada e comando de portas.
- Antena do Trem: é um equipamento do trem responsável pela leitura do sinal de via.
- Operador: é responsável por executar comandos após o recebimento de indicações de falhas e alarmes.

- Sistema de Freios: é responsável pela aplicação do freio no trem.

O caso de uso *Verificação de Limite de Velocidade* tem o objetivo de decodificar o código de velocidade a fim de verificar o limite da velocidade permitida ao trem.

O caso de uso *Calcular Velocidade Máxima* tem o objetivo de calcular a velocidade máxima permitida ao trem.

O caso de uso *Proteção Sobre Velocidade* define as ações a serem tomadas caso o trem esteja com a velocidade acima da velocidade máxima permitida. Caso o trem esteja com velocidade constante, é acionado o caso de uso *Proteção de Velocidade Constante* e, se o trem estiver com velocidade em transição é acionado o caso de uso *Proteção de Velocidade em Transição*.

O caso de uso *Controle de Partida* tem o objetivo de verificar as condições a serem satisfeitas para que a partida do trem seja habilitada.

O caso de uso *Controle de Abertura de Portas* solicita a abertura de portas e o teste dos freios no momento em que o trem chega na plataforma, depois de uma viagem.

O caso de uso *Controle do Sistema de Freio* executa o teste nos freios do trem e atribui o nível de restrição, de acordo com a situação dos freios.

O caso de uso *Controle de Fechamento de Portas* verifica as condições para a partida do trem e solicita o fechamento das portas.

O caso de uso *Cálculo de Velocidade e Aceleração* tem a finalidade de verificar a velocidade e aceleração real do trem.

A Tabela 46 apresenta uma especificação do caso de uso *Verificação de Limite de Velocidade*. A descrição dos seus campos encontra-se no Apêndice B. As demais especificações do ATC encontram-se em Bombonatti (2010).

Tabela 46 – Especificação do caso de uso Verificação de Limite de Velocidade (ATC).

Nome do Caso de Uso	Verificação de Limite de Velocidade
Descrição Breve	Este caso de uso lê e decodifica o código de velocidade para verificar o limite de velocidade.
Atores	Relógio
Evento Inicial	Pulso do relógio para verificação do limite de velocidade.
Fluxo Básico	<ol style="list-style-type: none"> 1. O ATC calcula a velocidade real. O caso de uso <i>Cálculo de Velocidade e Aceleração</i> é incluído para calcular a velocidade real. 2. O ATC calcula a velocidade máxima permitida. O caso de uso <i>Cálculo de Velocidade Máxima</i> é incluído para calcular a velocidade máxima permitida. 3. O ATP verifica se o trem está com código de velocidade constante ou em

	<p>transição.</p> <p>4. O ATC compara a velocidade máxima permitida com a velocidade real. {<i>Sobre Velocidade</i>}</p> <p>5. O caso de uso base termina.</p>
Fluxos Alternativos	<p><u>3A. Código de velocidade constante.</u></p> <p>3A1. O ATC envia o valor igual ao da velocidade máxima permitida como referência para o ATO.</p> <p><u>3B. Código de velocidade em transição.</u></p> <p>3B1. Se a velocidade estiver em transição para valor menor o ATC envia uma velocidade de referência com o valor reduzido para o ATO.</p> <p>3B2. Se a velocidade estiver em transição para valor maior o ATC envia a velocidade máxima permitida como referência para o ATO.</p>
Sub-fluxos	Nenhum.
Requisitos Especiais	Nenhum.
Pré-condição	Trem em movimento.
Pós-condição	Velocidade dentro do limite de tolerância.
Pontos de Extensão	E3. O ponto de extensão { <i>Sobre Velocidade</i> } ocorre no passo 4 do fluxo básico quando a velocidade real é maior do que a velocidade máxima permitida.
Pontos de Inclusão	<p>I1. O ponto de inclusão do caso de uso <i>Cálculo de Velocidade e Aceleração</i> ocorre no passo 1 do fluxo básico.</p> <p>I2. O ponto de inclusão do caso de uso <i>Cálculo de Velocidade Máxima</i> ocorre no passo 2 do fluxo básico.</p>

A especificação de requisitos não-funcionais do ATC é apresentada na Tabela 47.

Tabela 47 – Especificação de requisitos não-funcionais (ATC).

Id	Requisito	Prioridade
1	O ATC deverá detectar corretamente o sinal de código de velocidade existente nos circuitos de via.	1
2	O ATC deverá medir a velocidade real do trem com precisão.	2
3	O ATC deverá detectar de forma correta a velocidade zero.	3
4	O ATC deve detectar de forma correta o real estado das portas (abertas ou fechadas).	4
5	O ATC deve interpretar corretamente o sinal de indicação do número de carros com problemas nos freios, para efeito de indicação do nível de restrição a ser imposto ao trem.	5
6	O nível de restrição deve ser determinado corretamente em função da interpretação do sinal de problemas nos freios.	6
7	O ATC só deve permitir a habilitação de partida do trem quando todas as portas estiverem fechadas, o trem estiver parado, houver a detecção de código de velocidade compatível com o perfil da via.	7
8	O ATC deverá aplicar corretamente o corte da tração e acionamento máximo do freio de serviço.	8
9	O ATC deverá aplicar corretamente o freio de emergência.	9
10	O ATC deverá abrir as portas do trem do lado correto.	10

Na Tabela 47, as prioridades do ATC foram apresentadas seqüencialmente para que os requisitos fossem analisados de maneira uniforme, na ordem em que aparecem.

O diagrama de estados do trem, recebido como dado de entrada, é apresentado na Figura 75.

Figura 75 – Diagrama de estados do trem (ATC).

Os estados do trem são os seguintes:

- **Parado na Plataforma:** é o estado em que o trem permanece parado na plataforma; o sinal de parada programada ativa deve estar ativo.
- **Partida:** os freios do trem são liberados e o comando de propulsão é aplicado.
- **Movimento:** é o estado em que o trem está em movimento; o trem controla a velocidade, aceleração, sobre-velocidade e velocidade máxima permitida.
- **Parado:** é o estado em que o processo de desaceleração do trem é iniciado, até sua parada. Durante este estado, a partir do momento em que o trem se aproxima da plataforma e atinge uma velocidade mínima, inicia-se o procedimento de abertura de portas.
- **Parada de Emergência:** é o estado em que o trem está parado devido a situações de emergência no meio do percurso.

Durante o estado inicial em que o trem está parado na plataforma, ocorre a abertura de portas, o teste de freios e a definição do nível de restrição a ser aplicado. Para que o trem saia do estado parado na plataforma e atinja o estado de partida, é necessário que as portas estejam fechadas, o código de velocidade seja diferente de zero e o sinal de parada programada esteja não ativo. O trem permanece na plataforma, caso uma das condições descritas não seja atendida. Durante a partida, os freios são liberados e o comando de propulsão é aplicado. A transição entre o estado de partida para o estado de movimento ocorre quando o trem estiver em velocidade de partida.

Durante o movimento o trem pode entrar no estado de parada de emergência, em um local diferente da plataforma. Para que isso ocorra, a velocidade do trem deve ser igual a zero. Para que o trem saia do estado de parada de emergência e avance até o estado de partida, é necessário que as portas estejam fechadas e a velocidade seja diferente de zero.

Quando o trem estiver em movimento, em direção à plataforma, deve ser executado o controle de velocidade e aceleração, a proteção contra sobre-velocidade e a identificação da velocidade máxima permitida.

Quanto o trem se aproxima da plataforma, o sinal de parada programada começa a ser recebido pelo trem. Nesta situação, o trem entra no estado parando. O processo de desaceleração é iniciado, a partir do momento em que o trem aproxima-se da plataforma e atinge uma velocidade mínima. A partir da velocidade mínima, o trem inicia o procedimento de abertura de portas. Quando a velocidade do trem estiver igual a zero, o trem estará novamente no estado parado na plataforma.

5.4.2 Sub-processo 2: Relacionar interesses funcionais e não-funcionais

O sub-processo 2, para relacionar interesses funcionais e não-funcionais, é composto das atividades para identificar interesses funcionais e não-funcionais (2.1) e integrar interesses funcionais e não-funcionais (2.2).

Na atividade para identificar os interesses funcionais e não-funcionais (2.1) o modelo de casos de uso, a especificação de requisitos não-funcionais e o

diagrama de estados do trem, foram recebidos como dados de entrada para esta atividade.

Como resultado desta foram gerados os *softgoals* apresentados na Figura 76.

Figura 76 – *Softgoals* (ATC).

Os *softgoals* representam os interesses não-funcionais do ATC, que por sua vez podem representar um ou mais RNFs do mesmo tipo. A lista de interesses do ATC é apresentada na Tabela 48.

Tabela 48 – Tabela de interesses (ATC).

Interesse	Natureza
Verificação de Limite de Velocidade	Funcional
Cálculo de Velocidade Máxima	
Proteção Sobre Velocidade	
Cálculo de Velocidade e Aceleração	
Proteção de Velocidade Constante	
Proteção de Velocidade em Transição	
Controle de Partida	
Controle de Abertura de Portas	
Controle de Fechamento de Portas	
Controle de Sistema de Freios	
Acurácia	Não-Funcional
Confiabilidade	
Safety	

Os RNFs devem ser classificados de acordo com os interesses não-funcionais definidos para o ATC. Esta classificação é apresentada na Tabela 49 que é a Tabela 48, acrescida de colunas referentes ao interesse e a origem do interesse.

Tabela 49 – Tabela de RNF (ATC).

Id	Requisito	Prioridade	Interesse	Origem do Interesse
1	O ATC deverá detectar corretamente o sinal de código de velocidade existente nos circuitos de via.	1	Confiabilidade	ISO/IEC 9126
2	O ATC deverá medir a velocidade real do trem com precisão.	2	Acurácia	NFR
3	O ATC deverá detectar de forma correta a velocidade zero.	3	Acurácia	NFR
4	O ATC deve detectar de forma correta o real estado das portas (abertas ou fechadas).	4	Acurácia	NFR

Id	Requisito	Prioridade	Interesse	Origem do Interesse
5	O ATC deve interpretar corretamente o sinal de indicação do número de carros com problemas nos freios, para efeito de indicação do nível de restrição a ser imposto ao trem.	5	Acurácia	NFR
6	O nível de restrição deve ser determinado corretamente em função da interpretação do sinal de problemas nos freios.	6	Safety	ISO/IEC 9126
7	O ATC só deve permitir a habilitação de partida do trem quando todas as portas estiverem fechadas, o trem estiver parado, houver a detecção de código de velocidade compatível com o perfil da via.	7	Safety	ISO/IEC 9126
8	O ATC deverá aplicar corretamente o corte da tração e acionamento máximo do freio de serviço.	8	Safety	ISO/IEC 9126
9	O ATC deverá aplicar corretamente o freio de emergência.	9	Safety	ISO/IEC 9126
10	O ATC deverá abrir as portas do trem do lado correto.	10	Safety	ISO/IEC 9126

O interesse de confiabilidade é composto pelo RNF (id 1). O interesse de acurácia é composto dos RNFs (id 2, id 3, id 4 e id 5). O interesse de *safety* é composto dos RNFs (id 6, id 7, id 8 e id 9). A origem dos interesses de acurácia é o NFR *Framework*, enquanto que a origem dos interesses de confiabilidade e *safety* é a norma ISO/IEC 9126 (ISO/IEC 9126, 2001).

A atividade para integrar interesses funcionais e não-funcionais (2.2) utilizou as informações geradas na atividade 2.1 para que o diagrama integrado fosse elaborado. Este diagrama é apresentado na Figura 76.

Figura 77 – Diagrama integrado (ATC).

Neste diagrama é possível observar associações entre *softgoals* e casos de uso, e entre *softgoals* e relação ator-caso de uso. Nenhuma associação entre *softgoals* e o limite do sistema ocorreu no ATC.

O *softgoal* de confiabilidade está associado à relação *Antena do Trem-Cálculo de Velocidade Máxima*, pois o RNF (id 1) define que o sinal de velocidade deve ser corretamente detectado. O *softgoal* de acurácia está associado à relação *Tacômetro-Cálculo de Velocidade e Aceleração*, pois os RNFs (id 2 e id 3) estabelecem que a velocidade real e a velocidade zero devem ser medidas com precisão. O *softgoal* de acurácia está também associado às relações *Sistema de Portas-Control de Fechamento de Portas*, *Sistema de Portas-Control de Abertura de Portas* e *Sistema de Portas-Control de Partida*, pois o RNF (id 4) define que o ATC deve detectar corretamente o real estado das portas. Além disso, o *softgoal* de acurácia também se associa à relação *Sistema de Freios-Control de Sistema de Freios*, pois o requisito não-funcional (id 5) estabelece que o ATC deve detectar corretamente o estado dos freios.

O *softgoal* de *safety* está associado ao caso de uso *Controle de Sistema de Freios*, pois o RNF (id 6) define que o nível de restrição deve ser atribuído corretamente, conforme situação dos freios. O *softgoal* de *safety* também está associado ao caso de uso *Controle de Partida*, pois o RNF (id 7) estabelece que o trem somente poderá partir caso condições como: portas fechadas, trem parado e código de velocidade compatível com o perfil de via, forem satisfeitas. O *softgoal* de *safety* também deve ser associado ao caso de uso *Proteção Sobre Velocidade*, pois os RNFs (id 8 e id 9) definem que o ATC deve aplicar corretamente os freios de serviço e de emergência. Finalmente, o *softgoal* de *safety* se associa ao caso de uso *Controle de Abertura de Portas*, pois o RNF (id 10) estabelece que o ATC deverá abrir a porta do lado correto.

5.4.3 Sub-processo 3: Identificar aspectos

O sub-processo 3, para identificar aspectos, é composto das atividades para identificar propagações (3.1) e identificar aspectos (3.2).

Na atividade para identificar propagações (3.1), o diagrama integrado e os SIGs iniciais são recebidos como dados de entrada e as propagações relacionadas a cada um dos requisitos não-funcionais, que compõe um interesse não-funcional, devem ser analisadas.

As heurísticas para analisar as propagações de um requisito não-funcional nos elementos do diagrama de casos de uso estão definidas na seção 4.5 deste trabalho.

O resultado desta análise pode ser observado na Tabela 50 que é a Tabela 49, acrescida de colunas ponto de associação e propagação.

Tabela 50 – Tabela de RNF com propagação (ATC).

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
1	O ATC deverá detectar corretamente o sinal de código de velocidade existente nos circuitos de via.	1	Confiabilidade	ISO/IEC 9126	Confiabilidade [Antena do Trem- Cálculo de Velocidade Máxima]	Nenhuma
2	O ATC deverá medir a velocidade real do trem com precisão.	2	Acurácia	NFR	Acurácia [Tacômetro- Cálculo de Velocidade e Aceleração]	Nenhuma

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
3	O ATC deverá detectar de forma correta a velocidade zero.	3	Acurácia	NFR	Acurácia [Tacômetro-Cálculo de Velocidade e Aceleração]	Nenhuma
4	O ATC deve detectar de forma correta o real estado das portas (abertas ou fechadas).	4	Acurácia	NFR	Acurácia[Sistema de Portas-Controle de Partida, Sistema de Portas-Controle de Abertura de Portas, Sistema de Portas-Controle de Fechamento de Portas]	Nenhuma
5	O ATC deve interpretar corretamente o sinal de indicação do número de carros com problemas nos freios, para efeito de indicação do nível de restrição a ser imposto ao trem.	5	Acurácia	NFR	Acurácia[Sistema de Freios-Controle de Sistema de Freios]	Nenhuma
6	O nível de restrição deve ser determinado corretamente em função da interpretação do sinal de problemas nos freios.	6	<i>Safety</i>	ISO/IEC 9126	Safety [Controle de Sistema de Freios]	Nenhuma
7	O ATC só deve permitir a habilitação de partida do trem quando todas as portas estiverem fechadas, o trem estiver parado, houver a detecção de código de velocidade compatível com o perfil da via.	7	<i>Safety</i>	ISO/IEC 9126	Safety [Controle de Partida]	Nenhuma
8	O ATC deverá aplicar corretamente o corte da tração e acionamento máximo do freio de serviço.	8	<i>Safety</i>	ISO/IEC 9126	Safety [Proteção Sobre Velocidade]	Proteção de Velocidade Constante, Proteção de Velocidade em Transição
9	O ATC deverá aplicar corretamente o freio de emergência.	9	<i>Safety</i>	ISO/IEC 9126	Safety [Proteção Sobre Velocidade]	Proteção de Velocidade Constante, Proteção de Velocidade em Transição
10	O ATC deverá abrir as portas do trem do lado correto.	10	<i>Safety</i>	ISO/IEC 9126	Safety [Controle de Abertura de Portas]	Nenhuma

As propagações ocorreram somente nos requisitos não-funcionais (id 8 e id 9), pois o caso de uso de generalização *Proteção Sobre Velocidade* propagou os requisitos não-funcionais para seus casos de uso especializados, *Proteção de Velocidade Constante e Proteção de Velocidade em Transição*.

A atividade para identificar aspectos utiliza as heurísticas definidas na seção 4.5 deste trabalho. A Tabela 51 apresenta tabela de RNF com aspectos, que é a tabela de propagações acrescida da coluna aspecto.

Tabela 51 – Tabela de RNF com aspectos (ATC).

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação	Aspecto
1	O ATC deverá detectar corretamente o sinal de código de velocidade existente nos circuitos de via.	1	Confiabilidade	ISO/IEC 9126	Confiabilidade [Antena do Trem- Cálculo de Velocidade Máxima]	Nenhuma	Não
2	O ATC deverá medir a velocidade real do trem com precisão.	2	Acurácia	NFR	Acurácia [Tacômetro- Cálculo de Velocidade e Aceleração]	Nenhuma	Não
3	O ATC deverá detectar de forma correta a velocidade zero.	3	Acurácia	NFR	Acurácia [Tacômetro- Cálculo de Velocidade e Aceleração]	Nenhuma	Não
4	O ATC deve detectar de forma correta o real estado das portas (abertas ou fechadas).	4	Acurácia	NFR	Acurácia[Sistema de Portas-Controle de Partida, Sistema de Portas-Controle de Abertura de Portas, Sistema de Portas-Controle de Fechamento de Portas]	Nenhuma	Sim
5	O ATC deve interpretar corretamente o sinal de indicação do número de carros com problemas nos freios, para efeito de indicação do nível de restrição a ser imposto ao trem.	5	Acurácia	NFR	Acurácia[Sistema de Freios-Controle de Sistema de Freios]	Nenhuma	Não
6	O nível de restrição deve ser determinado corretamente em função da interpretação do sinal de problemas	6	Safety	ISO/IEC 9126	Safety [Controle de Sistema de Freios]	Nenhuma	Não

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação	Aspecto
	nos freios.						
7	O ATC só deve permitir a habilitação de partida do trem quando todas as portas estiverem fechadas, o trem estiver parado, houver a detecção de código de velocidade compatível com o perfil da via.	7	Safety	ISO/IEC 9126	Safety [Controle de Partida]	Nenhuma	Não
8	O ATC deverá aplicar corretamente o corte da tração e acionamento máximo do freio de serviço.	8	Safety	ISO/IEC 9126	Safety [Proteção Sobre Velocidade]	Proteção de Velocidade Constante, Proteção de Velocidade em Transição	Sim
9	O ATC deverá aplicar corretamente o freio de emergência.	9	Safety	ISO/IEC 9126	Safety [Proteção Sobre Velocidade]	Proteção de Velocidade Constante, Proteção de Velocidade em Transição	Sim
10	O ATC deverá abrir as portas do trem do lado correto.	10	Safety	ISO/IEC 9126	Safety [Controle de Abertura de Portas]	Nenhuma	Não

O RNF (id 4) de acurácia é considerado aspecto pois está associado a mais de um elemento do diagrama de casos de uso. Os RNFs (id 8 e id 9) também são considerados aspectos, pois se propagam através de dois casos de uso *Proteção de Velocidade Constante* e *Proteção de Velocidade em Transição*.

5.4.4 Sub-processo 4: Decompor requisitos não-funcionais

Este sub-processo é composto pelas atividades para selecionar SIG inicial a ser detalhado (4.1), decompor *softgoals* (4.2), definir prioridades (4.3), identificar operacionalizações (4.4), identificar correlações (4.5), registrar decisões de projeto (4.6), selecionar alternativas de projeto (4.7), aplicar o procedimento de avaliação (4.8) e definir relacionamentos de metas de

operacionalizações (4.9). Todas estas atividades têm como principal objetivo elaborar os SIGs com base nos SIGs iniciais definidos no diagrama integrado.

A atividade para selecionar SIG inicial a ser detalhado (4.1) recebe como entrada o diagrama integrado, os SIGs iniciais e a especificação de requisitos não-funcionais. A seleção do SIG inicial para o detalhamento é feita considerando as prioridades dos RNFs contidas na especificação de requisitos não-funcionais (Tabela 47).

Para ilustrar a atividade para decompor requisitos não-funcionais são apresentados exemplos dos RNFs de confiabilidade, acurácia e *safety* que possuem, respectivamente, as prioridades 1, 4, 8 e 9.

O SIG que representa o RNF de confiabilidade associado à relação *Antena do Trem-Cálculo de Velocidade Máxima* é apresentado na Figura 78.

Figura 78 – SIG1 de confiabilidade (ATC).

O SIG que representa o RNF de confiabilidade utilizou a decomposição de subtipos de RNFs baseada na norma ISO/IEC 9126 (ISO/IEC 9126, 2001). Foi utilizada a característica de qualidade confiabilidade para expressar o tipo do *softgoal* e a sub-característica tolerância à falhas para expressar o subtipo.

O método de decomposição de atributos foi utilizado para deixar explícito que o código de velocidade deve ter tratamento para tolerância a falhas.

A meta deste SIG foi definida pelo engenheiro de requisitos, uma vez que, os catálogos NFR *Framework* não contemplam o RNF de confiabilidade. A meta define a utilização de equipamento redundante a fim de que o sinal de código de velocidade existente nos circuitos de via seja detectado corretamente. A leitura executada pelo equipamento redundante deve ser utilizada, caso o equipamento principal falhe.

O SIG que representa o requisito não-funcional de acurácia associado à relação *Sistema de Portas-Controlle de Fechamento de Portas*, *Sistema de Portas- Controle de Abertura de Portas* e *Sistema de Portas-Controlle de Partida* é apresentado na Figura 79.

Figura 79 – SIG 4 de acurácia (ATC).

O SIG que representa o RNF de acurácia utiliza o método de decomposição de conservação. O *softgoal* de operacionalização é definido com base no catálogo de requisito não-funcional de acurácia do NFR *Framework* (CHUNG, et al.; 2000). A meta estabelecida para o SIG define que a verificação do estado das portas deverá passar por um rastreamento periódico a fim de que seu estado real seja confirmado.

