

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**TRANSFORMAÇÃO DE MODELOS DE
PROCESSOS DE NEGÓCIO EM BPMN
PARA MODELOS DE SISTEMA
UTILIZANDO CASOS DE USO DA UML**

LEONARDO RODRIGUEZ HEREDIA

Dissertação apresentada como requisito parcial à
obtenção do grau de Mestre em Ciência da
Computação na Pontifícia Universidade Católica do
Rio Grande do Sul

Orientador: Prof. Dr. Ricardo Melo Bastos

Porto Alegre
Janeiro de 2012

H542t

Heredia, Leonardo Rodriguez

Transformação de modelos de processos de negócio em BPMN para modelos de sistema utilizando casos de uso da UML / Leonardo Rodriguez Heredia. – Porto Alegre, 2012. 126 f.

Diss. (Mestrado) – Fac. de Informática, PUCRS.
Orientador: Prof. Dr. Ricardo Melo Bastos.

1. Informática. 2. UML (Informática). 3. Modelagem de Sistemas. I. Bastos, Ricardo Melo.
II. Título.

CDD 005.114

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "**Transformação de Modelos de Processos de Negócio em BPMN para Modelos de Sistema Utilizando Casos de Uso da UML**", apresentada por Leonardo Rodriguez Heredia, como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, Sistemas de Informação, aprovada em 26/01/2012 pela Comissão Examinadora:



Prof. Dr. Ricardo Melo Bastos -
Orientador

PPGCC/PUCRS



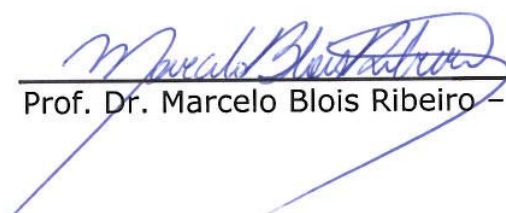
Prof. Dr. Rafael Prikladnicki -

PPGCC/PUCRS



Prof. Dr. José Palazzo Moreira de Oliveira -

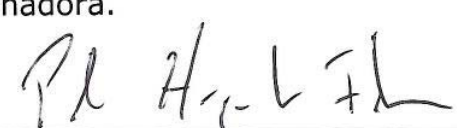
UFRGS



Prof. Dr. Marcelo Blois Ribeiro -

GE - Brasil

Homologada em 08/05/2012, conforme Ata No. 010 pela Comissão Coordenadora.



Prof. Dr. Paulo Henrique Lemelle Fernandes
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 - P32- sala 507 - CEP: 90619-900

Fone: (51) 3320-3611 - Fax (51) 3320-3621

E-mail: ppgcc@pucrs.br

www.pucrs.br/facin/pos

*Dedico esse trabalho
ao meu pai, Carlos, a minha
mãe, Suely, e aos meus
irmãos, Alexandre e Tales.*

AGRADECIMENTOS

Agradeço a Deus por me dar condições para iniciar e concluir esse trabalho. Sem a ajuda Dele eu não teria conseguido.

A minha família por me apoiar e estimular ao longo do mestrado e ao longo da vida! O apoio de vocês é fundamental!

A todos aqueles que me ajudaram e apoiaram durante o curso. Obrigado pelas orações que com certeza foram ouvidas.

Ao meu orientador, o Prof. Ricardo Melo Bastos, pelas orientações e por sempre me transmitir tranquilidade e confiança de que o trabalho estava no caminho correto.

Ao Prof. Marcelo Blois por acompanhar o trabalho e contribuir com sugestões e orientações para o mesmo.

À HP Brasil, em especial aos meus gerentes no decorrer do curso, que permitiram a flexibilidade de horário em diversos momentos.

Aos professores e funcionários Instituto de Informática da PUCRS pelo trabalho de qualidade que é realizado nessa instituição.

Ao convênio HP/PUCRS pelo apoio financeiro para realização do curso.