O SIG que representa o RNF de *safety* associado ao caso de uso *Proteção Sobre Velocidade* é apresentado na Figura 80.

Figura 80 – SIG 8 de *safety* (ATC).

Este SIG foi utilizado para representar a decomposição de dois RNFs de *safety*. Isso ocorreu pois os RNFs (id 8 e id 9) são relacionados ao tratamento adequado dos freios de serviço e de emergência, ambos pertencentes ao caso de uso *Proteção de Sobre Velocidade*.

Foi utilizado o método de decomposição de casos de uso, definido na seção 4.6.1 deste trabalho. A propagação dos RNFs (id 8 e id 9) através dos casos de uso especializados de *Proteção de Velocidade Constante* e *Proteção de Velocidade em Transição* é representada neste SIG.

Os *softgoals* de operacionalização foram definidos pelo engenheiro de requisitos, devida à inexistência de catálogo específico para o RNF de *safety*.

A meta referente ao RNF (id 8) estabelece que as condições do acionamento do freio máximo de serviço devem ser controladas. Situação

similar ocorre com a meta referente ao RNF (id 9), que estabelece que as condições de acionamento do freio máximo de emergência devem ser controladas.

5.4.5 Sub-processo 5: Elaborar modelo composto

Este sub-processo é composto pelas atividades para identificar metas operacionais e restrições de projeto (5.1), definir pontos de composição (5.2), confirmar propagação (5.3), resolver conflitos (5.4) e atualizar o modelo do sistema de software (5.5).

A atividade para identificar metas operacionais e restrições de projeto (5.1) recebe como dados de entrada os SIGs e a tabela de RNF com aspectos. Nesta atividade deve ser elaborada a tabela de RNF com metas operacionais e restrições de projeto, que é a tabela de aspectos (Tabela 51) acrescida das colunas: id SIG, meta, origem da meta, meta operacional ou restrição de projeto e descrição. Esta tabela tem a finalidade de registrar a classificação do tipo de meta referente a cada SIG e a ação a ser tomada caso a meta seja classificada como meta operacional ou restrição de projeto.

A tabela de RNF com metas operacionais e restrições de projeto é apresentada na Tabela 52.

Tabela 52 – Tabela de RNF com metas operacionais e restrições de projeto (ATC).

Id	Id SIG	Meta	Origem da Meta	Meta Operacional ou Restrição de Projeto	Descrição
1	SIG1	Incluir equipamento de leitura dualizado.	Definido pelo engenheiro de requisitos	Restrição de Projeto	Incluir equipamento de leitura de código de velocidade dualizado a fim de garantir a leitura em caso de falha do equipamento principal.
2	SIG2	Fazer uma verificação cruzada utilizando o histórico de velocidades no trecho.	NFR	Meta Operacional	Novo caso de uso Verificação Cruzada
3	SIG3	Realizar dupla verificação da velocidade a fim de detectar com precisão velocidades abaixo do limite compatível com o perfil de via.	NFR	Restrição de Projeto	Incluir um canal de leitura de velocidade de detecção distinto a fim de garantir a leitura precisa de velocidades inferiores ao limite compatível com o perfil de via.

Id	Id SIG	Meta	Origem da Meta	Meta Operacional ou Restrição de Projeto	Descrição
4	SIG4	Utilizar verificações periódicas a fim de analisar o estado das portas.	NFR	Meta Operacional	Novo caso de uso Verificação do Estado das Portas
5	SIG5	Confirmar via canal idêntico a informação sobre o estado dos freios.	Definido pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Confirmação Estado dos Freios
6	SIG6	Garantir que os níveis de restrição referentes às diversas situações de falhas de freios estejam corretamente especificados.	Definido pelo engenheiro de requisitos	Restrição de Projeto	Garantir através do processo de verificação de documentação que todos os níveis de restrição relacionados às falhas de freios estejam especificados. A verificação deve ser feita por um especialista no sistema ATC.
7	SIG7	Confirmar portas fechadas, trem parado e detecção de código de velocidade compatível com o perfil da via.	Definido pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Confirmação da Condição de Partida
8	SIG8	Controlar condições de acionamento do máximo freio de serviço.	Definido pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Controle de Freio Máximo de Serviço
9	SIG8	Controlar condições de acionamento do freio de emergência.	Definido pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Controle de Freio Emergência
10	SIG9	Controlar condições da abertura de portas.	Definido pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Confirmação de Abertura de Portas

Nesta tabela podem ser observadas as restrições de projeto a serem impostas ao ATC e os novos casos de uso a serem adicionados no modelo de casos de uso.

O RNF (id 1) foi considerado restrição de projeto, pois a inclusão de um equipamento dualizado de leitura do sinal de via afeta a definição da arquitetura de hardware do ATC. O requisito não-funcional (id 2) foi considerado meta operacional, pois irá adicionar uma nova função ao ATC. Esta nova função define que seja realizada uma verificação cruzada entre as velocidades já praticadas no trecho e a velocidade real, a fim de obter maior precisão nas medições.

A atividade para definir pontos de composição (5.2) tem a finalidade de identificar os pontos no fluxo de seqüência de eventos dos casos de uso, onde as metas operacionais devem ser inseridas. A Tabela 53 apresenta a tabela de

RNF com composição, que é a tabela de RNF com metas operacionais e restrições de projeto (Tabela 52), acrescida das colunas: caso de uso, ponto de composição, regra de composição e condições da composição.

Tabela 53 – Tabela de RNF com composição (ATC).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição
2	Cálculo de Velocidade e Aceleração	2	sobreposição.depois	Nenhuma
4	Controle de Partida	4	sobreposição.depois	Nenhuma
	Controle de Abertura de Portas	3	sobreposição.depois	Nenhuma
	Controle de Fechamento de Portas	2	sobreposição.depois	Nenhuma
5	Controle de Sistema de Freios	4	sobreposição.depois	Nenhuma
7	Controle de Partida	4	sobreposição.depois	Nenhuma
8	Proteção de Velocidade Constante	3	empacotamento	Se a diferença entre a velocidade real do trem e a velocidade limite determinada pelo perfil de via e pelo nível de restrição for maior que o limite compatível com a via, o caso de uso Controle Máximo de Freio de Serviço deve ser executado.
	Proteção de Velocidade em Transição	3	empacotamento	Se a diferença entre a velocidade real do trem e a velocidade limite determinada pelo perfil de via e pelo nível de restrição for maior que o limite compatível com a via, o caso de uso Controle Máximo de Freio de Serviço deve ser executado.
9	Proteção de Velocidade Constante	4	empacotamento	Se a diferença entre a velocidade real do trem e a velocidade limite for maior que um valor limite superior, o caso de uso Controle de Freio de Emergência deve ser executado.
	Proteção de Velocidade em Transição	4	empacotamento	Se a diferença entre a velocidade real do trem e a velocidade limite for maior que um valor limite superior, o caso de uso Controle de Freio de Emergência deve ser executado.
10	Controle de Abertura de Portas	3	sobreposição.antes	Nenhuma

Somente são exibidas tabela de RNF com composição as linhas que representam metas operacionais. Os RNFs (id 4, id 8 e id 9) se repetem na tabela, o que mostra os impactos destes RNFs nos demais casos de uso do ATC.

Foram utilizadas para o ATC as regras de composição de sobreposição, que não requerem a definição da condição da composição e a regra de empacotamento, que precisa da definição da condição de composição.

Os RNFs (id 8 e id 9) utilizam a regras de empacotamento e, desta forma, o controle da execução dos novos casos de uso *Controle de Freio Máximo de Serviço* e *Controle de Freio de Emergência* depende do atendimento à regra de composição definida na coluna condições da composição da tabela de RNF com composição.

A atividade de confirmação das propagações (5.3) não detectou nenhuma necessidade de revisão para as propagações definidas inicialmente no ATC.

Na atividade para resolver conflitos (5.4) foi detectado um conflito no caso de uso *Controle de Partida*, conforme apresentado na Tabela 54.

Tabela 54 – Tabela de RNF com prioridades (ATC).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição	Prioridade da Composição
4	Controle de Partida	4	sobreposição.depois	Nenhuma	1
7	Controle de Partida	4	sobreposição.depois	Nenhuma	2

Os RNFs (id 4 e id 7) apresentam o mesmo ponto de composição com a mesma regra de composição. Nesta situação, o requisito não-funcional (id 4) é prioritário, pois o ATC deve detectar corretamente o estado das portas antes de habilitar a partida do trem.

Na última atividade para atualizar o modelo do sistema de software (5.5) um novo diagrama de casos de uso é gerado, com os novos elementos incluídos. A Figura 81 apresenta o diagrama de casos de uso atualizado.

Figura 81 – Diagrama de casos de uso atualizado (ATC).

Os casos de uso destacados são os novos casos de uso incluídos no diagrama. Os casos de uso *Controle de Freio Máximo de Serviço*, *Verificação do Estado das Portas* e *Controle de Freio de Emergência* possuem o estereótipo <<aspecto>> e são incluídos por diversos casos de uso do ATC. Os demais casos de uso, não são considerados aspectos do ATC, por isso não possuem o estereótipo.

As especificações dos novos casos de uso incluídos devem ser elaboradas. A especificação do caso de uso *Verificação Cruzada* é apresentada na Tabela 55.

Tabela 55 – Especificação do caso de uso Verificação Cruzada (ATC).

Nome do Caso de Uso	Verificação Cruzada
Descrição Breve	Este caso de uso tem o objetivo de confirmar a medição da velocidade real do trem.
Atores	Acionado pelo caso de uso Cálculo de Velocidade e Aceleração.
Evento Inicial	Cálculo da velocidade real.

Fluxo Básico	<ol style="list-style-type: none"> 1. O ATC analisa a velocidade calculada e compara com o histórico de velocidade já percorrido no trecho. 2. O ATC identifica que o perfil de velocidade é compatível com o perfil da via. 3. O caso de uso termina.
Fluxos Alternativos	2A. ATC identifica perfil de velocidade não compatível com o perfil da via 2A1. O ATC envia um sinal para o operador reportando a condição de velocidade.
Sub-fluxos	Nenhum.
Requisitos Especiais	2
Pré-condição	Nenhuma.
Pós-condição	Velocidade real analisada.
Pontos de Extensão	Nenhum.
Pontos de Inclusão	Nenhum.

O campo requisitos especiais deve ser preenchido com o identificador do requisito não-funcional que deu origem a nova especificação.

5.5 Sistema de Gerenciamento de Crise

O sistema de gerenciamento de crise (SGC) está baseado no documento de Kienzle, Guelfi e Mustafiz (2009). Este sistema de software foi selecionado por ser um sistema mais complexo que os demais e por contemplar diversos tipos de RNFs.

Desastres naturais (terremotos, tsunamis, furacões, enchentes), ataques terroristas ou sabotagens (explosões, sequestros), acidentes (emergência de contaminação, explosões de prédios, colisões de carros) e crises tecnológicas são exemplos de situações de emergência imprevisíveis, que podem levar a efeitos severos, caso não sejam tratados imediatamente. O impacto de uma crise pode variar de situações menores até situações que possam afetar vários segmentos da sociedade. O gerenciamento da crise envolve a identificação, a verificação e a manipulação da situação da crise. O SGC facilita o processo de orquestração da comunicação entre todas as partes interessadas na manipulação da crise, através da alocação e do gerenciamento de recursos, e fornece acesso a informações importantes da crise aos usuários autorizados.

5.5.1 Sub-processo 1: Obter os requisitos do sistema de software

Foi utilizado o trabalho de Kienzle, Guelfi e Mustafiz (2009), que contém a descrição do sistema, o diagrama de casos de uso e partes das especificações dos casos de uso. Após realizar a verificação do material existente, foi executada a atividade de elaborar os requisitos. Esta atividade consistiu do levantamento do glossário.

O glossário do SGC é apresentado na Tabela 56.

Tabela 56 – Glossário (SGC).

Termo	Descrição
Crise	Desastres naturais (terremotos, tsunamis, furacões, enchentes), ataques terroristas ou sabotagens (explosões, sequestros), acidentes (emergência de contaminação, explosões de prédios, colisões de carros) e crises tecnológicas.
Sistema de supervisão com vídeo	Sistema externo utilizado para monitorar estradas, ruas ou rotas congestionadas.
Missão	Conjunto de tarefas necessárias para solucionar um problema.

A Figura 82 apresenta o diagrama de casos de uso.

Figura 82 – Diagrama de casos de uso (SGC).

Os atores do SGC são os seguintes:

- Coordenador: é responsável pelo gerenciamento da crise e alocação de recursos.
- Vítima: é responsável por informar os dados da crise através de chamadas telefônicas ao centro de gerenciamento de crise.

- Companhia de Telefone: é uma entidade externa contactada para verificar a veracidade das informações transmitidas pela vítima.
- Sistema de Supervisão: é uma entidade externa que monitora o tráfego nas estradas e cidades com o uso de câmeras.
- Empregado do SGC: é um recurso humano interno que é qualificado e capaz de executar missões relacionadas à sua área de experiência. O empregado do SGC age como um facilitador quando existe necessidade de operar recursos locais, como dirigir caminhões ou ambulâncias.
- Sistema de Recurso Externo: fornece recursos especializados em executar missões relacionadas à sua área de experiência. O recurso externo age como um facilitador quando está responsável por operar recursos externos, como caminhões de polícia ou caminhões de bombeiro.
- Super Observador: é um especialista responsável por observar o local da crise e definir missões relacionadas à crise.
- Enfermeiro: é responsável pelos primeiros socorros às vítimas da crise.
- Hospital: é responsável por fornecer dados médicos sobre as vítimas da crise e receber pacientes.
- Recurso: é fornecedor de recurso abstrato que pode representar um recurso interno ou externo.

O caso de uso *Resolver Crise* tem o objetivo de alocar os empregados do SGC e recursos externos necessários para executar as missões relacionadas à crise.

O caso de uso *Capturar o Relatório da Crise* tem o objetivo de registrar os dados de uma crise com base em informações obtidas pelo coordenador através da vítima.

O caso de uso *Atribuir Recurso Interno* tem o objetivo de encontrar, contactar e atribuir uma missão a um empregado do SGC.

O caso de uso *Solicitar Recurso Externo* tem a finalidade de contactar entidades externas para solicitar um recurso externo, tais como: caminhão de bombeiro, carro de polícia ou serviço externo de ambulância.

O caso de uso *Executar Missão* tem a finalidade de executar missões a fim de ajudar na solução da crise.

O caso de uso *Executar Missão Super Observador* tem a finalidade de obter dados da situação no local da crise e solicitar missões apropriadas.

O caso de uso *Executar Missão Resgate* tem a finalidade alocar um enfermeiro para executar a missão que envolve o transporte da vítima até o hospital mais adequado.

O caso de uso *Autenticar Usuário* tem a finalidade de autenticar o acesso dos empregados ao SGC.

A Tabela 57 apresenta uma especificação do caso de uso *Resolver Crise*. As demais especificações do SGC encontram-se em Bombonatti (2010).

Tabela 57 – Especificação do caso de uso Resolver Crise (SGC).

Nome do Caso de Uso	Resolver Crise
Descrição Breve	Este caso de uso tem o objetivo de solicitar recursos adequados para executar suas missões.
Atores	Coordenador (ator primário) Recurso (ator secundário)
Evento Inicial	Chamada da vítima.
Fluxo Básico	<p>A vítima faz uma chamada ao centro de crises, que é respondida pelo coordenador.</p> <ol style="list-style-type: none"> 1. O coordenador captura o relatório da crise. O caso de uso <i>Capturar o Relatório da Crise</i> é incluído para captura do relatório. 2. O sistema recomenda ao coordenador as missões que serão executadas com base nas informações atuais sobre a crise e recursos. 3. O coordenador seleciona uma ou mais missões recomendadas pelo sistema. <p>Para cada missão, que pode ser disparado em paralelo:</p> <ol style="list-style-type: none"> 4. Para cada recurso interno requerido pela missão selecionada, o sistema atribui um recurso interno. O caso de uso <i>Atribuir Recurso Interno</i> é incluído. 5. Para cada recurso externo requerido pela missão selecionada, o sistema atribui um recurso externo. O caso de uso <i>Atribuir Recurso Externo</i> é incluído. 6. O recurso notifica o sistema da chegada no local da missão. 7. O recurso executa sua missão. O caso de uso <i>Executar Missão</i> é incluído. 8. O recurso notifica o sistema da sua saída do local da missão. 9. Em paralelo aos passos 6-8, o coordenador recebe atualizações sobre a situação da missão do sistema. 10. Em paralelo aos passos 6-8, o sistema informa aos recursos sobre mudanças relevantes nas informações da missão/crise. 11. O recurso submete o relatório final da missão para o sistema. 12. Em paralelo aos passos 4-8, o coordenador recebe novas informações sobre a crise do sistema. 13. O coordenador fecha o arquivo para a resolução da crise. 14. O caso de uso termina com sucesso.
Fluxos Alternativos	<ol style="list-style-type: none"> 1A. O coordenador não está logado no sistema. 1A1. O coordenador se autentica no sistema. O caso de uso <i>Autenticar Usuário</i> é incluído. 1A2. O caso de uso continua no passo 1. 4A. O recurso interno não está disponível para ser alocado no passo 4. 4A1. O sistema solicita um recurso externo (o caso de uso continua em paralelo com o passo 5). 5A. O recurso externo não está disponível depois do passo 5.

	<p>5A1. O caso de uso continua em paralelo com o passo 2.</p> <p>6A. O recurso não consegue contatar o sistema. 6A1. O Super Observador notifica o sistema de que o recurso chegou no local da missão.</p> <p>6B. Embora o recurso deva estar no local da missão, o recurso ainda não notificou o sistema. 6B1. O sistema solicita ao recurso para prover informações atualizadas sobre sua localização. 6B2. O caso e uso continua no passo 6.</p> <p>7A. Uma ou mais missões adicionais são requeridas no passo 6. 7A1. O caso de uso continua em paralelo com o passo 2.</p> <p>7B. A missão falha. 7B. O caso de uso continua no passo 2.</p> <p>8A. O recurso não consegue contatar o sistema. 8A1. O Super Observador notifica o sistema de que o recurso está deixando o local da missão.</p> <p>8B. Embora a missão deva estar completa, o recurso não deixou o local da missão. 8B1. O sistema solicita ao recurso para prover a razão do seu atraso. 8B2. O caso de uso continua no passo 7.</p> <p>9A. Mudanças na missão são necessárias. 9A1. O caso de uso continua em paralelo ao passo 2.</p> <p>11A. O recurso não submete o relatório da missão. 11A1. O caso de uso termina sem o relatório da missão,</p> <p>12A. Mudanças na missão são necessárias. 12A1. O caso de uso continua em paralelo com o passo 2.</p>
Sub-fluxos	Nenhum.
Requisitos Especiais	Nenhum.
Pré-condições	Nenhuma.
Pós-condições	Missão finalizada.
Pontos de Extensão	Nenhum.
Pontos de Inclusão	<p>I1. O caso de uso <i>Capturar o Relatório da Crise</i> é incluído no passo 1 do fluxo básico.</p> <p>I2. O caso de uso <i>Atribuir Recursos Internos</i> é incluído no passo 4 do fluxo básico.</p> <p>I3. O caso de uso <i>Atribuir Recursos Externos</i> é incluído no passo 5 do fluxo básico.</p> <p>I4. O caso de uso <i>Executar Missão</i> é incluído no passo 7 do fluxo básico.</p> <p>I5. O caso de uso <i>Autenticar Usuário</i> é incluído no passo 1A1 do fluxo alternativo.</p>

A especificação de requisitos não-funcionais do SGC é descrita na Tabela 58.

Tabela 58 – Especificação de requisitos não-funcionais (SGC).

Id	Requisito	Prioridade
1	O sistema deve estar em operação 24 horas por dia, todos os dias, sem parar, durante o ano todo exceto por um tempo máximo de 2 horas a cada 30 dias para manutenção.	2
2	O sistema deve recuperar uma falha em no máximo 30 segundos.	2
3	A manutenção pode ser postergada ou interrompida se uma crise é eminente, sem afetar as capacidades do sistema.	2
4	O sistema não deve exceder a taxa de falha de 0,001%.	2
5	O sistema deve prover suporte para armazenar e atualizar as seguintes informações da crise resolvida e em andamento: tipo de crise, localização da crise, relatório da vítima, localização da vítima, dados da vítima, tempo reportado, duração da resolução, recursos empregados, vítimas civis, vítimas do gerenciamento de crise, estratégias utilizadas, missões utilizadas, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, log de comunicação, log de decisões, log de problemas encontrados.	3
6	O sistema deve prover suporte para acessar as seguintes informações da crise resolvida e em andamento: tipo de crise, localização da crise, relatório da vítima, localização da vítima, dados da vítima, tempo reportado, duração da resolução, recursos empregados, vítimas civis, vítimas do gerenciamento de crise, estratégias utilizadas, missões utilizadas, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, log de comunicação, log de decisões, log de problemas encontrados.	3
7	O sistema deve fornecer suporte para armazenar e atualizar as seguintes informações sobre os recursos (internos e externos) disponíveis e utilizados: tipo de recurso (humano ou equipamento), capacidade, equipe de resgate, localização, tempo estimado para chegada no local da crise.	3
8	O sistema deve fornecer suporte para acessar as seguintes informações sobre os recursos (internos e externos) disponíveis e utilizados: tipo de recurso (humano ou equipamento), capacidade, equipe de resgate, localização, tempo estimado para chegada no local da crise.	3
9	O sistema deve prover suporte para armazenar e atualizar as seguintes informações sobre as estratégias de resolução da crise: tipo de crise, guia passo-a-passo para resolver a crise, configuração das missões requeridas, vínculo com estratégias alternativas, aplicações em crises prévias, taxa de sucesso.	3
10	O sistema deve prover suporte para acessar as seguintes informações sobre as estratégias de resolução da crise: tipo de crise, guia passo-a-passo para resolver a crise, configuração das missões requeridas, vínculo com estratégias alternativas, aplicações em crises prévias, taxa de sucesso.	3
11	O centro de controle deve receber e atualizar as seguintes informações durante o andamento da crise, em intervalos que não excedam 30 segundos: recursos empregados, vítimas civis, vítimas do gerenciamento de crise, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, tempo estimado para chegada no local da crise.	3
12	A atraso na comunicação da informação entre o centro de controle e a equipe de resgate, bem como entre membros do tipo de resgate, não pode exceder 500 milissegundos.	3
13	O sistema deve ser capaz de recuperar qualquer informação armazenada com o máximo de atraso de 500 milissegundos.	3
14	O sistema deve definir políticas de acesso para várias classes de usuários. A política de acesso deve descrever os componentes e as informações que cada classe deve adicionar, acessar e atualizar.	4
15	Se um usuário permanecer inativo por mais de 30 minutos, o sistema deve solicitar a sua reautenticação.	4
16	O sistema deve autenticar usuários com base nas políticas de acesso quando estes acessarem pela primeira vez qualquer componente ou informação.	4

Id	Requisito	Prioridade
17	Todas as comunicações do sistema devem usar canais seguros em conformidade com o padrão de criptografia AES-128.	4
18	Recursos de resgate devem ser capazes de acessar a informação em movimento.	5
19	O sistema deve prover informação sensível à localidade para os recursos de resgate.	5
20	Recursos de resgate devem comunicar sua localização ao centro de controle.	5
21	O sistema deve acessar mapas detalhados, dados de terreno e condições do tempo para o local da crise e as rotas que levam até o local.	5
22	O sistema deve registrar as seguintes informações estatísticas, nas crises em andamento e resolvidas: taxa de progressão; tempo médio de resposta das equipes de resgate, tempo de resposta individual de cada equipe de resgate, taxa de sucesso de cada equipe de resgate, taxa de vítimas, taxa de sucesso das missões.	6
23	O sistema deve prover ferramentas de análise estatística para analisar os dados da crise em múltiplas crises.	6
24	O sistema deve gerenciar ao menos 1000 chamadas de vítimas ao mesmo tempo.	7
25	O sistema deve gerenciar comunicação, coordenação e acesso à informação a pelo menos 20000 recursos de resgate em atuação ao mesmo tempo.	7
26	O sistema deve gerenciar pelo menos 100 crises ao mesmo tempo.	7
27	O sistema deve gerenciar pelo menos 200 missões por crise ao mesmo tempo.	7
28	O sistema deve monitorar emissões do local da crise para determinar as distâncias de operação segura para os recursos de resgate.	1
29	O sistema deve monitorar as condições de terreno e tempo no local da crise para garantir a operação segura e afastamento dos recursos de resgate, e remoção de vítimas civis.	1
30	O sistema deve determinar o perímetro do local da crise para garantir a segurança dos civis e remoção de vítima a uma distância segura.	1
31	O sistema deve monitorar atividades criminais para garantir a segurança dos recursos de resgate, civis e vítimas.	1
32	A segurança do pessoal do resgate deve ter prioridade para o sistema.	1
33	O sistema deve recomendar estratégias alternativas para lidar com a crise, bem como com as condições de mudança da crise (exemplo: condições do tempo, condições do terreno, atividade civil ou criminal).	8
34	O sistema deve recomendar ou listar recursos alternativos em caso de indisponibilidade ou falta de recursos apropriados.	8
35	O sistema deve ser capaz de usar canais de comunicação alternativos em caso de indisponibilidade ou falta dos canais existentes.	8
36	O sistema deve ser capaz de manter comunicação efetiva em áreas acidentadas ou com grande ruído no local da crise.	8
37	O sistema deve ter acesso aos dados de mapa, terreno e temperatura com 99% de acurácia.	9
38	O sistema deve registrar dados até o receptor sem modificações.	9
39	A comunicação entre o sistema e os recursos de resgate deve ter o fator de deterioração máxima de 0,0001 por 1000 quilômetros.	9

Os RNFs, bem como suas respectivas prioridades, faziam parte da descrição dada em Kienzle; Guelfi; Mustafiz (2009). O conjunto de requisitos

que recebeu a maior prioridade foi aquele relacionado a *safety*, pois trata diretamente da vida da equipe e das vítimas envolvidas na crise.

5.5.2 Sub-processo 2: Relacionar interesses funcionais e não-funcionais

O sub-processo 2, para relacionar interesses funcionais e não-funcionais, é composto das atividades para identificar interesses funcionais e não-funcionais (2.1) e integrar interesses funcionais e não-funcionais (2.2).

Na atividade para identificar os interesses funcionais e não-funcionais (2.1) foram utilizados o modelo de casos de uso e a especificação de requisitos não-funcionais, recebidos como dados de entrada para esta atividade.

Como resultado da atividade de identificação de interesses funcionais e não-funcionais (2.1) foram gerados os *softgoals* (Figura 83), a lista de interesses (Tabela 59) e a tabela de RNF (Tabela 60).

Figura 83 – *Softgoals* (SGC).

Os *softgoals* representam os interesses não-funcionais do SGC, que por sua vez podem representar um ou mais RNFs do mesmo tipo. A lista de interesses do SGC é apresentada na Tabela 59.

Tabela 59 – Tabela de interesses (SGC).

Interesse	Natureza
Resolver Crise	Funcional
Capturar o Relatório da Crise	
Autenticar Usuário	
Atribuir Recurso Interno	
Solicitar Recurso Externo	
Executar Missão	
Executar Missão Resgate	
Executar Missão Super Observador	
Adaptabilidade	Não-Funcional
Disponibilidade	
Confiabilidade	
Espaço	
Tempo	

Interesse	Natureza
Security	
Observabilidade	
Acurácia	
Safety	
Desempenho	
Mobilidade	

Cada um dos interesses foi definido de acordo com uma origem. Esta origem é identificada na Tabela 60. A identificação desta informação é importante para que decomposições sejam feitas através dos SIGs.

Tabela 60 – Tabela de RNF (SGC).

Id	Requisito	Prioridade	Interesse	Origem do Interesse
1	O sistema deve estar em operação 24 horas por dia, todos os dias, sem parar, durante o ano todo exceto por um tempo máximo de 2 horas a cada 30 dias para manutenção.	2	Disponibilidade	ISO/IEC 9126
2	O sistema deve recuperar uma falha em no máximo 30 segundos.	2	Disponibilidade	ISO/IEC 9126
3	A manutenção pode ser postergada ou interrompida se uma crise é eminente, sem afetar as capacidades do sistema.	2	Disponibilidade	ISO/IEC 9126
4	O sistema não deve exceder a taxa de falha de 0,001%.	2	Confiabilidade	ISO/IEC 9126
5	O sistema deve prover suporte para armazenar e atualizar as seguintes informações da crise resolvida e em andamento: tipo de crise, localização da crise, relatório da vítima, localização da vítima, dados da vítima, tempo reportado, duração da resolução, recursos empregados, vítimas civis, vítimas do gerenciamento de crise, estratégias utilizadas, missões utilizadas, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, log de comunicação, log de decisões, log de problemas encontrados.	3	Espaço	NFR
6	O sistema deve prover suporte para acessar as seguintes informações da crise resolvida e em andamento: tipo de crise, localização da crise, relatório da vítima, localização da vítima, dados da vítima, tempo reportado, duração da resolução, recursos empregados, vítimas civis, vítimas do gerenciamento de crise, estratégias utilizadas, missões utilizadas, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, log de comunicação, log de decisões, log de problemas encontrados.	3	Espaço	NFR
7	O sistema deve fornecer suporte para armazenar e atualizar as seguintes informações sobre os recursos (internos e externos) disponíveis e utilizados: tipo de recurso (humano ou equipamento), capacidade, equipe de resgate, localização, tempo estimado para chegada no local da crise.	3	Espaço	NFR

Id	Requisito	Prioridade	Interesse	Origem do Interesse
8	O sistema deve fornecer suporte para acessar as seguintes informações sobre os recursos (internos e externos) disponíveis e utilizados: tipo de recurso (humano ou equipamento), capacidade, equipe de resgate, localização, tempo estimado para chegada no local da crise.	3	Espaço	NFR
9	O sistema deve prover suporte para armazenar e atualizar as seguintes informações sobre as estratégias de resolução da crise: tipo de crise, guia passo-a-passo para resolver a crise, configuração das missões requeridas, vínculo com estratégias alternativas, aplicações em crises prévias, taxa de sucesso.	3	Espaço	NFR
10	O sistema deve prover suporte para acessar as seguintes informações sobre as estratégias de resolução da crise: tipo de crise, guia passo-a-passo para resolver a crise, configuração das missões requeridas, vínculo com estratégias alternativas, aplicações em crises prévias, taxa de sucesso.	3	Espaço	NFR
11	O centro de controle deve receber e atualizar as seguintes informações durante o andamento da crise, em intervalos que não excedam 30 segundos: recursos empregados, vítimas civis, vítimas do gerenciamento de crise, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, tempo estimado para chegada no local da crise.	3	Tempo	NFR
12	A atraso na comunicação da informação entre o centro de controle e a equipe de resgate, bem como entre membros do tipo de resgate, não pode exceder 500 milissegundos.	3	Tempo	NFR
13	O sistema deve ser capaz de recuperar qualquer informação armazenada com o máximo de atraso de 500 milissegundos.	3	Tempo	NFR
14	O sistema deve definir políticas de acesso para várias classes de usuários. A política de acesso deve descrever os componentes e as informações que cada classe deve adicionar, acessar e atualizar.	4	Security	NFR
15	Se um usuário permanecer inativo por mais de 30 minutos, o sistema deve solicitar a sua reautenticação.	4	Security	NFR
16	O sistema deve autenticar usuários com base nas políticas de acesso quando estes acessarem pela primeira vez qualquer componente ou informação.	4	Security	NFR
17	Todas as comunicações do sistema devem usar canais seguros em conformidade com o padrão de criptografia AES-128.	4	Security	NFR
18	Recursos de resgate devem ser capazes de acessar a informação em movimento.	5	Mobilidade	Lista Chung
19	O sistema deve prover informação sensível à localidade para os recursos de resgate.	5	Mobilidade	Lista Chung
20	Recursos de resgate devem comunicar sua localização ao centro de controle.	5	Mobilidade	Lista Chung

Id	Requisito	Prioridade	Interesse	Origem do Interesse
21	O sistema deve acessar mapas detalhados, dados de terreno e condições do tempo para o local da crise e as rotas que levam até o local.	5	Mobilidade	Lista Chung
22	O sistema deve registrar as seguintes informações estatísticas, nas crises em andamento e resolvidas: taxa de progressão; tempo médio de resposta das equipes de resgate, tempo de resposta individual de cada equipe de resgate, taxa de sucesso de cada equipe de resgate, taxa de vítimas, taxa de sucesso das missões.	6	Observabilidade	Lista Chung
23	O sistema deve prover ferramentas de análise estatística para analisar os dados da crise em múltiplas crises.	6	Observabilidade	Lista Chung
24	O sistema deve gerenciar ao menos 1000 chamadas de vítimas ao mesmo tempo.	7	Desempenho	NFR
25	O sistema deve gerenciar comunicação, coordenação e acesso à informação a pelo menos 20000 recursos de resgate em atuação ao mesmo tempo.	7	Desempenho	NFR
26	O sistema deve gerenciar pelo menos 100 crises ao mesmo tempo.	7	Desempenho	NFR
27	O sistema deve gerenciar pelo menos 200 missões por crise ao mesmo tempo.	7	Desempenho	NFR
28	O sistema deve monitorar emissões do local da crise para determinar as distâncias de operação segura para os recursos de resgate.	1	Safety	ISO/IEC 9126
29	O sistema deve monitorar as condições de terreno e tempo no local da crise para garantir a operação segura e afastamento dos recursos de resgate, e remoção de vítimas civis.	1	Safety	ISO/IEC 9126
30	O sistema deve determinar o perímetro do local da crise para garantir a segurança dos civis e remoção de vítima a uma distância segura.	1	Safety	ISO/IEC 9126
31	O sistema deve monitorar atividades criminais para garantir a segurança dos recursos de resgate, civis e vítimas.	1	Safety	ISO/IEC 9126
32	A segurança do pessoal do resgate deve ter prioridade para o sistema.	1	Safety	ISO/IEC 9126
33	O sistema deve recomendar estratégias alternativas para lidar com a crise, bem como com as condições de mudança da crise (exemplo: condições do tempo, condições do terreno, atividade civil ou criminal).	8	Adaptabilidade	ISO/IEC 9126
34	O sistema deve recomendar ou listar recursos alternativos em caso de indisponibilidade ou falta de recursos apropriados.	8	Adaptabilidade	ISO/IEC 9126
35	O sistema deve ser capaz de usar canais de comunicação alternativos em caso de indisponibilidade ou falta dos canais existentes.	8	Adaptabilidade	ISO/IEC 9126
36	O sistema deve ser capaz de manter comunicação efetiva em áreas acidentadas ou com grande ruído no local da crise.	8	Adaptabilidade	ISO/IEC 9126
37	O sistema deve ter acesso aos dados de mapa, terreno e temperatura com 99% de acurácia.	9	Acurácia	NFR

Id	Requisito	Prioridade	Interesse	Origem do Interesse
38	O sistema deve registrar dados até o receptor sem modificações.	9	Acurácia	NFR
39	A comunicação entre o sistema e os recursos de resgate deve ter o fator de deterioração máxima de 0,0001 por 1000 quilômetros.	9	Acurácia	NFR

Uma grande diversidade de interesses faz parte do SGC. Interesses com origens provenientes do NFR *Framework*, norma ISO/IEC 9126 (ISO/IEC 9126, 2001) e da Lista de Chung et al. (2000) são identificados para o SGC.

Nesta aplicação optou-se por utilizar os subtipos do requisito não-funcional de desempenho, tempo e espaço, a fim de proporcionar maior clareza na classificação destes interesses.

A atividade para integrar interesses funcionais e não-funcionais (2.2) utilizou as informações geradas na atividade 2.1 para que o diagrama integrado fosse elaborado. Este diagrama é apresentado na Figura 84.

Figura 84 – Diagrama integrado (SGC).

No diagrama integrado do SGC, diversos *softgoals* estão associados ao limite do sistema: adaptabilidade, disponibilidade, confidencialidade, espaço, tempo, *security*, observabilidade, acurácia e *safety*.

Diversos *softgoals* também estão associados ao caso de uso *Resolver Crise*: desempenho, adaptabilidade, acurácia, mobilidade. Isso se deve ao fato de que este caso de uso engloba grande parte das funções relacionadas ao gerenciamento de uma crise.

5.5.3 Sub-processo 3: Identificar aspectos

O sub-processo 3, para identificar aspectos, é composto das atividades para identificar propagações (3.1) e identificar aspectos (3.2).

Na atividade para identificar propagações (3.1), o diagrama integrado e os SIGs iniciais são recebidos como dados de entrada e as propagações relacionadas a cada um dos requisitos não-funcionais, que compõem um interesse não-funcional, devem ser analisadas.

As heurísticas para analisar as propagações de um requisito não-funcional nos elementos do diagrama de casos de uso estão definidas na seção 4.5 deste trabalho.

O resultado desta análise pode ser observado na Tabela 61 que é a Tabela 60, acrescida de colunas ponto de associação e propagação.

Tabela 61 – Tabela de RNF com propagação (SGC).

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
1	O sistema deve estar em operação 24 horas por dia, todos os dias, sem parar, durante o ano todo exceto por um tempo máximo de 2 horas a cada 30 dias para manutenção.	2	Disponibilidade	ISO/IEC 9126	Disponibilidade [SGC]	Sistema
2	O sistema deve recuperar uma falha em no máximo 30 segundos.	2	Disponibilidade	ISO/IEC 9126	Disponibilidade [SGC]	Sistema
3	A manutenção pode ser postergada ou interrompida se uma crise é eminente, sem afetar as capacidades do sistema.	2	Disponibilidade	ISO/IEC 9126	Disponibilidade [SGC]	Sistema
4	O sistema não deve exceder a taxa de falha de 0,001%.	2	Confiabilidade	ISO/IEC 9126	Confiabilidade [SGC]	Sistema

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
5	O sistema deve prover suporte para armazenar e atualizar as seguintes informações da crise resolvida e em andamento: tipo de crise, localização da crise, relatório da vítima, localização da vítima, dados da vítima, tempo reportado, duração da resolução, recursos empregados, vítimas civis, vítimas do gerenciamento de crise, estratégias utilizadas, missões utilizadas, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, log de comunicação, log de decisões, log de problemas encontrados.	3	Espaço	NFR	Espaço [SGC]	Sistema
6	O sistema deve prover suporte para acessar as seguintes informações da crise resolvida e em andamento: tipo de crise, localização da crise, relatório da vítima, localização da vítima, dados da vítima, tempo reportado, duração da resolução, recursos empregados, vítimas civis, vítimas do gerenciamento de crise, estratégias utilizadas, missões utilizadas, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, log de comunicação, log de decisões, log de problemas encontrados.	3	Espaço	NFR	Espaço [SGC]	Sistema
7	O sistema deve fornecer suporte para armazenar e atualizar as seguintes informações sobre os recursos (internos e externos) disponíveis e utilizados: tipo de recurso (humano ou equipamento), capacidade, equipe de resgate, localização, tempo estimado para chegada no local da crise.	3	Espaço	NFR	Espaço [SGC]	Sistema
8	O sistema deve fornecer suporte para acessar as seguintes informações sobre os recursos (internos e externos) disponíveis e utilizados: tipo de recurso (humano ou equipamento), capacidade, equipe de resgate, localização, tempo estimado para chegada no local da crise.	3	Espaço	NFR	Espaço [SGC]	Sistema

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
9	O sistema deve prover suporte para armazenar e atualizar as seguintes informações sobre as estratégias de resolução da crise: tipo de crise, guia passo-a-passo para resolver a crise, configuração das missões requeridas, vínculo com estratégias alternativas, aplicações em crises prévias, taxa de sucesso.	3	Espaço	NFR	Espaço [SGC]	Sistema
10	O sistema deve prover suporte para acessar as seguintes informações sobre as estratégias de resolução da crise: tipo de crise, guia passo-a-passo para resolver a crise, configuração das missões requeridas, vínculo com estratégias alternativas, aplicações em crises prévias, taxa de sucesso.	3	Espaço	NFR	Espaço [SGC]	Sistema
11	O centro de controle deve receber e atualizar as seguintes informações durante o andamento da crise, em intervalos que não excedam 30 segundos: recursos empregados, vítimas civis, vítimas do gerenciamento de crise, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, tempo estimado para chegada no local da crise.	3	Tempo	NFR	Tempo [SGC]	Sistema
12	A atraso na comunicação da informação entre o centro de controle e a equipe de resgate, bem como entre membros do tipo de resgate, não pode exceder 500 milissegundos.	3	Tempo	NFR	Tempo [SGC]	Sistema
13	O sistema deve ser capaz de recuperar qualquer informação armazenada com o máximo de atraso de 500 milissegundos.	3	Tempo	NFR	Tempo [SGC]	Sistema

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
14	O sistema deve definir políticas de acesso para várias classes de usuários. A política de acesso deve descrever os componentes e as informações que cada classe deve adicionar, acessar e atualizar.	4	Security	NFR	Security [Coordenador-Resolver Crise, Coordenador-Capturar Relatório da Crise, Companhia de Telefone-Capturar Relatório da Crise, Sistema de Supervisão-Capturar Relatório da Crise, Empregado do SGC-Atribuir Recurso Interno, Sistema de Recurso Externo-Solicitar Recurso Externo, Recurso-Executar Missão, Super Observador-Executar Missão Super Observador, Hospital-Executar Missão Resgate, Enfermeiro-Executar Missão Resgate]	Nenhuma
15	Se um usuário permanecer inativo por mais de 30 minutos, o sistema deve solicitar a sua reautenticação.	4	Security	NFR	Security [Autenticar Usuário]	Nenhuma
16	O sistema deve autenticar usuários com base nas políticas de acesso quando estes acessarem pela primeira vez qualquer componente ou informação.	4	Security	NFR	Security [Autenticar Usuário]	Nenhuma
17	Todas as comunicações do sistema devem usar canais seguros em conformidade com o padrão de criptografia AES-128.	4	Security	NFR	Security [SGC]	Sistema
18	Recursos de resgate devem ser capazes de acessar a informação em movimento.	5	Mobilidade	Lista Chung	Mobilidade [Recurso-Executar Missão]	Nenhuma
19	O sistema deve prover informação sensível à localidade para os recursos de resgate.	5	Mobilidade	Lista Chung	Mobilidade [Resolver Crise]	Nenhuma

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
20	Recursos de resgate devem comunicar sua localização ao centro de controle.	5	Mobilidade	Lista Chung	Mobilidade [Recurso-Executar Missão]	Hospital-Executar Missão Resgate, Enfermeiro-Executar Missão Resgate, Super Observador-Executar Missão Super Observador
21	O sistema deve acessar mapas detalhados, dados de terreno e condições do tempo para o local da crise e as rotas que levam até o local.	5	Mobilidade	Lista Chung	Mobilidade [Resolver Crise]	Nenhuma
22	O sistema deve registrar as seguintes informações estatísticas, nas crises em andamento e resolvidas: taxa de progressão; tempo médio de resposta das equipes de resgate, tempo de resposta individual de cada equipe de resgate, taxa de sucesso de cada equipe de resgate, taxa de vítimas, taxa de sucesso das missões.	6	Observabilidade	Lista Chung	Observabilidade [SGC]	Sistema
23	O sistema deve prover ferramentas de análise estatística para analisar os dados da crise em múltiplas crises.	6	Observabilidade	Lista Chung	Observabilidade [SGC]	Sistema
24	O sistema deve gerenciar ao menos 1000 chamadas de vítimas ao mesmo tempo.	7	Desempenho	NFR	Desempenho [Capturar o Relatório da Crise]	Nenhuma
25	O sistema deve gerenciar comunicação, coordenação e acesso à informação a pelo menos 20000 recursos de resgate em atuação ao mesmo tempo.	7	Desempenho	NFR	Desempenho [Empregado do SGC-Atribuir Recurso Interno, Sistema de Recurso Externo-Solicitar Recurso Externo, Super Observador-Executar Missão Super Observador, Recurso-Executar Missão, Hospital-Executar Missão Resgate, Enfermeiro-Executar Missão Resgate]	Nenhuma

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
26	O sistema deve gerenciar pelo menos 100 crises ao mesmo tempo.	7	Desempenho	NFR	Desempenho [Resolver Crise]	Capturar o Relatório da Crise, Atribuir Recurso Interno, Solicitar Recurso Externo, Executar Missão, Executar Missão Resgate, Executar Missão Super Observador
27	O sistema deve gerenciar pelo menos 200 missões por crise ao mesmo tempo.	7	Desempenho	NFR	Desempenho [Executar Missão]	Executar Missão Resgate, Executar Missão Super Observador
28	O sistema deve monitorar emissões do local da crise para determinar as distâncias de operação segura para os recursos de resgate.	1	<i>Safety</i>	ISO/IEC 9126	Safety [Executar Missão Super Observador]	Nenhuma
29	O sistema deve monitorar as condições de terreno e tempo no local da crise para garantir a operação segura e afastamento dos recursos de resgate, e remoção de vítimas civis.	1	<i>Safety</i>	ISO/IEC 9126	Safety [Executar Missão Super Observador]	Nenhuma
30	O sistema deve determinar o perímetro do local da crise para garantir a segurança dos civis e remoção de vítima a uma distância segura.	1	<i>Safety</i>	ISO/IEC 9126	Safety [Executar Missão Super Observador]	Nenhuma
31	O sistema deve monitorar atividades criminais para garantir a segurança dos recursos de resgate, civis e vítimas.	1	<i>Safety</i>	ISO/IEC 9126	Safety [Executar Missão Super Observador]	Nenhuma
32	A segurança do pessoal do resgate deve ter prioridade para o sistema.	1	<i>Safety</i>	ISO/IEC 9126	Safety [Executar Missão Super Observador]	Nenhuma
33	O sistema deve recomendar estratégias alternativas para lidar com a crise, bem como com as condições de mudança da crise (exemplo: condições do tempo, condições do terreno, atividade civil ou criminal).	8	Adaptabilidade	ISO/IEC 9126	Adaptabilidade [Resolver Crise]	Executar Missão, Executar Missão Super Observador
34	O sistema deve recomendar ou listar recursos alternativos em caso de indisponibilidade ou falta de recursos apropriados.	8	Adaptabilidade	ISO/IEC 9126	Adaptabilidade [Resolver Crise]	Executar Missão, Executar Missão Super Observador

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação
35	O sistema deve ser capaz de usar canais de comunicação alternativos em caso de indisponibilidade ou falta dos canais existentes.	8	Adaptabilidade	ISO/IEC 9126	Adaptabilidade [SGC]	Sistema
36	O sistema deve ser capaz de manter comunicação efetiva em áreas acidentadas ou com grande ruído no local da crise.	8	Adaptabilidade	ISO/IEC 9126	Adaptabilidade [SGC]	Sistema
37	O sistema deve ter acesso aos dados de mapa, terreno e temperatura com 99% de acurácia.	9	Acurácia	NFR	Acurácia [Resolver Crise]	Nenhuma
38	O sistema deve registrar dados até o receptor sem modificações.	9	Acurácia	NFR	Acurácia [SGC]	Sistema
39	A comunicação entre o sistema e os recursos de resgate deve ter o fator de deterioração máxima de 0,0001 por 1000 quilômetros.	9	Acurácia	NFR	Acurácia [SGC]	Sistema

Diversas propagações de RNFs ocorreram através do limite do sistema do SGC. Nenhuma propagação ocorreu nos RNFs que compõem o interesse de *safety*; entretanto, propagações foram observadas nos RNFs de mobilidade, desempenho e adaptabilidade.

Os RNFs de *safety* foram associados somente ao caso de uso *Executar Missão Super Observador*, pois o Super Observador é um especialista que poderá monitorar e controlar as variáveis relacionadas às emissões, distância segura, perímetro e atividades criminais do local da crise a fim de propor soluções adequadas para cada uma das situações.

Com relação aos RNFs de desempenho, o gerenciamento simultâneo de diversas crises e missões requer a participação de diversos casos de uso, justificando desta forma, as propagações que ocorreram através dos elementos do diagrama de casos de uso.

Os requisitos não-funcionais de adaptabilidade, por sua vez, apresentam a necessidade de receber, do SGC, sugestões sobre recursos e estratégias alternativas, para lidar com a crise. Esta necessidade fez com que estes requisitos fossem propagados através de casos de uso como: *Resolver Crise* e *Executar Missão Super Observador*, que necessitam de recursos e estratégias alternativas, quando as estratégias e recursos originalmente definidos, estão indisponíveis.

A atividade para identificar aspectos (3.2) utiliza as heurísticas definidas na seção 4.5 deste trabalho. A Tabela 62 apresenta tabela de RNF com aspectos, que é a tabela de propagações acrescida da coluna aspecto.

Tabela 62 – Tabela de RNF com aspectos (SGC).

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação	Aspecto
1	O sistema deve estar em operação 24 horas por dia, todos os dias, sem parar, durante o ano todo exceto por um tempo máximo de 2 horas a cada 30 dias para manutenção.	2	Disponibilidade	ISO/IEC 9126	Disponibilidade [SGC]	Sistema	Sim
2	O sistema deve recuperar uma falha em no máximo 30 segundos.	2	Disponibilidade	ISO/IEC 9126	Disponibilidade [SGC]	Sistema	Sim
3	A manutenção pode ser postergada ou interrompida se uma crise é eminente, sem afetar as capacidades do sistema.	2	Disponibilidade	ISO/IEC 9126	Disponibilidade [SGC]	Sistema	Sim
4	O sistema não deve exceder a taxa de falha de 0,001%.	2	Confiabilidade	ISO/IEC 9126	Confiabilidade [SGC]	Sistema	Sim
5	O sistema deve prover suporte para armazenar e atualizar as seguintes informações da crise resolvida e em andamento: tipo de crise, localização da crise, relatório da vítima, localização da vítima, dados da vítima, tempo reportado, duração da resolução, recursos empregados, vítimas civis, vítimas do gerenciamento de crise, estratégias utilizadas, missões utilizadas, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, log de comunicação, log de decisões, log de problemas encontrados.	3	Espaço	NFR	Espaço [SGC]	Sistema	Sim
6	O sistema deve prover suporte para acessar as seguintes informações da crise resolvida e em andamento: tipo de crise, localização da crise, relatório da vítima, localização da vítima, dados da vítima, tempo reportado, duração da resolução, recursos empregados, vítimas civis, vítimas do gerenciamento de crise, estratégias utilizadas,	3	Espaço	NFR	Espaço [SGC]	Sistema	Sim

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação	Aspecto
	missões utilizadas, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, log de comunicação, log de decisões, log de problemas encontrados.						
7	O sistema deve fornecer suporte para armazenar e atualizar as seguintes informações sobre os recursos (internos e externos) disponíveis e utilizados: tipo de recurso (humano ou equipamento), capacidade, equipe de resgate, localização, tempo estimado para chegada no local da crise.	3	Espaço	NFR	Espaço [SGC]	Sistema	Sim
8	O sistema deve fornecer suporte para acessar as seguintes informações sobre os recursos (internos e externos) disponíveis e utilizados: tipo de recurso (humano ou equipamento), capacidade, equipe de resgate, localização, tempo estimado para chegada no local da crise.	3	Espaço	NFR	Espaço [SGC]	Sistema	Sim
9	O sistema deve prover suporte para armazenar e atualizar as seguintes informações sobre as estratégias de resolução da crise: tipo de crise, guia passo-a-passo para resolver a crise, configuração das missões requeridas, vínculo com estratégias alternativas, aplicações em crises prévias, taxa de sucesso.	3	Espaço	NFR	Espaço [SGC]	Sistema	Sim
10	O sistema deve prover suporte para acessar as seguintes informações sobre as estratégias de resolução da crise: tipo de crise, guia passo-a-passo para resolver a crise, configuração das missões requeridas, vínculo com estratégias alternativas, aplicações em crises prévias, taxa de sucesso.	3	Espaço	NFR	Espaço [SGC]	Sistema	Sim

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação	Aspecto
11	O centro de controle deve receber e atualizar as seguintes informações durante o andamento da crise, em intervalos que não excedam 30 segundos: recursos empregados, vítimas civis, vítimas do gerenciamento de crise, localização do super observador, perímetro da crise, localização da equipe de resgate no local da crise, níveis de emissão no local da crise, tempo estimado para chegada no local da crise.	3	Tempo	NFR	Tempo [SGC]	Sistema	Sim
12	A atraso na comunicação da informação entre o centro de controle e a equipe de resgate, bem como entre membros do tipo de resgate, não pode exceder 500 milissegundos.	3	Tempo	NFR	Tempo [SGC]	Sistema	Sim
13	O sistema deve ser capaz de recuperar qualquer informação armazenada com o máximo de atraso de 500 milissegundos.	3	Tempo	NFR	Tempo [SGC]	Sistema	Sim
14	O sistema deve definir políticas de acesso para várias classes de usuários. A política de acesso deve descrever os componentes e as informações que cada classe deve adicionar, acessar e atualizar.	4	<i>Security</i>	NFR	Security [Coordenador-Resolver Crise, Coordenador-Capturar Relatório da Crise, Companhia de Telefone-Capturar Relatório da Crise, Sistema de Supervisão-Capturar Relatório da Crise, Empregado do SGC-Atribuir Recurso Interno, Sistema de Recurso Externo-Solicitar Recurso Externo, Recurso-Executar Missão, Super Observador-Executar Missão Super Observador, Hospital-Executar Missão Resgate, Enfermeiro-Executar Missão]	Nenhuma	Sim

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação	Aspecto
					Resgate]		
15	Se um usuário permanecer inativo por mais de 30 minutos, o sistema deve solicitar a sua reautenticação.	4	Security	NFR	Security [Autenticar Usuário]	Nenhuma	Não
16	O sistema deve autenticar usuários com base nas políticas de acesso quando estes acessarem pela primeira vez qualquer componente ou informação.	4	Security	NFR	Security [Autenticar Usuário]	Nenhuma	Não
17	Todas as comunicações do sistema devem usar canais seguros em conformidade com o padrão de criptografia AES-128.	4	Security	NFR	Security [SGC]	Sistema	Sim
18	Recursos de resgate devem ser capazes de acessar a informação em movimento.	5	Mobilidade	Lista Chung	Mobilidade [Recurso-Executar Missão]	Nenhuma	Não
19	O sistema deve prover informação sensível à localidade para os recursos de resgate.	5	Mobilidade	Lista Chung	Mobilidade [Resolver Crise]	Nenhuma	Não
20	Recursos de resgate devem comunicar sua localização ao centro de controle.	5	Mobilidade	Lista Chung	Mobilidade [Recurso-Executar Missão]	Hospital-Executar Missão Resgate, Enfermeiro-Executar Missão Resgate, Super Observador-Executar Missão Super Observador	Sim
21	O sistema deve acessar mapas detalhados, dados de terreno e condições do tempo para o local da crise e as rotas que levam até o local.	5	Mobilidade	Lista Chung	Mobilidade [Resolver Crise]	Nenhuma	Não
22	O sistema deve registrar as seguintes informações estatísticas, nas crises em andamento e resolvidas: taxa de progressão; tempo médio de resposta das equipes de resgate, tempo de resposta individual de cada equipe de resgate, taxa de sucesso de cada equipe de resgate, taxa de vítimas, taxa de sucesso das missões.	6	Observabilidade	Lista Chung	Observabilidade [SGC]	Sistema	Sim

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação	Aspecto
23	O sistema deve prover ferramentas de análise estatística para analisar os dados da crise em múltiplas crises.	6	Observabilidade	Lista Chung	Observabilidade [SGC]	Sistema	Sim
24	O sistema deve gerenciar ao menos 1000 chamadas de vítimas ao mesmo tempo.	7	Desempenho	NFR	Desempenho [Capturar o Relatório da Crise]	Nenhuma	Não
25	O sistema deve gerenciar comunicação, coordenação e acesso à informação a pelo menos 20000 recursos de resgate em atuação ao mesmo tempo.	7	Desempenho	NFR	Desempenho [Empregado do SGC-Atribuir Recurso Interno, Sistema de Recurso Externo-Solicitar Recurso Externo, Super Observador-Executar Missão Super Observador, Recurso-Executar Missão, Hospital-Executar Missão Resgate, Enfermeiro-Executar Missão Resgate]	Nenhuma	Sim
26	O sistema deve gerenciar pelo menos 100 crises ao mesmo tempo.	7	Desempenho	NFR	Desempenho [Resolver Crise]	Capturar o Relatório da Crise, Atribuir Recurso Interno, Solicitar Recurso Externo, Executar Missão, Executar Missão Resgate, Executar Missão Super Observador	Sim
27	O sistema deve gerenciar pelo menos 200 missões por crise ao mesmo tempo.	7	Desempenho	NFR	Desempenho [Executar Missão]	Executar Missão Resgate, Executar Missão Super Observador	Sim
28	O sistema deve monitorar emissões do local da crise para determinar as distâncias de operação segura para os recursos de resgate.	1	Safety	ISO/IEC 9126	Safety [Executar Missão Super Observador]	Nenhuma	Não

Id	Requisito	Prioridade	Interesse	Origem do Interesse	Ponto de Associação	Propagação	Aspecto
29	O sistema deve monitorar as condições de terreno e tempo no local da crise para garantir a operação segura e afastamento dos recursos de resgate, e remoção de vítimas civis.	1	<i>Safety</i>	ISO/IEC 9126	Safety [Executar Missão Super Observador]	Nenhuma	Não
30	O sistema deve determinar o perímetro do local da crise para garantir a segurança dos civis e remoção de vítima a uma distância segura.	1	<i>Safety</i>	ISO/IEC 9126	Safety [Executar Missão Super Observador]	Nenhuma	Não
31	O sistema deve monitorar atividades criminais para garantir a segurança dos recursos de resgate, civis e vítimas.	1	<i>Safety</i>	ISO/IEC 9126	Safety [Executar Missão Super Observador]	Nenhuma	Não
32	A segurança do pessoal do resgate deve ter prioridade para o sistema.	1	<i>Safety</i>	ISO/IEC 9126	Safety [Executar Missão Super Observador]	Nenhuma	Não
33	O sistema deve recomendar estratégias alternativas para lidar com a crise, bem como com as condições de mudança da crise (exemplo: condições do tempo, condições do terreno, atividade civil ou criminal).	8	Adaptabilidade	ISO/IEC 9126	Adaptabilidade [Resolver Crise]	Executar Missão, Executar Missão Super Observador	Sim
34	O sistema deve recomendar ou listar recursos alternativos em caso de indisponibilidade ou falta de recursos apropriados.	8	Adaptabilidade	ISO/IEC 9126	Adaptabilidade [Resolver Crise]	Executar Missão, Executar Missão Super Observador	Sim
35	O sistema deve ser capaz de usar canais de comunicação alternativos em caso de indisponibilidade ou falta dos canais existentes.	8	Adaptabilidade	ISO/IEC 9126	Adaptabilidade [SGC]	Sistema	Sim
36	O sistema deve ser capaz de manter comunicação efetiva em áreas acidentadas ou com grande ruído no local da crise.	8	Adaptabilidade	ISO/IEC 9126	Adaptabilidade [SGC]	Sistema	Sim
37	O sistema deve ter acesso aos dados de mapa, terreno e temperatura com 99% de acurácia.	9	Acurácia	NFR	Acurácia [Resolver Crise]	Nenhuma	Não
38	O sistema deve registrar dados até o receptor sem modificações.	9	Acurácia	NFR	Acurácia [SGC]	Sistema	Sim
39	A comunicação entre o sistema e os recursos de resgate deve ter o fator de deterioração máxima de 0,0001 por 1000 quilômetros.	9	Acurácia	NFR	Acurácia [SGC]	Sistema	Sim

Os RNFs associados ao limite de sistema foram considerados aspectos, pois estes requisitos podem impactar diversos casos de uso ou diversos

artefatos em fases posteriores do desenvolvimento. Exemplos são RNFs de confiabilidade, disponibilidade, espaço e tempo.

Os RNFs de mobilidade (id 20), desempenho (id 26 e id 27) e adaptabilidade (id 33 e id 34) foram considerados aspectos por apresentarem propagações em diversos elementos do diagrama de casos de uso. Os RNFs de *security* (id 14) e desempenho (id 25) também foram considerados aspectos por estarem associados a diversos elementos do diagrama de casos de uso.

5.5.4 Sub-processo 4: Decompor requisitos não-funcionais

Este sub-processo é composto pelas atividades para selecionar SIG inicial a ser detalhado (4.1), decompor *softgoals* (4.2), definir prioridades (4.3), identificar operacionalizações (4.4), identificar correlações (4.5), registrar decisões de projeto (4.6), selecionar alternativas de projeto (4.7), aplicar o procedimento de avaliação (4.8) e definir relacionamentos de metas de operacionalizações (4.9). Todas estas atividades têm como principal objetivo elaborar os SIGs com base nos SIGs iniciais definidos no diagrama integrado.

A atividade para selecionar SIG inicial a ser detalhado (4.1) recebe como entrada o diagrama integrado, os SIGs iniciais e a especificação de requisitos não-funcionais. A seleção do SIG inicial para o detalhamento é feita considerando as prioridades dos RNFs contidas na especificação de requisitos não-funcionais (Tabela 60).

Para ilustrar a atividade para decompor requisitos não-funcionais são apresentados exemplos dos RNFs de *safety*, disponibilidade, espaço e tempo, que possuem, respectivamente, as prioridades 1, 2 e 3.

O SIG que representa o RNF de *safety* associado ao caso de uso *Executar Missão Super Observador* é apresentado na Figura 85.

Figura 85 – SIG1 de *safety* (SGC).

O SIG que representa o RNF de *safety* utiliza a decomposição por atributos relacionada ao caso de uso *Executar Missão Super Observador*. Estes atributos devem ser monitorados pelo Super Observador no local da crise. Os *softgoals* de operacionalização foram definidos pelo engenheiro de requisitos, uma vez que não há catálogo disponível para o RNF de *safety*. Foram identificadas quatro metas para quatro RNFs relacionados ao interesse de *safety*. As metas referem-se ao cálculo e monitoramento dos níveis de emissão no local da crise, ao monitoramento das condições de tempo e temperatura, à determinação do perímetro da crise e o monitoramento das atividades criminais. Como este requisito é o mais prioritário para o SGC, dois sinais de exclamação foram inseridos no SIG inicial indicando o tipo muito crítico para todas as operações relacionadas ao interesse de *safety*.

O SIG que representa o RNF de disponibilidade associado ao SGC é apresentado na Figura 86.

Figura 86 – SIG 2 de disponibilidade (SGC).

O SIG que representa o RNF de disponibilidade utiliza a decomposição de tipo de RNF proveniente da norma ISO/IEC 9126 (ISO/IEC 9126, 2001). O engenheiro de requisitos optou por representar a característica de qualidade de mais alto nível, a confiabilidade, e sua sub-característica de disponibilidade. Detectou-se uma correlação positiva relacionada ao interesse de confiabilidade, caso o interesse de disponibilidade fosse suficientemente satisfeito. Os *softgoals* de operacionalização são provenientes das táticas definidas por Bass, Clements e Kazman (2003). Três RNFs foram representados e decompostos por este SIG. A primeira meta, relacionada ao RNF (id 1) refere-se ao uso da temporização de homem morto para detectar componentes do sistema de software que não estejam mais respondendo. Nestes casos uma notificação de correção é enviada. A segunda meta refere-se ao ponto de verificação, resincronização e redundância passiva. O ponto de verificação refere-se à possibilidade de restauração do sistema utilizando o seu estado anterior mais consistente, em casos de falha. A resincronização é requerida quando o sistema de software precisa ser restaurado depois de uma falha. A abordagem de atualização do sistema de software depende do tempo em que o sistema ficou em falha e da quantidade de informação a ser atualizada. Finalmente, a terceira meta refere-se ao monitoramento de processos,

indexação, por sua vez, prejudica o requisito de espaço, pois um índice deve ser armazenado. Em contrapartida, a indexação auxilia o acesso mais rápido de informações. As informações do SGC foram agrupadas por tipo neste SIG, sendo informações sobre a crise, recursos e estratégias de resolução da crise, a fim de simplificar a representação do grafo. Seis RNFs foram representados neste SIG por estarem relacionados ao limite de sistema e ao mesmo tipo de interesse.

O SIG que representa o RNF de tempo associado ao SGC é apresentado na Figura 88.

Figura 88 – SIG 5 de tempo (SGC).

O SIG que representa o RNF de tempo define duas metas, uma definida pelo engenheiro de requisitos e outra baseada nas táticas de Bass, Clements e

Kazman (2003). O SIG de tempo contempla a representação e decomposição de dois requisitos RNFs (id 11 e id 12), associados ao limite do sistema. A meta, definida pelo engenheiro de requisitos e relacionada ao requisito (id 11) refere-se à atualização periódica de dados a cada 30 segundos de acordo com relógio do sistema. A meta baseada nas táticas refere-se ao aumento da eficiência computacional, redução de sobrecarga computacional e limite do tempo de execução. O aumento da eficiência computacional trata da melhoria dos algoritmos utilizados em áreas críticas a fim de diminuir a latência. A redução da sobrecarga computacional sugere a retirada de passos intermediários para processamento de eventos, o que reduz consideravelmente a latência. O uso desta tática representa uma correlação positiva com o *softgoal* de modificabilidade, uma vez que o uso de passos intermediários auxilia no controle e localização de pontos a serem modificados.

A limitação do tempo de execução define que um tempo limite deve ser estabelecido para que uma resposta seja dada a um evento, ou mesmo que o número de iterações seja limitado a fim de reduzir o tempo de execução.

5.5.5 Sub-processo 5: Elaborar modelo composto

Este sub-processo é composto pelas atividades para identificar metas operacionais e restrições de projeto (5.1), definir pontos de composição (5.2), confirmar propagação (5.3), resolver conflitos (5.4) e atualizar o modelo do sistema de software (5.5).

A atividade para identificar metas operacionais e restrições de projeto (5.1) recebe como dados de entrada os SIGs e a tabela de RNF com aspectos. Nesta atividade deve ser elaborada a tabela de RNF com metas operacionais e restrições de projeto; que é a tabela de aspectos (Tabela 62) acrescida das colunas: id SIG, meta, origem da meta, meta operacional ou restrição de projeto e descrição. Esta tabela tem a finalidade de registrar a classificação do tipo de meta referente a cada SIG e a ação a ser tomada caso a meta seja classificada como meta operacional ou restrição de projeto.

A tabela de RNF com metas operacionais e restrições de projeto é apresentada na Tabela 63.

Tabela 63 – Tabela de RNF com metas operacionais e restrições de projeto (SGC).

Id	Id do SIG	Meta	Origem da Meta	Meta Operacional ou Restrição de Projeto	Descrição
1	SIG2	Utilizar a tática de temporização de homem morto a fim de auxiliar na detecção de falhas.	Tática	Restrição de Projeto	Os componentes do sistema devem emitir a mensagem de homem morto periodicamente.
2	SIG2	Implementar ponto de verificação, resincronização e redundância passiva a fim de recuperar falhas.	Tática	Restrição de Projeto	Implementar <i>back up</i> de informações, sincronização de estado e ponto de verificação a fim de que o sistema volte à operação sem prejuízos.
3	SIG2	Monitorar processos, transações e remover serviços a fim de prevenir falhas.	Tática	Restrição de Projeto	A manutenção será realizada de maneira preventiva, com a remoção do serviço, controle de transações e monitoração de processo.
4	SIG3	Definir uma arquitetura de hardware com equipamentos que possuam especificação de alta disponibilidade, verificação e correção de erros.	Definida pelo engenheiro de requisitos	Restrição de Projeto	Definição de uma arquitetura de hardware que possua alta disponibilidade, verificação e correção de erros.
5	SIG4	Armazenar e atualizar informações da crise utilizando vários atributos por tupla.	NFR	Restrição de Projeto	Definir a arquitetura do banco de dados com os dados da crise com base em vários atributos por tupla.
6	SIG4	Indexar o armazenamento com as informações da crise para que seja de rápida recuperação.	NFR	Restrição de Projeto	A cada novo armazenamento e atualização de informações da crise indexar os dados conforme id da crise.
7	SIG4	Armazenar e atualizar informações sobre os recursos utilizando vários atributos por tupla.	NFR	Restrição de Projeto	Definir a arquitetura do banco de dados com os dados da crise com base em vários atributos por tupla.
8	SIG4	Indexar o armazenamento com as informações dos recursos para que seja de rápida recuperação.	NFR	Restrição de Projeto	A cada novo armazenamento e atualização de informações dos recursos indexar dados conforme id da crise.
9	SIG4	Armazenar e atualizar informações sobre as estratégias de resolução da crise utilizando vários atributos por tupla.	NFR	Restrição de Projeto	Definir a arquitetura do banco de dados com os dados da crise com base em vários atributos por tupla.
10	SIG4	Indexar o armazenamento com as informações da crise para que seja de rápida recuperação.	NFR	Restrição de Projeto	A cada novo armazenamento e atualização de informações sobre as estratégias de resolução da crise indexar dados conforme id da crise.
11	SIG5	Atualizar dados periodicamente a cada 30 segundos de acordo com o relógio do sistema.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Atualizar Dados.
12	SIG5	Aumentar a eficiência computacional, reduzir o overhead computacional e limitar o tempo de execução a fim de obter melhorias na comunicação.	Tática	Restrição de Projeto	Aplicar estratégias de implementação a fim de aumentar a eficiência computacional, reduzir o overhead e limitar o tempo de execução.

Id	Id do SIG	Meta	Origem da Meta	Meta Operacional ou Restrição de Projeto	Descrição
13	SIG5	Definida no SIG Espaço [SGC] com a indexação de dados da crise.	NA	NA	NA
14	SIG6	Definir políticas de acesso por classe de usuário.	Definida pelo engenheiro de requisitos	Restrição de Projeto	Definir políticas de acesso por classe de usuário.
15	SIG7	Depois de 30 minutos solicitar reautenticação do usuário e emitir alarme de software.	NFR	Meta Operacional	Novo caso de uso Solicitar Reautenticação
16	SIG7	Solicitar identificação, autenticação única por senha e validação simples de regras de acesso com base nas políticas de acesso definidas por classe de usuário.	NFR	Meta Operacional	Novo caso de uso Validar Acesso
17	SIG7	Para todas as comunicações do sistema deve ser utilizado o padrão de encriptação AES-128.	NFR	Meta Operacional	Novo caso de uso Criptografar Comunicação
18	SIG8	Os recursos devem acessar informação em movimento.	Definida pelo engenheiro de requisitos	Restrição de Projeto	O sistema deve permitir o uso de equipamentos móveis para o acesso à informação.
19	SIG9	O sistema deve acessar o sistema GPS e solicitar informações sobre rotas e localização.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Obter Localização
20	SIG8	Os recursos devem comunicar sua localização para o centro de controle.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Comunicar Localização
20	SIG8	Os recursos devem comunicar sua localização para o centro de controle.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Comunicar Localização
21	SIG9	O sistema deve ter acesso à informações de mapa, terreno e condições do tempo do local da crise.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Obter Informações Geográficas e Meteorológicas
22	SIG10	A informação estatística deve ser armazenada em outro dispositivo de armazenamento.	Definida pelo engenheiro de requisitos	Restrição de Projeto	Armazenar informações estatísticas em outro dispositivo de armazenamento.
23	SIG10	Uma ferramenta de análise estatística deve estar disponível para análise de múltiplas crises.	Definida pelo engenheiro de requisitos	Restrição de Projeto	Uma ferramenta de análise estatística deve estar disponível para análise de múltiplas crises.
24	SIG11	Controlar o número máximo de chamadas e os recursos utilizados para processá-las.	Tática	Restrição de Projeto	Implementar mecanismos para controlar o número máximo de chamadas recebidas e a alocação dos recursos utilizados.
25	SIG12	Aplicar concorrência utilizando diferentes <i>threads</i> .	Tática	Restrição de Projeto	Utilizar processamento de requisições em paralelo através de <i>threads</i> .
26	SIG13	Aplicar concorrência utilizando diferentes <i>threads</i> .	Tática	Restrição de Projeto	Utilizar processamento de requisições em paralelo através de <i>threads</i> .

Id	Id do SIG	Meta	Origem da Meta	Meta Operacional ou Restrição de Projeto	Descrição
27	SIG14	Aplicar concorrência utilizando diferentes <i>threads</i> .	Tática	Restrição de Projeto	Utilizar processamento de requisições em paralelo através de <i>threads</i> .
28	SIG1	O sistema deve calcular e monitorar os níveis de emissão.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Monitorar Emissões
29	SIG1	O sistema deve monitorar condições de tempo e terreno.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Monitorar Terreno e Tempo
30	SIG1	O sistema deve determinar o perímetro da crise.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Determinar Perímetro
31	SIG1	O sistema deve monitorar atividades criminais.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Monitorar Crimes
32	SIG1	Definição de prioridade no SIG1.	NA	Meta Operacional	Prioridade deve ser definida em cada caso de uso safety.
33	SIG15	O sistema deve recomendar estratégias alternativas para lidar com a crise.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Definir Estratégias Alternativas
34	SIG15	O sistema deve recomendar recursos alternativos para lidar com a crise.	Definida pelo engenheiro de requisitos	Meta Operacional	Novo caso de uso Definir Recursos Alternativos
35	SIG16	O sistema deve possuir canais alternativos em casos de indisponibilidade.	Definida pelo engenheiro de requisitos	Restrição de Projeto	Definir arquitetura com multi-canais de comunicação e monitoramento de indisponibilidade.
36	SIG16	O sistema deve possuir canais que suportem comunicação em áreas acidentadas ou de grande ruído.	Definida pelo engenheiro de requisitos	Restrição de Projeto	Utilizar canais que suportem comunicação em áreas acidentadas ou de grande ruído.
37	SIG17	Fazer verificação em outro canal para confirmar informações de mapa, terreno e tempo.	NFR	Meta Operacional	Novo caso de uso Confirmar Informações
38	SIG18	Implementar código de verificação para garantir informação atualizada e consistente entre emissor e receptor.	NFR	Restrição de Projeto	Utilizar códigos de verificação em todas as comunicações.
39	SIG18	Melhorar o canal de comunicação a fim de respeitar o fator de deterioração.	NFR	Restrição de Projeto	Definir arquitetura de canais a fim de respeitar o fator de deterioração.

Nesta tabela podem ser observadas as restrições de projeto a serem impostas ao SGC e os novos casos de uso a serem adicionados no modelo de casos de uso.

O RNF (id 1) foi considerado uma restrição de projeto, pois a emissão contínua de mensagens para detectar componentes do sistema de software que não estejam mais respondendo deverá ser uma estratégia adotada durante o projeto e implementação dos componentes do SGC. O RNF (id 11) foi considerado uma meta operacional, pois adiciona uma nova função ao SGC.

Esta nova função define que os dados do sistema devem ser atualizados a cada 30 segundos.

A atividade para definir pontos de composição (5.2) tem a finalidade de identificar os pontos no fluxo de seqüência de eventos dos casos de uso, onde as metas operacionais devem ser inseridas. A Tabela 64 apresenta a tabela de RNF com composição, que é a tabela de RNF com metas operacionais e restrições de projeto (Tabela 63), acrescida das colunas: caso de uso, ponto de composição, regra de composição e condições da composição.

Tabela 64 – Tabela de RNF com composição (SGC).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição
11	Resolver Crise	10	sobreposição.depois	Nenhuma
	Atribuir Recurso Interno	2	sobreposição.depois	Nenhuma
	Solicitar Recurso Externo	2	sobreposição.depois	Nenhuma
	Executar Missão Super Observador	7	sobreposição.depois	Nenhuma
	Executar Missão Resgate	4	sobreposição.depois	Nenhuma
15	Autenticar Usuário	3	sobreposição.depois	Nenhuma
16	Autenticar Usuário	3	substituição	O sistema identifica, autentica e valida o acesso do usuário com base nas políticas de acesso.
17	Resolver Crise	6C1, 8, 8A1, 8B1, 10, 11	sobreposição.depois	Nenhuma
	Capturar o Relatório da Crise	2A1, 2A2, 3A1	sobreposição.depois	Nenhuma
	Atribuir Recurso Interno	1, 1A1, 1A2, 2, 2A	sobreposição.depois	Nenhuma
	Solicitar Recurso Externo	1, 2, 2A, 2B	sobreposição.depois	Nenhuma
	Executar Missão Super Observador	1, 2, 3, 4, 5, 6, 7A1, 7B1, 8, 8A1	sobreposição.depois	Nenhuma
	Executar Missão Resgate	1, 2, 3, 4, 4A, 4A1, 5, 6, 7	sobreposição.depois	Nenhuma
	Autenticar Usuário	1, 2	sobreposição.depois	Nenhuma
19	Resolver Crise	4, 5	sobreposição.depois	Nenhuma
20	Executar Missão Resgate	1	sobreposição.antes	Nenhuma
	Executar Missão Super Observador	1	sobreposição.antes	Nenhuma

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição
21	Resolver Crise	4, 5	sobreposição.depois	Nenhuma
28	Executar Missão Super Observador	1	sobreposição.antes	Nenhuma
29	Executar Missão Super Observador	1	sobreposição.antes	Nenhuma
30	Executar Missão Super Observador	1	sobreposição.antes	Nenhuma
31	Executar Missão Super Observador	1	sobreposição.antes	Nenhuma
33	Resolver Crise	12	sobreposição.depois	Nenhuma
	Executar Missão Super Observador	7A1	sobreposição.depois	Nenhuma
	Monitorar Emissões	4A2	substituição	O sistema sugere estratégias alternativas para a crise em andamento.
	Monitorar Terreno e Tempo	2A2, 4A2	substituição	O sistema sugere estratégias alternativas para a crise em andamento.
	Monitorar Crimes	4A2	substituição	O sistema sugere estratégias alternativas para a crise em andamento.
34	Resolver Crise	4A1, 5A	sobreposição.depois	Nenhuma
	Executar Missão Super Observador	7A1	sobreposição.depois	Nenhuma
37	Obter Informações Geográficas e Meteorológicas	3	sobreposição.depois	Nenhuma

Somente são exibidas na Tabela 64 as linhas que representam metas operacionais, com o caso que receberá a inclusão de um novo caso de uso, o passo do seu fluxo afetado, o tipo de regra de composição e, caso necessário, a condição da composição. Os RNFs (id 11, id 16, id 17, id 20, id 33 e id 34) possuem diversas linhas na tabela, pois são aspectos do SGC.

O SGC utilizou as regras de composição de sobreposição e substituição. Um exemplo do uso da regra de substituição é do requisito não-funcional (id 16). O novo caso de uso *Validar Acesso* substitui o passo três do fluxo básico do caso de uso *Autenticar Usuário*.

A atividade para confirmar as propagações (5.3) detectou necessidade de revisão dos SIGs 5, 7, 15 e 17.

No SIG 5, referente à meta relacionada ao RNF de tempo (id 11), a propagação que inicialmente ocorria no SGC ficou limitada aos casos de uso *Resolver Crise*, *Atribuir Recurso Interno*, *Solicitar Recurso Externo*, *Executar Missão*, *Executar Missão Super Observador* e *Executar Missão Resgate*, pois

os dados do SGC a serem recebidos e atualizados devem ser providos pelos casos de uso mencionados.

No SIG 7, referente ao RNF de *security* (id 17), a propagação que inicialmente ocorria no SGC foi distribuída para os casos de uso *Resolver Crise*, *Capturara o Relatório da Crise*, *Atribuir Recurso Interno*, *Solicitar Recurso Externo*, *Executar Missão*, *Executar Missão Super Observador*, *Executar Missão Resgate*, *Autenticar Usuário*, pois foram identificados os pontos, nos passos dos casos de uso, onde deverá ser aplicada criptografia de dados.

No SIG 15, referente ao RNF de adaptabilidade (id 33 e id 34), novos casos de uso, como *Monitorar Emissões*, *Monitorar Terreno e Tempo* e *Monitorar Crimes* se inter-relacionam pois estratégias alternativas devem ser sugeridas pelo sistema, caso alguma situação de perigo seja detectada.

Finalmente no SIG 17, referente ao RNF de acurácia (id 37), a associação do requisito não-funcional deve ser feita no novo caso de uso *Obter Informações Geográficas e Meteorológicas*, pois este caso de uso tem a responsabilidade de coletar informações de tempo, terreno e mapa, retirando desta forma, a responsabilidade inicialmente atribuída ao caso de uso *Resolver Crise*.

Conflitos foram detectados no SGC na atividade para resolver conflitos (5.4) nos casos de uso *Atribuir Recurso Interno*, *Solicitar Recurso Externo*, *Executar Missão Resgate*, *Executar Missão Super Observador* e *Resolver Crise*.

Os conflitos detectados no caso de uso *Atribuir Recurso Interno* estão apresentados na Tabela 65.

Tabela 65 – Tabela de RNF com prioridades – caso de uso Atribuir Recurso Interno (SGC).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição	Prioridade da Composição
11	Atribuir Recurso Interno	2	sobreposição.depois	Nenhuma	2
17	Atribuir Recurso Interno	1, 1A1, 1A2, 2, 2A	sobreposição.depois	Nenhuma	1

O RNF de *security*, relacionado à criptografia dos dados, deve ser composto antes do requisito não-funcional de tempo, que diz respeito à atualização de dados.

Os conflitos detectados no caso de uso *Solicitar Recurso Externo* são apresentados na Tabela 66.

Tabela 66 – Tabela de RNF com prioridades – caso de uso Solicitar Recurso Externo (SGC).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição	Prioridade da Composição
11	Solicitar Recurso Externo	2	sobreposição.depois	Nenhuma	2
17	Solicitar Recurso Externo	1, 2, 2A, 2B	sobreposição.depois	Nenhuma	1

Neste caso de uso uma situação similar ao caso de uso *Atribuir Recurso Interno* foi detectada para o caso de uso *Solicitar Recurso Externo*. O RNF de *security*, relacionado à criptografia de dados, deve ser composto antes do requisito não-funcional de tempo.

Os conflitos detectados no caso de uso *Executar Missão Resgate* são apresentados na Tabela 67.

Tabela 67 – Tabela de RNF com prioridades – caso de uso Executar Missão Resgate (SGC).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição	Prioridade da Composição
11	Executar Missão Resgate	4	sobreposição.depois	Nenhuma	2
17	Executar Missão Resgate	1, 2, 3, 4, 4A, 4A1, 5, 6, 7	sobreposição.depois	Nenhuma	Passo 1: 1 Passo 4: 1
20	Executar Missão Resgate	1	sobreposição.antes	Nenhuma	2

No passo 1 do caso de uso *Executar Missão Resgate*, o requisito não-funcional de *security* possui maior prioridade de composição em relação ao requisito não-funcional de mobilidade, e no passo 4, o interesse de *security* possui maior prioridade de composição em relação ao requisito não-funcional de tempo. Este fato se deve à necessidade de criptografia de dados a ser executada antes de qualquer operação.

Os conflitos detectados no caso de uso *Executar Missão Super Observador* são apresentados na Tabela 68.

Tabela 68 – Tabela de RNF com prioridades – caso de uso Executar Missão Super Observador (SGC).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição	Prioridade da Composição
17	Executar Missão Super Observador	1, 2, 3, 4, 5, 6, 7A1, 7B1, 8, 8A1	sobreposição.depois	Nenhuma	Passo 1: 1 Passo 7A1: 1
20	Executar Missão Super Observador	1	sobreposição.antes	Nenhuma	2
28	Executar Missão Super Observador	1	sobreposição.antes	Nenhuma	3
29	Executar Missão Super Observador	1	sobreposição.antes	Nenhuma	5
30	Executar Missão Super Observador	1	sobreposição.antes	Nenhuma	4
31	Executar Missão Super Observador	1	sobreposição.antes	Nenhuma	6
33	Executar Missão Super Observador	7A1	sobreposição.depois	Nenhuma	2
34	Executar Missão Super Observador	7A1	sobreposição.depois	Nenhuma	2

No passo 1 do caso de uso *Executar Missão Super Observador*, o RNF de *security* possui a maior prioridade de composição, seguido dos RNFs de *mobilidade* e *safety*. No passo 7A1, o RNF de *security* vem seguido do RNF de *adaptabilidade*, na definição de estratégias e recursos alternativos.

Os conflitos detectados no caso de uso *Resolver Crise* são apresentados na Tabela 69.

Tabela 69 – Tabela de RNF com prioridades – caso de uso Resolver Crise (SGC).

Id	Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição	Prioridade da Composição
11	Resolver Crise	10	sobreposição.depois	Nenhuma	2
17	Resolver Crise	6C1, 8, 8A1, 8B1, 10, 11	sobreposição.depois	Nenhuma	1
19	Resolver Crise	4, 5	sobreposição.depois	Nenhuma	1
21	Resolver Crise	4, 5	sobreposição.depois	Nenhuma	2

No passo 1 do caso de uso *Resolver Crise*, o RNF de *security* possui a maior prioridade de composição. Nos passos 4 e 5 a obtenção dos dados de

localização é prioritária em relação à obtenção dos dados de mapa, terreno e condições de tempo.

Na última atividade para atualizar o modelo do sistema de software (5.5), um novo diagrama de casos de uso foi gerado, com os novos elementos incluídos. Como o SGC é um sistema de software com muitos RNFs a serem integrados, vários diagramas atualizados de casos de uso foram gerados, a fim de facilitar a visualização. Os diagramas atualizados foram separados pelos seguintes interesses: *acurácia* e *safety*, adaptabilidade, mobilidade, *security* e tempo. Os diagramas foram separados desta forma pois em algumas situações ocorreram interdependências entre estes RNFs, que geraram relacionamentos entre os casos de uso incluídos no modelo. A Figura 89 apresenta um exemplo do diagrama integrado com o interesse de tempo.

Figura 89 – Diagrama de casos de uso atualizado – tempo (SGC).

O caso de uso destacado é o novo caso de uso incluído no diagrama, identificado com o estereótipo <<aspecto>>. Os demais diagramas atualizados do SGC encontram-se em Bombonatti (2010).

As especificações dos novos casos de uso incluídos devem ser elaboradas. A especificação do caso de uso *Atualizar Dados* é apresentada na Tabela 70.

Tabela 70 – Especificação do caso de uso Atualizar Dados (SGC).

Nome do Caso de Uso	Atualizar Dados
Descrição Breve	Este caso de uso tem o objetivo de atualizar dados da crise a cada 30 segundos.
Atores	Relógio.
Evento Inicial	Início da coleta de dados de 30 segundos.
Fluxo Básico	<ol style="list-style-type: none"> 1. O sistema solicita informações sobre o andamento da crise. 2. O sistema armazena a atualização das informações sobre a crise. 3. O caso de uso termina.
Fluxos Alternativos	Nenhum.
Sub-fluxos	Nenhum.
Requisitos Especiais	11
Pré-condições	Crise em andamento.
Pós-condições	Dados atualizados.
Pontos de Extensão	Nenhum.
Pontos de Inclusão	Nenhum.

O campo requisitos especiais deve ser preenchido com o identificador do requisito não-funcional que deu origem a nova especificação.

5.6 Avaliação das Aplicações do PARNAFOA

Com base nas respostas às questões relacionadas no início deste capítulo, são apresentadas, nesta seção, as informações obtidas sobre a aplicação do PARNAFOA nos cinco sistemas de software selecionados.

A apresentação dos dados obtidos na apresentação do PARNAFOA segue a ordem cronológica de construção, ou seja, na seguinte sequência: SGC, SGH, SBM, ATC e ONC.

Na Tabela 71 são apresentadas as informações consolidadas para avaliação conjunta das aplicações, de acordo com a sua ordem de construção.

Tabela 71 – Avaliação do PARNAFOA.

#	Questões	SGC	SGH	SBM	ATC	ONC
1	Qtd. RNFs	39	8	7	10	3
2	Qtd. Casos de uso	8	8	7	10	5
3	Qtd. atores	10	4	4	9	5
4	Qtd. <i>softgoals</i>	11	5	4	3	2
5	Adequação dos dados de entrada	<ul style="list-style-type: none"> •Elaboração do glossário 	<ul style="list-style-type: none"> •Elaboração do glossário •Revisão do modelo de casos de uso 	<ul style="list-style-type: none"> •Elaboração do glossário •Revisão do modelo de casos de uso 	<ul style="list-style-type: none"> •Elaboração do glossário •Elaboração da especificação de RNFs •Elaboração do modelo de casos de uso •Elaboração do diagrama de estados 	<ul style="list-style-type: none"> •Elaboração do glossário •Elaboração da especificação de RNFs •Revisão do modelo de casos de uso
6	Tempo gasto	15t	4t	3t	10t	6t
7	Melhoria que a aplicação trouxe ao PARNAFOA	<ul style="list-style-type: none"> •Refinamento das decomposições de tópico •Refinamento das heurísticas de ponto de associação •Refinamento das regras de propagação •Inclusão do id dos RNFs no texto das metas •Uso das táticas 	<ul style="list-style-type: none"> • Refinamento das decomposições de tipo 	<ul style="list-style-type: none"> •Análise da propagação por RNF ao invés do interesse 	<ul style="list-style-type: none"> • Preenchimento do campo de requisitos especiais, com o id do RNF, dos casos de uso provenientes das metas operacionais 	<ul style="list-style-type: none"> • Possibilidade de revisão das propagações • Possibilidade de detalhamento das especificações dos RNFs
8	Tipos de pontos de associação	Sistema Caso de uso Relação ator-caso de uso	Sistema Caso de uso Relação ator-caso de uso	Sistema Caso de uso Relação ator-caso de uso	Caso de uso Relação ator-caso de uso	Caso de uso
9	Tipos de propagação	Sistema Caso de uso Relação ator-caso de uso	Sistema Caso de uso	Nenhuma propagação	Caso de uso	Caso de uso
10	Propagação em casos de uso de inclusão	Sim	Não aplicável	Não aplicável	Não	Sim
11	Propagação em casos de uso de extensão	Não aplicável	Sim	Não aplicável	Não aplicável	Não aplicável
12	Propagação em casos de uso de generalização	Sim	Não aplicável	Não aplicável	Sim	Não aplicável
13	Aspectos identificados	29	6	1	3	1
14	Desmembramento de RNFs	Não	Não	Sim	Não	Sim
15	Revisão da propagação de RNFs	Sim	Sim	Não	Não	Não
16	Metas operacionais	14	3	6	7	1
17	Restrições de projeto	23	5	2	3	3

Como resposta à questão #1 é possível destacar que o SGC possui uma quantidade de RNFs superior aos demais sistemas de software, seguido pelo ATC, SGH, SBM e ONC.

Na questão #2, o ATC apresentou a maior quantidade de casos de uso, seguido pelo SGC, SGH, SBM e ONC.

Na questão #3, o SGC apresentou maior quantidade de atores, seguido pelos sistemas de software do ATC, ONC, SGH e SBM.

Na questão #4, a maior quantidade de *softgoals* foi apresentada pelo SGC, seguido pelo SGH, SBM, ATC e ONC.

Como resposta à questão #5, todos os sistemas de software apresentaram necessidade de adequação dos dados de entrada recebidos. O ATC foi o sistema de software que mais precisou de adequações. Isto se deve ao fato de que o ATC não tinha seus requisitos definidos a partir de um modelo de casos de uso.

Como resposta à questão #6, o maior tempo gasto nas aplicações do PARNAFOA foi a do SGC, com 15 unidades de tempo, seguido pelo ATC, ONC, SGH e SBM. A unidade de tempo é de aproximadamente 1 dia. O SGC levou mais tempo para ser concluído pois se trata de um sistema mais complexo, com quantidade maior de RNFs a serem tratados. O ATC apresentou a maior quantidade de adequações dos dados de entrada, o que consumiu uma parte do tempo da aplicação. O ONC ficou em terceiro lugar devido a maior dificuldade no entendimento do domínio da aplicação. O SGH e SBM são sistemas mais simples, de mais fácil entendimento, e com uma quantidade menor de RNFs, necessitaram de menos tempo.

Como resposta à questão #7, foram identificadas as contribuições que cada um dos sistemas de software apresentou para a melhoria do PARNAFOA. Como o SGC foi o primeiro sistema de software a ser considerado, a quantidade de melhorias apresentadas por ele também foi maior.

Nas questões #8 e #9, percebeu-se que nenhuma das aplicações apresentou pontos de associação e propagações no elemento ator do diagrama de casos de uso. Isto ocorre porque é raro que as especificações de RNFs estabeleçam regras ou procedimentos operacionais referentes aos atores dos sistemas de software, o que não quer dizer que a definição de RNFs associados aos atores não seja necessária.

Na aplicação do SGC, diversas propagações de RNFs ocorreram através dos elementos do diagrama de casos de uso. A propagação do RNF de desempenho ocorreu através de diversos casos de uso do SGC para o armazenamento periódico de dados; a propagação do RNF de mobilidade ocorreu através dos casos de uso com relacionamento de generalização; a propagação do RNF de adaptabilidade nos casos de uso do SGC, para lidar com estratégias e recursos alternativos merecem destaque. A associação de diversos *softgoals* no caso de uso *Resolver Crise* mostra que este caso de uso engloba diversas funções relacionadas ao SGC.

Na aplicação do SGH, o RNF relacionado à utilização de log das operações para fins de auditoria, gerou propagação em diversos casos de uso do SGH. Nesta aplicação ocorreu propagação do RNF de desempenho para manipular reservas concorrentes através de um caso de uso de extensão. O inter-relacionamento entre os RNFs foi percebido com a relação entre os novos casos de uso *Definir Preferência* e *Armazenar Log*, incluídos no modelo de casos de uso do SGH.

Na aplicação do SBM, não foram detectadas propagações dos RNFs.

Na aplicação do ATC, propagações entre casos de uso ocorreram somente entre casos de uso com relacionamento de generalização. O RNF de acurácia, relacionado à detecção do estado real das portas, foi propagado através de diversos casos de uso do ATC.

Na aplicação do ONC, foi detectada a propagação a partir do RNF de acurácia nos casos de uso que utilizam dados de posicionamento de veículos.

Na análise das respostas às questões #10, #11 e #12, percebeu-se que propagações de RNFs em casos de uso especializados ocorreram em todos os sistemas de software que possuíam este tipo de relacionamento. A mesma situação ocorre na propagação de um RNF em casos de uso estendidos. Para casos de uso incluídos, a propagação de RNFs nem sempre ocorre, como apresentado no ATC. Isso se deve ao fato de que os RNFs de *safety* são freqüentemente muito específicos às funções nas quais estão associados.

Como resposta à questão #13, todos os sistemas de software detectaram RNFs que foram considerados aspectos. O SGC apresentou a maior quantidade de aspectos, por possuir a maior quantidade de RNFs.

Como resposta à questão #14, o desmembramento de RNFs na especificação de RNFs foi necessário somente nos sistemas de software do SBM e ONC. O desmembramento do RNF de acurácia ocorreu no SBM e no ONC. Em ambas situações, o desmembramento foi necessário a fim de trazer maior clareza à especificação de RNFs.

Na questão #15, as revisões de propagação foram necessárias somente nos sistemas de software do SGC e SGH. A revisão de propagação ocorre quando se percebe que uma determinada propagação de um RNF não foi adequadamente tratada até o sub-processo 5 do PARNAFOA.

Na questão #16, o SGC foi o sistema de software que mais identificou metas operacionais, que foram incluídas como novos casos de uso no modelo de casos de uso, seguido pelo ATC, SBM, SGH e ONC.

Como resposta à questão #17, a maior quantidade de restrições de projeto foi detectada no SGC seguido pelo SGH, ATC, ONC e SBM. As restrições de projeto devem ser cuidadosamente gerenciadas para que possam ser incorporadas em fases posteriores do desenvolvimento.

5.7 Considerações Finais

A aplicação do PARNAFOA em cinco sistemas de software permitiu a identificação de melhorias e o detalhamento com maior precisão de seus sub-processos, procedimentos e artefatos.

Além disso, foi possível exercitar a elaboração dos SIGs, o uso dos catálogos do NFR *Framework*, a utilização da norma ISO/IEC 9126, a lista de RNFs de Chung et al. (2000) e as táticas de Bass, Clements e Kazman (2003).

Nas diversas aplicações, o *softgoal* de *security* sempre esteve associado à relação ator-caso de uso, pois este requisito trata das regras de comunicação pelas quais o ator interage com o sistema de software.

Foi possível também perceber que determinados RNFs dos sistemas de software, na maior parte dos casos, estão associados ao limite do sistema. Neste trabalho, os RNFs desta natureza foram: modificabilidade, acessibilidade, informativo, adaptabilidade, disponibilidade, observabilidade.

Isso se deve ao fato destes RNFs não se associarem especificamente a nenhum elemento do caso de uso, mas ao sistema de software como um todo.

Os RNFs tais como: desempenho, confiabilidade, acurácia e *safety* relacionaram-se, em algumas situações, ao limite do sistema e, em outras, aos casos de uso. Em poucos casos ocorreu a associação destes tipos de RNFs à relação ator-caso de uso. Isso ocorre, pois estes RNFs podem ser específicos aos casos de uso do sistema de software ou mais gerais. O tipo de associação a ser feita também depende do nível de detalhamento dos requisitos não-funcionais dos sistemas de software.

Com a aplicação do PARNAFOA nos cinco sistemas de software foi possível constatar que a maneira pela qual os RNFs se propagam possui relação com a forma pela qual os requisitos funcionais são modelados. Caso os requisitos funcionais estejam espalhados no modelo de casos de uso, os requisitos não-funcionais também ficam espalhados. Exemplos desta situação podem ser encontrados no ONC, na verificação do posicionamento de veículos executada por diversos casos de uso. No SGC esta situação ocorre no tratamento de estratégias e recursos alternativos, em que dois casos de uso fazem uso deste tipo de estratégia. No ATC, a verificação do estado real das portas tem impacto em três casos de uso do sistema de software. Caso estes requisitos funcionais estivessem representados em um único caso de uso, a propagação dos RNFs seguiria a mesma regra.

A inexistência ou existência de propagações dos RNFs, através dos elementos do modelo de casos de uso, pode indicar se um caso de uso realmente executa a função que lhe foi conferida, ou se executa outras funções não diretamente relacionadas ao seu objetivo.

É importante ressaltar que as características de espalhamento e entrelaçamento existentes no sistema de software não deixam de existir com o uso do PARNAFOA, porém estas relações são mapeadas, gerenciadas e rastreadas.

Foi possível também confirmar que os aspectos definidos durante a fase de requisitos ocorrem quando um caso de uso é incluído ou estendido mais de uma vez por outros casos de uso, como mencionado por Alencar et al. (2006).

Foi possível também verificar que, somente com o preenchimento do campo requisitos especiais nas especificações de casos de uso, conforme

estabelecido pelo método tradicional de casos de uso (JACOBSON; BOOCH; RUMBAUGH, 1999), não é possível determinar o impacto dos RNFs através dos elementos do diagrama de casos de uso. Uma análise detalhada da associação e propagação de um RNF, como proposta pelo PARNAFOA é necessária para esclarecer estes impactos.

Constatou-se que o uso do PARNAFOA auxilia na determinação de metas associadas aos RNFs para análises de segurança, e a identificação da rastreabilidade das metas adicionadas através de novos casos de uso no modelo de casos de uso do sistema de software.

O uso da tabela de RNF, que é preenchida no decorrer da execução das atividades, permite visualizar a rastreabilidade dos RNFs desde o sub-processo inicial. Em contrapartida, o trabalho de preenchimento desta tabela é manual e requer disciplina para que as informações sejam adequadamente colocadas nas tabelas.

Constatou-se também que muitas decisões de projeto devem ser tomadas pelo engenheiro de requisitos, que vão desde a escolha da maneira pela qual os RNFs são decompostos pelos SIGs, à determinação da propagação e à identificação das metas. Estas decisões podem variar de acordo com a experiência do engenheiro de requisitos envolvido, o que pode levar a diferentes soluções de projeto. É necessário que os engenheiros de requisitos sejam treinados nos métodos e processo do PARNAFOA para que se possa obter maior homogeneidade dos resultados.

6. Considerações Finais

“Cedo ou tarde, você vai aprender, assim como eu aprendi, que existe uma diferença entre CONHECER o caminho e TRILHAR o caminho.” – Morpheus

Neste capítulo são apresentadas as considerações finais, através das conclusões, as contribuições e os trabalhos futuros que podem ser desenvolvidos.

6.1 Conclusões

Este trabalho apresentou a definição e aplicação do PARNAFOA. Este processo permite que RNFs sejam incorporados ao modelo do sistema de software desde as fases iniciais do seu desenvolvimento, através da utilização de métodos para tratamento de RNFs e métodos orientados a aspectos.

A aplicação do PARNAFOA em cinco tipos de sistemas de software, com características diversas, permitiu o seu aprimoramento. Cada aplicação, por exercitá-lo de forma diferente, permitiu que melhorias e detalhamento dos procedimentos e a simplificação de sub-processos e artefatos pudessem ser realizadas.

O PARNAFOA mostrou como integrar interesses funcionais e não-funcionais. A análise da propagação de um RNF mostra como as funções que surgem, devido a este requisito são incorporadas nos elementos do modelo de casos de uso. Neste contexto, as relações de inclusão, extensão e generalização do modelo de casos de uso devem ser consideradas durante a análise da propagação de um RNF.

Foi possível concluir que a maneira pela qual os interesses funcionais estão modelados interfere na forma de propagação dos requisitos não-funcionais no modelo de casos de uso dos sistemas de software. Um exemplo é o requisito não-funcional de desempenho para armazenamento de dados, que pode ser composto ao modelo do sistema em apenas um caso de uso ou em diversos

casos de uso, dependendo da maneira como o armazenamento de dados foi modelado.

Desta forma, a propagação de um RNF no modelo de casos de uso pode dar indicações sobre deficiências e possíveis melhorias a serem realizadas no modelo de requisitos funcionais dos sistemas de software.

A maneira pela qual um RNF é propagado no modelo de casos de uso também pode ser determinada pelas escolhas dos *softgoals* de operacionalizações selecionados como estratégias de projeto. Uma escolha pode representar uma nova função a ser adicionada no sistema de software, através de uma meta operacional, ou uma restrição a ser imposta ao sistema de software, através de uma restrição de projeto.

Considerando o RNF de acurácia, opções relacionadas à utilização de recurso adicional de hardware para confirmar a acurácia de uma informação ou a opção pela implementação de um mecanismo de verificação através de software, para a mesma finalidade, representam uma diferença significativa para o desenvolvimento. A primeira opção trata de uma restrição de projeto e, a segunda de uma meta operacional que pode ser propagada em diversos pontos do modelo do sistema de software. Estas decisões ficam a cargo do engenheiro de requisitos e da equipe de projeto.

Desta forma, pode-se dizer que o PARNAFOA é um processo que estabelece diretrizes para que sua aplicação possa ser feita sistematicamente, em diversos tipos de aplicações, domínios e complexidades. Ao mesmo tempo, o PARNAFOA permite que decisões sejam tomadas pelos engenheiros de requisitos e pela equipe de projeto, fazendo com que o modelo do sistema gerado como saída seja adequado às necessidades dos sistemas de software a serem desenvolvidos.

6.2 Contribuições

“A descoberta consiste em ver o que todo mundo viu e pensar o que ninguém pensou.” – Jonathan Swift

As contribuições deste trabalho estão relacionadas com a definição do processo propriamente dito e ao detalhamento do método NFR *Framework* de

Chung et al. (2000) e da abordagem orientada a *softgoals* e casos de uso de Supakkul e Chung (2004).

Na definição do PARNAFOA foram integrados métodos de tratamento de RNFs e métodos orientados a aspectos. Sua descrição contou com a definição de sub-processos, atividades, procedimentos e artefatos. Sua representação foi feita através da linguagem de modelagem de processos BPMN.

Diversos métodos orientados a aspectos foram estudados e considerados na definição do processo. Os métodos apresentados no capítulo 3 deste trabalho não possuem propostas detalhadas como o PARNAFOA, nem mesmo o trabalho do AOSD-Europe, que consolida os trabalhos da área de engenharia de requisitos orientada a aspectos de Chitchyan et al. (2006). O detalhamento apresentado pelo PARNAFOA tem a finalidade de permitir que o processo seja utilizado e reproduzido por outras pessoas e equipes.

Os detalhamentos propostos para o NFR *Framework* contemplaram os métodos de decomposição de tipo e tópico de requisitos não-funcionais e os *softgoals* de operacionalização, nos catálogos de operacionalizações.

O detalhamento do PARNAFOA apresentou novos detalhamentos aos catálogos de métodos de decomposição de tipo de RNF do NFR *Framework*, pois estes são restritos aos requisitos de acurácia, segurança e desempenho. O PARNAFOA utilizou a norma ISO/IEC 9126 e a lista de requisitos não-funcionais proposta por Chung (CHUNG et al., 2000) para expandir as opções de tipos de requisitos não-funcionais disponíveis.

O PARNAFOA apresentou o detalhamento do catálogo de métodos de decomposição de tópicos de RNFs a fim de representar nos SIGs a propagação dos RNFs através dos elementos do diagrama de casos de uso. Métodos de decomposição de limite de sistema, caso de uso, ator e relação ator-caso de uso foram definidos. A necessidade do detalhamento do catálogo de métodos de decomposição de tópico de RNF ocorreu pois, originalmente, o NFR *Framework* não propõe o uso de elementos provenientes do modelo de casos de uso em sua estrutura de tópicos.

O PARNAFOA apresentou o detalhamento do catálogo de operacionalizações através do uso de táticas (BASS; CLEMENTS; KAZMAN, 2003). As táticas podem expandir os catálogos de operacionalizações dos

seguintes RNFs: disponibilidade, modificabilidade, desempenho, *security*, testabilidade e usabilidade.

O detalhamento proposto pelo PARNAFOA para a abordagem orientada a *softgoals* e casos de uso de Supakkul e Chung (2004) teve o objetivo de refinar as heurísticas para relacionar *softgoals* aos elementos de casos de uso, detalhar as heurísticas para analisar as propagações dos RNFs através dos elementos do modelo de casos de uso e, por fim, definir um conjunto de regras para identificar aspectos, regras que não estão disponíveis originalmente na proposta de Supakkul e Chung (2004).

6.3 Trabalhos Futuros

“Nunca é tarde para tentar o desconhecido. Nunca é tarde para ir mais além.” – Gabriele D’Annunzio

Como trabalhos futuros devem ser consideradas a integração das atividades de identificação, separação e composição de interesses funcionais, pois estes interesses podem influenciar a maneira pela qual os interesses não-funcionais são propagados através dos elementos do diagrama de casos de uso dos sistemas de software.

É considerado como trabalho futuro a integração do PARNAFOA às técnicas de desenvolvimento de projeto e implementação orientada a aspectos, a fim de estabelecer diretrizes para fases posteriores do desenvolvimento.

A formalização das propostas de detalhamentos dos catálogos de tipo, tópico e operacionalizações de RNFs do NFR *Framework*, conforme diretrizes estabelecidas pelo NFR *Framework*, também são consideradas trabalhos futuros. O objetivo é uniformizar a informação proveniente das diversas fontes e fazer com que não seja necessário realizar diversas consultas em locais distintos para decompor e analisar RNFs.

Considera-se também a construção de catálogos de decomposição de requisitos não-funcionais voltados para domínios específicos. Este tipo de catálogo poderá ser uma fonte de soluções reutilizáveis para aplicações do mesmo tipo.

A aplicação do PARNAFOA pode ser utilizada para análise de segurança de sistemas críticos. Os analistas de segurança podem avaliar o impacto dos RNFs de *safety* no sistema de software para, então, realizar a análise de segurança propriamente dita.

O tratamento das restrições de projeto identificadas pelo PARNAFOA deve ser detalhado e diretrizes sobre como manipular estas restrições em fases posteriores do desenvolvimento devem ser propostas.

A rastreabilidade de requisitos é outro ponto que merece destaque para a evolução do PARNAFOA. A definição dos atributos e elementos de origem e destino a serem rastreados, bem como diretrizes para o gerenciamento de requisitos deve ser considerada como trabalhos futuros neste processo.

A aplicação externa do PARNAFOA deve ser considerada como uma maneira de transmitir o conhecimento adquirido e avaliar o desempenho de outras equipes no uso do processo. A validação dos RNFs por especialistas em cada uma das aplicações também deve ser considerada.

Finalmente poderá ser desenvolvida uma ferramenta que dê suporte à visualização e manipulação dos dados existentes nas tabelas de RNFs. Através de uma ferramenta diferentes visões das tabelas de RNFs, com mais ou menos detalhes, poderiam ser geradas e manipuladas com mais facilidade.

REFERÊNCIAS

ALENCAR, F. et al. Identifying Candidate Aspects with I-Star Approach. In: INTERNATIONAL CONFERENCE ON ASPECT-ORIENTED SOFTWARE DEVELOPMENT, 5., Bonn-Alemanha, 2006.

ALVES, M.B. Ontology to Multi-Dimensional Separation of Concerns in Requirement Engineering. In: WORKSHOP ON EARLY ASPECTS AT AOSD, 9., França, 2010.

AOSD-Europe (European Network of Excellence on Aspect-Oriented Software Development). Inglaterra. Apresenta os trabalhos da área desenvolvidos pelo grupo. Disponível em: <<http://www.aosd-europe.net/>>. Acesso em: 12 abr. 2010.

ARAÚJO et al. Aspect-Oriented Requirements with UML. In: WORKSHOP ON ASPECT-ORIENTED MODELLING WITH UML, 2., Dresden-Alemanha, 2002.

ARAÚJO, J.; MOREIRA, A. An Aspectual Use Case Driven Approach. In: JORNADAS DE INGENIERÍA DE SOFTWARE Y BASE DE DATOS, 8., Alicante-Espanha, 2003.

BANDOR, M. Process and Procedure Definition: A Primer. 2007. Disponível em: <<http://www.sei.cmu.edu/library/abstracts/presentations/processpro.cfm>>. Acesso em 13 jan. 2010.

BANIASSAD, E. et al. Discovering Early Aspects. **IEEE Software**, v. 23, n.1, p. 61-70, 2006.

BASS, L.; CLEMENTS, P.; KAZMAN, R. **Software Architecture in Practice**. New York: Addison Wesley, 2003, 560p.

BERG, K.; CONEJERO, J.M.; CHITCHYAN, R. **AOSD Ontology 1.0 – Public Ontology of Aspect- Orientation**. AOSD-EUROPE., 2005. (Report, AOSD-Europe-UT-01).

BOMBONATTI, D.L.G. **PARNAFOA: Aplicações do Processo de Análise de Requisitos Orientado a Aspectos**. São Paulo: Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo, 2010. (Estudo Técnico).

BOMBONATTI, D.L.G.; MELNIKOFF, S.S.S. Survey on Early Aspects Approaches: Non-Functional Crosscutting Concerns Integration in Software Systems. In: WSEAS INTERNATIONAL CONFERENCE ON COMPUTER ENGINEERING AND APPLICATIONS, 4., 2010, Cambridge, p. 137-142.

BOOCH, G. Object-Oriented Development. **IEEE Transactions on Software Engineering**. v. 12, n.2, p. 211-221, 1986.

BRITO, I. et al. Handling Conflicts in Aspectual Requirements Compositions. **Transactions on Aspect-Oriented Software Development III: Springer Berlin / Heidelberg**, v. 4620/2007, p. 144-166, 2007.

BRITO, I.; MOREIRA, A. Integrating the NFR framework in a RE model. In: WORKSHOP ON EARLY ASPECTS: ASPECT-ORIENTED REQUIREMENTS ENGINEERING AND ARCHITECTURE DESIGN, 3., Vancouver-Canadá, 2004.

BRITO, I.; MOREIRA, A. Towards a Composition Process for Aspect-Oriented Requirements. In: WORKSHOP ON EARLY ASPECTS: ASPECT-ORIENTED REQUIREMENTS ENGINEERING AND ARCHITECTURE DESIGN, 2., Boston-USA, 2003.

BRITO, I.; MOREIRA, A.; ARAÚJO, J. A requirements model for quality attributes. In: WORKSHOP ON EARLY ASPECTS: ASPECT-ORIENTED REQUIREMENTS ENGINEERING AND ARCHITECTURE DESIGN, 1., Enschede-Holland, 2002.

CHITCHYAN et al. **Initial Version of Aspect-Oriented Requirements Engineering Model**. AOSD-Europe, 2006. (Report, AOSD-Europe-ULANC-17).

CHITCHYAN et al. **Survey of Aspect-Oriented Analysis and Design Approaches**. AOSD-EUROPE, 2005. (Report, AOSD-Europe-ULANC-9).

CHRISSIS, M.B.; KONRAD, M.; SHRUM, S. CMMi Second Edition: Guidelines for Process Integration and Product Improvement. United States: Addison Wesley, 2007, 676p.

CHUNG, L. et al. **Non-Functional Requirements in Software Engineering**. Londres: Kluwer Academic Publishers, 2000. 439p.

CHUNG, L.; SUPAKKUL, S. Representing NFRs and FRs: A Goal-Oriented and Use Case-Driven Approach, In: SERA 2004: Revised Selected Papers, **Lecture Notes in Computer Science (LNCS)** 3647/ 2005, 2005. p. 29-41.

CLARKE, S.; BANIASSAD, E. **Aspect-Oriented Analysis and Design: The Theme Approach**. New York: Addison Wesley Professional, 2005. 366p.

CYSNEIROS, L.M. **Requisitos Não-Funcionais: Da Elicitação ao Modelo Conceitual**. 2001. 224p. Tese (Doutorado) – Departamento de Informática, PUC Rio, Rio de Janeiro, 2001.

CYSNEIROS, L.M.; LEITE, J.C.S. Utilizando Requisitos Não-Funcionais para Análise de Modelos Orientados a Dados. In: WORKSHOP DE ENGENHARIA DE REQUISITOS, 12., Maringá, 1998.

CYSNEIROS, L.M.; LEITE, J.C.S.P.; NETO, J.M.S. A Framework for integrating non-functional requirements into conceptual models. **Requirements Engineering: Springer London**, v. 6, n. 2, p. 97-115, 2001.

DIJKSTRA, E. W. **A Discipline of Programming**. New York: Prentice Hall, Inc., 1976, 217p.

DOERR, J. et al. Non-Functional Requirements in Industry – Three Case Studies Adopting an Experience-based NFR Method. In: IEEE INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING, 13., 2005, Paris. P. 373-384.

DUAN, C. et al. A Probabilistic Approach to Evaluating Early Aspects. In: WORKSHOP ON EARLY ASPECTS: ASPECT-ORIENTED REQUIREMENTS ENGINEERING AND ARCHITECTURE DESIGN, 5., 2006, China.

EADDY, M. et al. Do Crosscutting Concerns Cause Defects? **IEEE Transactions on Software Engineering**. v. 34, p. 497-515, 2008.

GARCIA, S.; TURNER, R. **CMMi Survival Guide: Just Enough Process**

Improvement. New York: Addison Wesley, 2007, 299p.

GREENWOOD, P. et al. On the Contributions of an End-to End AOSD Testbed. In: WORKSHOP ON EARLY ASPECTS: ASPECT-ORIENTED REQUIREMENTS ENGINEERING AND ARCHITECTURE DESIGN, Minneapolis, USA, 2007.

HAGGE, L.; LAPPE, K. Sharing Requirements Engineering Experience Using Patterns. **IEEE Software**, v. 22, n. 1, p. 24-31, 2005.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 9126:** Information Technology – Software Product Evaluation – Quality Characteristics and Guidelines for their Use. Geneve, 2001.

JACOBSON, I. Aspects: the missing link – Part 2. **Software Development Magazine**, p. 44-48, 2003b.

JACOBSON, I. The Case for Aspects – Part 1. **Software Development Magazine**, p. 32-37, 2003a.

JACOBSON, I. Use Cases and Aspects – Working Seamlessly Together. **Journal of Object Technology**, v. 2, n. 4, p. 7-28, 2003.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J. **The Unified Software Development Process.** United States of América: Addison-Wesley, 1999. 463p.

JACOBSON, I.; NG, P. **Aspect-Oriented Software Development with Use Cases.** New York: Addison Wesley Professional, 2004. 464p.

KICZALES, G. et al. Aspect-Oriented Programming. In: EUROPEAN CONFERENCE OBJECT-ORIENTED PROGRAMMING, 11., v. 1241 of LNCS – Springer Verlag, 1997, p. 220-242.

KIENZLE, J.; GUELF, N.; MUSTAFIZ, S. **Crisis Management Systems.** Canadá: School of Computer Science – McGill University, 2009. 29p. Um estudo de caso para modelagem orientada a aspectos, SOCS-TR-2009.3.

KOTONYA, G.; SOMMERVILLE, I. **Requirements Engineering – Processes and Techniques.** England: John Wiley Professional, 1998. 294p.

KOTSAKIS, E.; KETSELIDIS, M. **Eye in the Sky (ISKY).** Bélgica: Joint Research Centre (JRC), 2001. (Draft 2).

LEE, M. et al. **StarUML 5.0 User Guide.** 1995. Disponível em: <<http://staruml.sourceforge.net/docs/user-guide%28en%29/toc.html>>. Acesso em: 09 abr. 2010.

LEFFINGWELL, D.; WIDRIG, J. **Managing Software Requirements – A Unified Approach.** United States of América: Addison-Wesley, 2001. 491p.

LINDVALL et al. An evolutionary testbed for software technology evaluation. **NASA Journal of Innovations in Systems and Software Engineering.** v.1, n.1. p. 3–11, 2005.

MELNIKOFF, S.S.S. **Estudo de Caso: Bóia para Dados Meteorológicos.** São Paulo: Escola Politécnica da Universidade de São Paulo, 2009. 20p. Material didático.

MENDES, A. **Arquitetura de Software**. Rio de Janeiro: Campus, 2002. 212p.

MENDONÇA, M.G. et al. A Framework for Software Engineering Experimental Replications. In: IEEE INTERNATIONAL CONFERENCE ON ENGINEERING OF COMPLEX COMPUTER SYSTEMS, 13., 2008, Belfast, Northern Ireland, p. 203-212.

MILI, H.; ELKHARRAZ, A.; MCHEUICK, H. Understanding separation of concerns. In: WORKSHOP ON EARLY ASPECTS: ASPECT-ORIENTED REQUIREMENTS ENGINEERING AND ARCHITECTURE DESIGN, 3., 2004, Vancouver, Canadá, p. 411-428.

MOREIRA, A.; ARAÚJO, J.; BRITO, I. Crosscutting Quality Attributes for Requirements Engineering. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND KNOWLEDGE ENGINEERING, 14., 2002, Ischia, Itália.

MOREIRA, A.; ARAÚJO, J.; RASHID, A. A Concern-Oriented Requirements Engineering Model. In: CONFERENCE ON ADVANCED INFORMATION SYSTEMS ENGINEERING, 2005, Porto, Portugal.

MOREIRA, A.; ARAÚJO, J.; RASHID, A. Multi-Dimensional Separation of Concerns in Requirements Engineering. In: REQUIREMENTS ENGINEERING CONFERENCE, Paris, 13., 2005, p. 285-296.

MUSSBACHER, G. Aspect-Oriented User Requirements Notation: Aspects in Goal and Scenario Models. **Models in Software Engineering: Springer Berlin / Heidelberg**, v. 5002/2008, p. 305-316, 2008.

MUSSBACHER, G.; AMYOT, D.; WEISS, M. Visualizing Early Aspects with Use Case Maps. **Transactions on Aspect-Oriented Software Development III: Springer Berlin / Heidelberg**, v.4620/2007, p. 105-143, 2007.

NIU, N.; EASTERBROOK, S.; YU, Y. A Taxonomy of Asymmetric Requirements Aspects. **Early Aspects: Current Challenges and Future Directions: Springer Berlin / Heidelberg**, v. 4765/2007, p. 1-18, 2007.

OBJECT MANAGEMENT GROUP (OMG-BPMN). **Dtc/2009-08-14: Business Process Modeling Notation**, v. 2.0 – Beta 1. 2009. 496p.

OBJECT MANAGEMENT GROUP (OMG-UML). **Formal/2009-02-02: OMG Unified Modeling Language – Superstructure**, v. 2.2. 2009. 740p.

PAECH, B. What is a requirements engineer? **IEEE Software**, v.25 , n.4 p. 16-17, 2008.

PIVETA, et al. Termos em português para Desenvolvimento de Software Orientado a Aspectos. In: WORKSHOP BRASILEIRO DE DESENVOLVIMENTO DE SOFTWARE ORIENTADO A ASPECTOS, 1., Brasília.

PMI. **Um Guia do Conhecimento em Gerenciamento de Projetos (Guia PMBOK)**. Editora do PMI, 4ª Edição, 2009.

RASHID, A. et al. Early Aspects: A Model for Aspect-Oriented Requirements Engineering. In: IEEE JOINT INTERNATIONAL CONFERENCE ON REQUIREMENTS ENGINEERING, 10., 2002, Essen – Alemanha. p. 199-202.

RASHID, A.; MOREIRA, A.; ARAÚJO, J. Modularization and Composition of Aspectual

Requirements. In: INTERNATIONAL CONFERENCE ON ASPECT-ORIENTED SOFTWARE DEVELOPMENT, 2., Boston, 2003.

RATIONAL UNIVERSITY. **Mastering Requirements Management with Use Cases**. USA: Rational Software Group – Professional Education and Product Training, 2003. REQ480 Mastering Requirements Management with Use Cases, Student Manual, Version 2003.06.00, Part# 800-026337-000.

ROSENHAINER, L. Identifying Crosscutting Concerns in Requirements Specifications. In: WORKSHOP ON EARLY ASPECTS: ASPECT-ORIENTED REQUIREMENTS ENGINEERING AND ARCHITECTURE DESIGN, 2004, Vancouver.

SARDINHA, A. et al. Conflict Management in Aspect-Oriented Requirements Engineering. In: WORKSHOP ON EARLY ASPECTS AT AOSD, 9., França, 2010.

SCHNEIDER, G.; WINTERS, J.P. Applying Use Cases: A Practical Guide. New York: Addison-Wesley Professional, 1997. 188p.

SILVA, L.F. **Uma Estratégia Orientada a Aspectos para Modelagem de Requisitos**. 2006. 222p. Tese (Doutorado) – Departamento de Informática, PUC Rio, Rio de Janeiro, 2006.

SILVA, L.F.; LEITE, J.C.S.P. Integração de Características Transversais Durante a Modelagem de Requisitos. In: WORKSHOP BRASILEIRO DE SOFTWARE ORIENTADO A ASPECTOS, 2., Uberlândia, 2005a.

SILVA, L.F.; LEITE, J.C.S.P. Uma linguagem de modelagem de requisitos orientados a aspectos. In: WORKSHOP EM ENGENHARIA DE REQUISITOS, 8., Portugal, 2005b.

SOFTWARE ENGINEERING STANDARDS COMMITTEE OF THE IEEE COMPUTER SOCIETY. **IEEE Std.830-1998**: IEEE Recommended Practice for Software Requirements Specifications. United States of América, 1998.

SOUSA, G. et al. Separation of Crosscutting Concerns from Requirements to Design: Adapting an Use Case Driven Approach. In: WORKSHOP ON EARLY ASPECTS: ASPECT-ORIENTED REQUIREMENTS ENGINEERING AND ARCHITECTURE DESIGN, 3., Lancaster, 2004.

SOUSA, G. **Uma Abordagem Direcionada a Casos de Uso para o Desenvolvimento de Software Orientado a Aspectos**. 2004. 177p. Dissertação (Mestrado) – Centro de Informática, Universidade Federal de Pernambuco, Recife, 2004.

SOUSA, G.; SILVA, I.; CASTRO, J. Adapting the NFR Framework to Aspect-Oriented Requirements Engineering. In: SIMPÓSIO BRASILEIRO DE ENGENHARIA DE SOFTWARE, 17., Manaus, 2003, p. 177-192.

SUPAKKUL, S.; CHUNG, L. Integrating FRs and NFRs: A Use Case and Goal Driven Approach. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING RESEARCH, MANAGEMENT AND APPLICATIONS ,2., 2004, Los Angeles. P. 30-37.

TARR, et al. N Degree of Separation: Multi-Dimensional Separation of Concerns. In: INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, 21., 1999, Califórnia-Los Angeles, p. 107-119.

THAYER, R.; DORFMAN, M. System and Software Requirements Engineering. **IEEE**

Computer Society Press, 1990.

VIEIRA, F.; BRITO, I.; MOREIRA, A. Using Multi-criteria Analysis to Handle Conflicts During Composition. In: WORKSHOP ON EARLY ASPECTS: ASPECT-ORIENTED REQUIREMENTS ENGINEERING AND ARCHITECTURE DESIGN, Shanguai-China, 2006.

XAVIER, L. **Integração de Requisitos Não-Funcionais a Processos de Negócio: Integrando BPMN e RNF**. 2009. 86 p. Dissertação (Mestrado) - Universidade Federal de Pernambuco, Recife, 2009.

WOHLIN, C.; HOST, M.; HENNINGSSON, K. Empirical Research Methods in Software Engineering. In: EMPIRICAL METHODS AND STUDIES IN SOFTWARE ENGINEERING, Experiences from ESERNET, 2003.

APÊNDICE A – Notação BPMN (*Business Process Modeling Notation*)

O BPMN foi desenvolvido pela BPMI (*Business Process Management Initiative*) da OMG (*Object Management Group*). O principal objetivo do BPMN é prover uma notação que seja entendida por todos os usuários de negócios, desde analistas de negócio que criam as versões iniciais dos processos, até os desenvolvedores técnicos responsáveis pela implementação da tecnologia que irá executar estes processos, e finalmente, para as pessoas de negócio que irão gerenciar e monitorar estes processos. Portanto, o BPMN cria um padrão para cobrir a lacuna entre as definições de processos e sua implementação (OMG-BPMN, 2009). A seguir são apresentados os principais elementos desta notação utilizados neste trabalho.

Elemento	Descrição
Sub-Processo	Um sub-processo representa um conjunto de atividades. Um sub-processo é parte de um processo. Um sub-processo pode ser representado com ou sem seus detalhes aparentes. A figura ao lado representa um sub-processo fechado, sem seus detalhes aparentes. A representação de um processo aberto, com seus detalhes aparentes, é similar à representação de uma atividade.
Sub-Processo Ad-Hoc	Um sub-processo <i>ad-hoc</i> contém atividades que podem ser executadas e uma ordem flexível.
Sub-Processo Loop	Um sub-processo <i>loop</i> é repetido até que se atinja a condição de parada.

Elemento	Descrição
Atividade	<p>Uma atividade é um elemento atômico incluído dentro de um processo.</p>
Evento de Início	<p>O evento de início indica quando um processo se inicia.</p>
Evento Intermediário	<p>O evento intermediário indica o momento em que algum evento ocorre, entre o início e o fim do processo, afetando o seu andamento.</p>
Evento de Fim	<p>O evento de fim indica quando um processo é finalizado.</p>
Link	<p>Um <i>link</i> é um mecanismo que conecta duas seções em um processo. Pode ser utilizado para criar situações de retorno ou evitar longas seqüências de eventos. Pode ser utilizado como um desvio. A seta preenchida indica a origem em que ocorre o desvio e a seta vazia indica o destino do desvio.</p>
Decisão	<p>Uma decisão é um local dentro de um processo de negócios onde o fluxo pode tomar dois ou mais caminhos alternativos.</p>
Grupo	<p>Um grupo oferece um mecanismo visual para agrupar elementos de mesmo tipo de um diagrama.</p>

Elemento	Descrição
<p>Anotação</p>	<p>A anotação de texto é um mecanismo para oferecer informações adicionais aos leitores dos diagramas BPMN.</p>
<p>Objeto de Dado</p>	<p>Os objetos de dado fornecem informações sobre como documentos, dados e outros objetos são usados e atualizados durante o processo. Embora um objeto de dado possa implicar em um documento eletrônico, eles podem ser usados para representar diferentes tipos de objetos, tanto eletrônicos quanto físicos.</p>
<p><i>Pool</i></p>	<p>Um <i>pool</i> representa um participante do processo. Um participante pode ser uma entidade de negócio específica (uma empresa, por exemplo), ou uma função de negócio.</p>
<p>Fluxo de Seqüência</p>	<p>Um fluxo de seqüência é usado para mostrar a ordem em que uma atividade é executada. Cada fluxo possui uma única origem e destino.</p>
<p>Associação</p>	<p>Uma associação é utilizada para associar informações e artefatos aos objetos do fluxo.</p>
<p>Fluxo de Mensagem</p>	<p>Um fluxo de mensagem é usado para mostrar mensagens entre duas entidades.</p>

APÊNDICE B – Descrição dos Artefatos

Nome do Artefato: Documentos existentes (#1)

Descrição: documentos existentes sobre o sistema de software podem incluir o *workload* do sistema, o *workload* organizacional, fluxo de informações, especificações de requisitos funcionais, prioridades organizacionais, entre outros.

Padrão Adotado: não aplicável.

Regras de Preenchimento: não aplicável.

Nome do Artefato: Modelo de casos de uso (#2), modelo de casos de uso atualizado (#2A)

Descrição: especificação de casos de uso e diagrama de casos de uso.

Padrão Adotado: padrão da especificação de casos de uso conforme definido em Rational University (2003). O diagrama de casos de uso deve ser elaborado conforme representação proposta na UML 2 (OMG-UML, 2009).

Regras de Preenchimento:

Item da Especificação	Descrição
Nome do caso de uso	<nome do caso de uso>
Descrição breve	<descrição resumida da função do caso de uso>
Atores	<lista dos atores envolvidos no caso de uso>
Evento Inicial	<evento iniciador do caso de uso>
Fluxo Básico	<descrição do fluxo principal com os passos numerados seqüencialmente>
Fluxos Alternativos	<descrição dos fluxos alternativos com os passos numerados seqüencialmente>
Sub-fluxos	<descrição dos sub-fluxos com os passos numerados seqüencialmente>
Requisitos Especiais	<requisitos não-funcionais aplicáveis ao caso de uso>
Pré-condição	<condição verdadeira antes da execução do caso de uso>
Pós-condição	<condição verdadeira após execução do caso de uso finalizar>
Pontos de Extensão	<lista dos pontos de extensão do caso de uso>
Pontos de Inclusão	<lista dos pontos de inclusão no caso de uso>

Os fluxos alternativos, sub-fluxos, extensões, e inclusões devem ser representados da seguinte forma:

<nº relacionado ao passo do fluxo básico><A (alternativo), S(sub-fluxo), E(extensão), I (inclusão)><nº seqüencial>. <descrição>

- **Modelo de casos de uso atualizado (#2A):** este modelo deverá ser gerado no final do processo com os elementos e especificações de casos de uso adicionais.

Nome do Artefato: Especificação de requisitos não-funcionais (#3)**Descrição:** descrição dos requisitos não-funcionais e suas respectivas prioridades.**Padrão Adotado:** a descrição dos requisitos deve seguir as diretrizes da IEEE 830 (IEEE Std.830, 1998).**Regras de Preenchimento:** identificar cada um dos requisitos com número seqüencial, incluir seu texto e prioridade.

Id	Requisito	Prioridade
<nº seqüencial>	<texto do Requisito>	<prioridade do requisito>

Nome do Artefato: Glossário (#4)**Descrição:** o glossário deve estabelecer os termos essenciais utilizados no contexto do sistema de software a ser desenvolvido.**Padrão Adotado:** padrão de glossário conforme definido em Rational University (2003).**Regras de Preenchimento:** a tabela possui as seguintes colunas:

- **Termo:** preencher com a identificação do termo utilizado no sistema de software.
- **Definição:** preencher com a definição do termo sob o ponto de vista das partes interessadas no desenvolvimento do sistema de software.

Termo	Definição
<nome do termo>	<definição>

Nome do Artefato: *Softgoals* (#5)**Descrição:** conjunto de *softgoal* definidos para o sistema de software.**Padrão Adotado:** conforme Chung et al. (2000).**Regras de Preenchimento:** preencher o *softgoal* com o tipo do requisito não funcional, conforme exemplo: **Nome do Artefato:** Lista de interesses (#6)**Descrição:** lista de interesses do sistema de software.**Padrão Adotado:** não aplicável.**Regras de Preenchimento:** a lista de interesses possui as seguintes colunas:

- **Interesse:** incluir o nome do interesse relacionado a cada um dos *softgoals* e casos de uso.
- **Natureza:** preencher com a natureza do interesse, sendo que será funcional se o interesse for proveniente do modelo de casos de uso, e não-funcional se o interesse for proveniente da definição de um *softgoal*.

Interesse	Natureza
<nome do interesse>	<funcional/não-funcional>

Nome do Artefato: Tabela de requisitos não-funcionais (#7), tabela de requisitos não-funcionais com propagação (#7A), tabela de requisitos não-funcionais com aspectos (#7B), tabela de requisitos não-funcionais com metas operacionais e restrições de projeto (#7C), tabela de requisitos não-funcionais com composição (#7D), tabela de requisitos não-funcionais com prioridades (#7E)

Descrição: tabela que relaciona informações dos requisitos não-funcionais ao longo da execução do PARNAFOA. A tabela de requisitos não-funcionais deve ser construída ao longo do processo, sendo que novas colunas devem ser adicionadas conforme as atividades forem sendo executadas.

Padrão Adotado: não aplicável.

Regras de Preenchimento:

- **Tabela de requisitos não-funcionais (Tabela de RNF) (#7):** a tabela de requisitos não-funcionais deve ser elaborada a partir da especificação de requisitos não-funcionais. Esta tabela possui as seguintes colunas:
 - **Id., requisito e prioridade:** provenientes da especificação de requisitos não-funcionais.
 - **Interesse:** nome do interesse relacionado ao requisito não-funcional;
 - **Origem do Interesse:** origem da classificação do interesse, podendo ser NFR (proveniente no catálogo de tipos de requisitos não-funcionais), ISO/IEC 9126 (características e sub-características de qualidade) ou Lista de Chung (proposta de lista de requisitos não-funcionais apresentada em Chung et al. (2000)).

Id	Requisito	Prioridade	Interesse	Origem do Interesse
<nº sequencial>	<texto do RNF>	<prioridade do RNF>	<nome do interesse>	<NFR/ISO IEC 9126/Lista Chung>

- **Tabela de requisitos não-funcionais com propagação (Tabela de RNF com propagação) (#7A):** descreve as propagações dos requisitos não-funcionais através dos elementos do diagrama de casos de uso. Esta tabela possui as seguintes colunas adicionais:
 - **Ponto de Associação:** descrição do SIG inicial com a estrutura tipo [tópico].
 - **Propagação:** lista das propagações identificadas para o requisito não-funcional.

Ponto de Associação	Propagação
<SIG inicial>	<lista das propagações>

- **Tabela de requisitos não-funcionais com aspectos (Tabela de RNF com aspectos) (#7B):** identifica se um requisito não-funcional é um aspecto. Esta tabela possui a seguinte coluna adicional:
 - **Aspecto:** identifica se o requisito não-funcional é um aspecto (sim/não).

Aspecto
<Sim/Não>

- **Tabela de requisitos não-funcionais com meta operacional ou restrição de projeto (Tabela de RNF com meta operacional ou restrição de projeto) (#7C):** classifica a meta definida para um requisito não-funcional como meta operacional

ou restrição de projeto.

- **Id SIG:** identificação do SIG utilizado para decompor o requisito não-funcional. Este identificador deve possuir o seguinte formato: SIG <nº seqüencial>.
- **Meta:** texto da meta relacionada ao requisito não-funcional.
- **Meta Operacional ou Restrição de Projeto:** classificação da meta como: meta operacional (novo elemento a ser adicionado no modelo de caso de uso) ou restrição de projeto.
- **Origem da Meta:** identificação da origem da meta como: NFR, Tática ou Definida pelo Engenheiro de Requisitos.
- **Descrição:** identificação do novo elemento a ser adicionado ao modelo de casos de uso, no caso de meta operacional ou da restrição, quando tratar-se de restrição de projeto.

Id SIG	Meta	Meta Operacional ou Restrição de Projeto	Origem da Meta	Descrição
<id do SIG que detalha o SIG inicial>	<texto da meta >	<meta operacional/ restrição de projeto>	<identificar a origem da meta >	<descrição da meta operacional ou restrição de projeto>

- **Tabela de requisitos não-funcionais com composição (Tabela de RNF com composição) (#7D):** define informações necessárias para a composição dos novos casos de uso identificados nas metas operacionais.
 - **Caso de Uso:** identificação do caso de uso que irá incluir o novo caso de uso.
 - **Ponto de Composição:** identificação do(s) ponto(s) de composição no fluxo de eventos do caso de uso onde o novo caso de uso será incluído.
 - **Regra de Composição:** identificação do tipo de regra de composição a ser aplicada no ponto de composição. Os tipos a serem selecionados são: sobreposição.depois, sobreposição.antes, substituição ou empacotamento.
 - **Condições da Composição:** definição das condições que devem ser estabelecidas em caso de utilização das regras de substituição ou empacotamento. Para a regra de composição de sobreposição esta coluna deve ser preenchida com “Nenhuma”.

Caso de Uso	Ponto de Composição	Regra de Composição	Condições da Composição
<nome do caso de uso>	<id do passo do cenário do caso de uso>	<definição da regras de composição>	<incluir as condições em caso de substituição e empacotamento>

- **Tabela de requisitos não-funcionais com prioridades (Tabela de RNF com prioridades) (#7E):** descreve a prioridade de composição para casos onde ocorrerem conflitos.
 - **Prioridade:** definição a prioridade da composição para cada caso de uso que possuir a mesma regra de composição no mesmo ponto de composição. Os valores menores possuem a maior prioridade. Caso não exista conflito, esta coluna deve ser preenchida com “Nenhuma”.

Prioridade da Composição
<identificar a prioridade de composição>

O preenchimento das tabelas de RNF com composição e com prioridades somente é aplicável quando existirem metas operacionais. Em caso de restrição de projeto todas as colunas devem ser preenchidas com “NA”.

Nome do Artefato: Diagrama integrado (#8)

Descrição: contém o diagrama de casos de uso relacionado aos *softgoals* do sistema de software.

Padrão Adotado: conforme descrito no item 2.3.2 (abordagem orientada a *softgoals* e casos de uso).

Regras de Preenchimento: conforme descrito no item 2.3.2. Um exemplo do diagrama é apresentado a seguir.

Nome do Artefato: SIG inicial (#9), SIG decomposto (#9A), SIG com prioridades (#9B), SIG com operacionalizações (#9C), SIG com correlações (#9D), SIG com decisões (#9E), SIG com alternativas (#9F), SIG avaliado (#9G), SIG (#9H)

Descrição: decomposição dos requisitos não-funcionais através do grafo de interdependência de metas (SIG), em seus vários estágios.

Padrão Adotado: conforme Chung et al. (2000).

Regras de Preenchimento:

- **SIG inicial (#10):** os SIGs iniciais são gerados no momento em que um relacionamento entre um elemento do diagrama de casos de uso e um *softgoal* é realizado. A ferramenta StarUML preenche automaticamente o tópico dos relacionamentos com atores e casos de uso. Para os relacionamentos de limite do sistema e relação ator-caso de uso, o tópico do *softgoal* deve ser preenchido manualmente.
- **SIG decomposto (#10A):** SIG com *softgoals* detalhados a partir dos catálogos de decomposição.
- **SIG com prioridades (#10B):** SIG com prioridades registradas.
- **SIG com operacionalizações (#10C):** SIG com operacionalizações refinadas e registradas.
- **SIG com correlações (#10D):** SIG com correlações registradas.
- **SIG com decisões (#10E):** SIG com decisões registradas através de *softgoals* de alerta.
- **SIG com alternativas (#10F):** o SIG com as alternativas identificadas através dos rótulos dos *softgoals* de operacionalizações, como as opções: suficientemente satisfeito (√) e rejeitado (X).
- **SIG avaliado (#10G):** o SIG avaliado como resultado da aplicação do procedimento de avaliação.
- **SIG (#10H):** o SIG resultante com as metas de operacionalização identificadas.

APÊNDICE C – Descrição das Atividades

Nome da Atividade e Identificação: Verificar Documentação de Entrada (1.1)
Objetivo: verificar se a documentação de entrada elaborada pelo elaborador de requisitos está de acordo com os padrões definidos.
Entrada(s): Documentos existentes (opcional) (#1) Modelo de casos de uso (#2) Especificação de requisitos não-funcionais (#3) Glossário (#4)
Procedimento: 1. Verificar padrão do modelo de casos de uso, conforme descrição do artefato #2. 2. Verificar padrão da especificação de requisitos não-funcionais, conforme descrição do artefato #3. 3. Verificar padrão do glossário, conforme descrição do artefato #4. 4. Caso algum item não esteja compatível com o padrão estabelecido, solicitar revisão da documentação para adequação.
Métodos e Ferramentas: não aplicável.
Saída(s): Documentos existentes (opcional) (#1) Modelo de casos de uso (#2) Especificação de requisitos não-funcionais (#3) Glossário (#4)

Nome da Atividade e Identificação: Identificar Interesses Funcionais e Não-Funcionais (2.1)
Objetivo: identificar interesses funcionais e não-funcionais do sistema de software.
Entrada(s): Modelo de casos de uso (#2) Especificação de requisitos não-funcionais (#3)
Procedimento: 1. Consultar o catálogo de tipos de requisitos não-funcionais, características e sub-características de qualidade de software da ISO/IEC 9126 e a lista proposta em Chung et al. (2000) a fim de obter a classificação mais adequada para um conjunto de requisitos não-funcionais do mesmo tipo. 2. Gerar as representações dos <i>softgoals</i> a serem utilizadas. 3. Preencher a lista de interesses funcionais e não-funcionais, com os interesses funcionais (casos de uso) e interesses não-funcionais (<i>softgoals</i>). 4. Preencher a tabela de RNF com o interesse relacionado a cada um dos requisitos não-funcionais.
Métodos e Ferramentas: Ferramenta StarUML Catálogo de tipos de requisitos não-funcionais Características de qualidade de software da ISO/IEC 9126 Lista de requisitos não-funcionais proposta por Chung et al. (2000)
Saída(s): <i>Softgoals</i> (#5) Lista de Interesses (#6) Tabela de RNF (#7)

Nome da Atividade e Identificação: Integrar Interesses Funcionais e Não-Funcionais (2.2)
Objetivo: associar os interesses funcionais e não-funcionais no diagrama de casos de uso.
Entrada(s): <i>Softgoals</i> (#5) Lista de Interesses (#6) Tabela de RNF (#7) Modelo de casos de uso (#2)
Procedimento: <ol style="list-style-type: none"> 1. Com base no entendimento dos requisitos não-funcionais associar os <i>softgoals</i> aos elementos do diagrama de casos de uso. 2. Como um <i>softgoal</i> pode representar mais de um requisito não-funcional, um mesmo <i>softgoal</i> poderá ser associado a um ou mais elementos do diagrama de casos de uso. 3. Para associar elementos de limite de sistema, o tópico do SIG inicial deve ser preenchido manualmente, com a sigla do nome do sistema de software. 4. Para associar elementos de relação ator-caso de uso, o tópico do SIG inicial deve ser preenchido manualmente com o padrão <nome do ator>-<nome do caso de uso>. 5. Para as associações com ator e casos de uso, a ferramenta StarUML preenche o tópico do SIG inicial automaticamente.
Métodos e Ferramentas: Ferramenta StarUML Regras de associação detalhadas conforme definido na seção 4.4 deste trabalho.
Saída(s): Diagrama integrado (#8) SIGs iniciais (#9)

Nome da Atividade e Identificação: Identificar Propagações (3.1)
Objetivo: identificar propagações dos requisitos não-funcionais através dos elementos do diagrama de casos de uso.
Entrada(s): Diagrama integrado (#8) SIGs iniciais (#9) Especificação de requisitos não-funcionais (#3)
Procedimento: <ol style="list-style-type: none"> 1. Identificar as propagações relacionadas a cada um dos requisitos não-funcionais com base nas regras de propagação e no entendimento da especificação de requisitos não-funcionais. 2. Preencher as colunas: ponto de associação e propagação da tabela de RNF com propagação para cada um dos requisitos não-funcionais.
Métodos e Ferramentas: Regras de propagação detalhadas na seção 4.5 deste trabalho. Planilha Excel
Saída(s): Tabela de RNF com propagação (#7A)

Nome da Atividade e Identificação: Identificar Aspectos (3.2)
Objetivo: identificar se os requisitos não-funcionais são ou não aspectos.
Entrada(s): Tabela de RNF com propagação (#7A)
Procedimento: <ol style="list-style-type: none"> 1. Aplicar as heurísticas definidas para identificação de aspectos para cada um dos requisitos não-funcionais. 2. Preencher a coluna aspecto da tabela de RNF com aspectos para cada um dos requisitos não-funcionais.
Métodos e Ferramentas: Heurísticas de identificação de aspectos definidas na seção 4.5 deste trabalho. Planilha Excel
Saída(s): Tabela de RNF com aspectos (#7B)

Nome da Atividade e Identificação: Selecionar SIG Inicial a ser Detalhado (4.1)
Objetivo: selecionar o SIG inicial a ser detalhado a partir de prioridades definidas na especificação de requisitos não-funcionais.
Entrada(s): Diagrama integrado (#8) Especificação de requisitos não-funcionais (#3)
Procedimento: <ol style="list-style-type: none"> 1. Consultar as prioridades definidas para os requisitos não-funcionais na especificação de requisitos não-funcionais. 2. Selecionar o requisito não-funcional com maior prioridade. 3. Selecionar o SIG inicial relacionado ao requisito não-funcional de maior prioridade. 4. Caso o mesmo SIG inicial esteja relacionado a mais de um requisito não-funcional, considerar o conjunto de requisitos no momento da decomposição. 5. Se for necessário, utilizar mais de um SIG para decompor os requisitos não-funcionais.
Métodos e Ferramentas: não aplicável.
Saída(s): SIG inicial (#9)

Nome da Atividade e Identificação: Decompor <i>Softgoals</i> (4.2)
Objetivo: decompor os SIGs iniciais.
Entrada(s): SIG inicial (#9) Tabela de RNF com aspectos (#7B) Especificação de requisitos não-funcionais (#3)
Procedimento: <ol style="list-style-type: none"> 1. Consultar a coluna origem do interesse a fim de identificar o tipo de detalhamento a ser aplicado. Se a escolha tiver sido NFR <i>Framework</i> utilizar os sub-tipos de requisitos presentes nos catálogos de tipos de requisitos não-funcionais. 2. Se a escolha tiver sido ISO/IEC 9126 utilizar as sub-categorias de qualidade de software para o detalhamento de sub-tipo de requisito não-funcional. 3. Se a escolha tiver sido a Lista Chung (CHUNG et al., 2000), detalhar o tipo de

<p>requisito de forma mais adequada.</p> <ol style="list-style-type: none"> 4. Representar opcionalmente no SIG o(s) tipo(s) de requisito não-funcional de mais alto nível antes de iniciar a sua decomposição, a fim de deixar a decomposição mais clara. 5. Utilizar os métodos de decomposição de limite de sistema, casos de uso, ator e relação ator-caso de uso para registrar as propagações identificadas para os requisitos não-funcionais. Estas propagações devem ser consultadas na tabela de RNF com aspectos para cada um dos requisitos não-funcionais a serem representados no SIG. 6. Utilizar o catálogo de decomposição de métodos do NFR <i>Framework</i> para refinar os <i>softgoals</i>.
<p>Métodos e Ferramentas: Ferramenta StarUML Catálogo de decomposição de tipos de requisitos não-funcionais Sub-categorias de qualidade de software da ISO/IEC 9126 Lista de Chung Catálogo de decomposição de métodos Métodos de decomposição de limite de sistema, casos de uso, ator e relação ator-caso de uso definidas na seção 4.6.1 deste trabalho.</p>
<p>Saída(s): SIG decomposto (#9A)</p>

Nome da Atividade e Identificação: Definir Prioridades (4.3)
Objetivo: registrar as prioridades dos requisitos não-funcionais no SIG.
Entrada(s): Especificação de requisitos não-funcionais (#3)
Procedimento: <ol style="list-style-type: none"> 1. Com base nas informações da especificação de requisitos não-funcionais registrar prioridades, conforme necessidade do sistema de software.
Métodos e Ferramentas: Ferramenta StarUML
Saída(s): SIG com prioridades (#9B)

Nome da Atividade e Identificação: Identificar Operacionalizações (4.4)
Objetivo: identificar operacionalizações no SIG.
Entrada(s): SIG decomposto (#9A)
Procedimento: <ol style="list-style-type: none"> 1. Consultar o catálogo de operacionalizações e a lista de táticas a fim de identificar o <i>softgoal</i> de operacionalização mais adequado para o SIG decomposto. 2. Caso não existam operacionalizações adequadas nos catálogos de operacionalizações e nas táticas, definir o <i>softgoal</i> de operacionalização que considerar mais adequado. 3. Registrar a(s) os <i>softgoals</i> de operacionalização(ões) no SIG decomposto.
Métodos e Ferramentas: Ferramenta StarUML Catálogo de operacionalizações

Táticas
Saída(s): SIG com operacionalizações (#9C)

Nome da Atividade e Identificação: Identificar Correlações (4.5)
Objetivo: identificar correlações positivas e negativas no SIG.
Entrada(s): SIG com operacionalizações (#9C)
Procedimento: <ol style="list-style-type: none"> 1. Consultar o catálogo de correlações do NFR <i>Framework</i> e as descrições das táticas. 2. Registrar correlações positivas e negativas entre <i>softgoals</i>. 3. Caso se detecte alguma correlação não presente nos catálogos, esta deve ser representada no SIG.
Métodos e Ferramentas: Ferramenta StarUML Catálogo de correlações Táticas
Saída(s): SIG com correlações (#9D)

Nome da Atividade e Identificação: Registrar Decisões de Projeto (4.6)
Objetivo: registrar decisões de projeto através dos <i>softgoals</i> de afirmação.
Entrada(s): SIG com correlações (#9D)
Procedimento: <ol style="list-style-type: none"> 1. Consultar o catálogo de argumentação. 2. Registrar <i>softgoals</i> de afirmação relacionados a cada SIG conforme decisões de projeto.
Métodos e Ferramentas: Ferramenta StarUML Catálogo de argumentação
Saída(s): SIG com decisões (#9E)

Nome da Atividade e Identificação: Selecionar Alternativas de Projeto (4.7)
Objetivo: identificar dentre o conjunto <i>softgoals</i> de operacionalizações aqueles que serão selecionados ou rejeitados.
Entrada(s): SIG com decisões (#9E)
Procedimento: <ol style="list-style-type: none"> 1. Identificar <i>softgoals</i> de operacionalizações que serão selecionados (✓). 2. Identificar <i>softgoals</i> de operacionalizações que serão rejeitados (X). 3. Registrar as decisões no SIG.
Métodos e Ferramentas: Ferramenta StarUML
Saída(s): SIG com alternativas (#9F)

Nome da Atividade e Identificação: Aplicar o Procedimento de Avaliação (4.8)
Objetivo: aplicar o procedimento de avaliação a fim de definir o grau de satisfação dos <i>softgoals</i> .
Entrada(s): SIG com alternativas (#9F)
Procedimento: <ol style="list-style-type: none"> 1. Aplicar o primeiro passo do procedimento de avaliação (avaliação individual) com base no catálogo de avaliação. 2. Aplicar o segundo passo do procedimento de avaliação (avaliação conjunta) com base no catálogo de avaliação. 3. Registrar as avaliações no SIG.
Métodos e Ferramentas: Ferramenta StarUML Catálogos de avaliação (CHUNG et al., 2000)
Saída(s): SIG avaliado (#9G)

Nome da Atividade e Identificação: Definir o Relacionamento de Metas de Operacionalização (4.9)
Objetivo: definir o conteúdo das metas e os relacionamentos de metas de operacionalização.
Entrada(s): SIG avaliado (#9G)
Procedimento: <ol style="list-style-type: none"> 1. A partir dos <i>softgoals</i> de operacionalizações selecionados, definir o conteúdo das metas. 2. Relacionar a meta aos seus respectivos <i>softgoals</i> de operacionalização através do relacionamento de meta de operacionalização. 3. Cada um dos requisitos não-funcionais deverá possuir apenas uma meta. 4. Caso mais de uma meta seja necessária para um mesmo requisito, recomenda-se o desmembramento do requisito não-funcional. A partir deste ponto o processo inicia-se novamente com a revisão deste requisito, no sub-processo 1, atividade 1.1 (verificar documentação de entrada). 5. Caso o mesmo SIG contemple a decomposição de mais de um requisito não-funcional, uma meta deverá ser definida para cada um dos requisitos. 6. O identificador do requisito não-funcional deve ser incluído no texto da meta, para fins de rastreabilidade. 7. A meta não deverá ser relacionada ao requisito funcional, conforme recomendação do NFR <i>Framework</i>, pois este relacionamento já está explícito no diagrama integrado, quando um <i>softgoal</i> é associado a um elemento do modelo de casos de uso.
Métodos e Ferramentas: Ferramenta StarUML
Saída(s): SIG (#9H)

Nome da Atividade e Identificação: Identificar Metas de Operacionalização e Restrições de Projeto (5.1)
Objetivo: identificar metas de operacionalizações que serão transformadas em metas operacionais ou restrições de projeto.
Entrada(s): SIGs (#9H) Tabela de RNF com aspectos (#7B)
Procedimento: <ol style="list-style-type: none"> 1. Preencher a coluna meta da tabela de propagação com o texto da meta relacionado a cada um dos requisitos não-funcionais. 2. Preencher a coluna meta operacional ou restrição de projeto com a classificação do tipo da meta. 3. Preencher a coluna origem da meta, que pode ser NFR <i>Framework</i>, tática ou definida pelo engenheiro de requisitos. 4. Em caso de meta classificada como meta operacional, preencher a coluna descrição com o novo caso de uso a ser adicionado ao modelo de casos de uso. 5. Em caso a meta classificada como restrição de projeto, preencher a coluna descrição com a restrição a ser imposta ao sistema de software. 6. Nunca relacionar o mesmo caso de uso a duas ou mais metas operacionais. Isso garante que controles adicionais não precisem ser implementados para separar diferentes requisitos que estiverem dentro de um mesmo caso de uso.
Métodos e Ferramentas: Planilha Excel
Saída(s): Tabela de RNF com metas operacionais e restrições de projeto (#7C)

Nome da Atividade e Identificação: Definir Pontos de Composição (5.2)
Objetivo: definir os pontos e regras de composição.
Entrada(s): Tabela de RNF com metas operacionais e restrições de projeto (#7C) Modelo de casos de uso (#2)
Procedimento: <ol style="list-style-type: none"> 1. Aplicar este procedimento somente quando uma meta for classificada como meta operacional. Restrições de projeto não possuem pontos de composição, somente restrições a serem impostas ao sistema de software. 2. Para cada meta preencher as colunas: caso de uso e ponto de composição. 3. Definir a regra de composição e preencher a coluna regra de composição. 4. Preencher a coluna condições da composição quando utilizadas as regras de composição por substituição ou empacotamento.
Métodos e Ferramentas: Planilha Excel
Saída(s): Tabela de RNF com composição (#7D)

Nome da Atividade e Identificação: Confirmar Propagação (5.3)
Objetivo: confirmar propagação após definição dos pontos de composição.
Entrada(s): Tabela de RNF com composição (#7D)
Procedimento: <ol style="list-style-type: none"> 1. Analisar se durante a identificação dos pontos de composição novas propagações foram identificadas, ou alteradas. 2. Caso seja necessário revisar as propagações, o processo deve reiniciar a partir do sub-processo 3, atividade 3.1 (identificar propagações). 3. A tabela de RNF com composição e o SIG devem ser revisados. 4. Caso seja necessário incluir propagações entre os casos de usos novos, a nova propagação deve ser identificada na coluna propagação. 5. Se não for necessário revisar as propagações, o processo continua na atividade 5.4.
Métodos e Ferramentas: não aplicável.
Saída(s): Tabela de RNF com composição (#7D) SIG (#9H)

Nome da Atividade e Identificação: Resolver Conflitos (5.4)
Objetivo: resolver conflitos a partir da tabela de RNF com composição.
Entrada(s): Tabela de RNF com composição (#7D)
Procedimento: <ol style="list-style-type: none"> 1. Aplicar filtros na tabela de RNF com composição selecionando os campos: caso de uso, ponto de composição e regra de composição. 2. Caso o filtro apresente os mesmos pontos de composição com as mesmas regras de composição para um mesmo caso de uso, prioridades para a composição devem ser estabelecidas. 3. As prioridades devem ser estabelecidas conforme entendimento dos requisitos não-funcionais. 4. Quanto menor o valor atribuído, maior será a prioridade para a composição.
Métodos e Ferramentas: Planilha Excel
Saída(s): Tabela de RNF com prioridades (#7E)

Nome da Atividade e Identificação: Atualizar Modelo do Sistema de Software (5.5)
Objetivo: atualização do modelo do sistema de software com as informações obtidas na composição.
Entrada(s): Tabela de RNF com prioridades (#7E)
Procedimento: <ol style="list-style-type: none"> 1. Com base na tabela RNF com prioridades identificar os casos de uso a serem incluídos no modelo de casos de uso. 2. Atualizar o diagrama de casos de uso com os novos casos de uso. 3. Quando um novo caso de uso estiver relacionado a um requisito não-funcional identificado como aspecto o estereótipo <<aspecto>> deve ser atribuído ao caso de uso.

4. Não utilizar estereótipo caso o requisito não-funcional relacionado ao caso de uso não for considerado aspecto.
5. Criar a especificações para os novos casos de uso.
6. Preencher o campo requisitos especiais dos novos casos de uso com o identificador do requisito não-funcional que deu origem ao caso de uso.
7. Caso tenha ocorrido inter-relacionamento entre os requisitos não-funcionais, e conseqüentemente entre os novos casos de uso, a tabela de RNF com prioridades deve ser revisada para que os campos: caso de uso, ponto de composição, regra de composição e condições da composição sejam preenchidos. Uma revisão das prioridades deve ser realizada.
8. Se necessário deve ser revisado o diagrama integrado, com seus SIGs iniciais.

Métodos e Ferramentas:

Ferramenta StarUML

Editor de texto

Saída(s):

Modelo de casos de uso atualizado (#2A)

Tabela de RNF com prioridades (#7E)

Diagrama integrado (#8)