TRANSFORMAÇÃO DE MODELOS DE PROCESSO DE NEGÓCIO EM BPMN PARA MODELOS DE SISTEMA UTILIZANDO CASOS DE USO DA UML

RESUMO

Os modelos de processos de negócio definidos pelos analistas de negócio servem como fonte de requisitos para a modelagem dos sistemas de informação que podem ser construídos para dar suporte a esses processos. A transformação de um modelo em nível de negócio para um modelo em nível de sistema não é trivial. A notação ou linguagem utilizada para modelar o negócio é diferente da utilizada para modelar sistema. Atualmente a BPMN é a notação padrão para construção de modelos de processos de negócio e a UML é a linguagem padrão para modelagem de sistemas. O objetivo desse trabalho é apresentar um método que auxilie na transformação de modelos de processos de negócio feitos em BPMN para modelos de sistema utilizando casos de uso da UML. A transformação é realizada através da análise de como cada elemento de um diagrama BPMN pode dar origem a elementos de casos de uso da UML. O método apresenta diferentes possibilidades de transformar um elemento da BPMN em casos de uso da UML, identificando os relacionamentos e informações iniciais da descrição dos casos de uso.

Palavras chave: modelagem de sistemas, modelagem de processos de negócio, BPMN, UML, casos de uso, transformação de modelos.

TRANSFORMING BPMN BUSINESS PROCESSES MODELS INTO SYSTEM MODELS USING UML USE CASES

ABSTRACT

The business processes models designed by business analysts act as requirement source to model information systems that may give support to these business processes. Transforming a business process model into a system model is not simple. The notation or language used to design business process model is different from the one used to design the system model. Today, BPMN is the pattern to build process models and UML is the pattern to design system models. The goal of this research is to present a method that helps the transformation business process model created in BPMN to system model using UML use cases. The transformation occurs through analysis of how each business element from a BPMN diagram can result in UML use cases elements. The method presents many alternatives to transform a BPMN element into a UML use case, by identifying the relationships and the initial information of cases descriptions.

Keywords: system modeling, business process modeling, BPMN, UML, use case, model transformation.

LISTA DE FIGURAS

Figura 1 - Representação de Atividades na BPMN	35
Figura 2 - <i>Gateways</i> da BPMN	36
Figura 3 - (a) Evento de início sem causa definida. (b) Evento de início tipo tempo. (c) Evento intermediário do tipo tempo. (d) Evento intermediário de exceção anexado a tarefa. (e) Evento de fim	37
Figura 4 - Objetos de Conexão da BPMN	38
Figura 5 - Partições da BPMN (Participante e Raia)	38
Figura 6 - Artefatos da BPMN	39
Figura 7 - Metamodelo da BPMN (adaptado de [OMG09b]).....	40
Figura 8 - Símbolos para atores e casos de uso usados no diagrama da UML	42
Figura 9 - Exemplo de relacionamento de extensão [BOO05]	44
Figura 10 - Exemplo de relacionamento de inclusão [BOO05].....	45
Figura 11 - Exemplo de relacionamento de generalização [BOO05]	46
Figura 12 – Metamodelo da UML para casos de uso [UML11]	46
Figura 13 - Metamodelo (resumido) de descrição de caso de uso [SOMé09].....	47
Figura 14 - Níveis da MDA [HAR04]	49
Figura 15 - Processo de negócio (traduzido do original) [LIE04]	53
Figura 16 - Diagrama de caso de uso resultante (traduzido do original) [LIE04]	53
Figura 17 - Diagrama de classes obtido a partir do processo de negócio [OKA07].....	54
Figura 18 - Processe de negócio com rótulos [VAR08]	55
Figura 19 - Descrição da tarefa Notify Customer obtida transformando o processo de negócio [VAR08].....	56
Figura 20 - Exemplo de processo de negócio em BPMN	57
Figura 21 - Casos de uso obtidos a partir do processo de negócio	58
Figura 22 - Exemplo de processo de negócio em BPMN	59
Figura 23 - Exemplo de classe gerada a partir da transformação	59
Figura 24 - Diagrama de casos de uso obtido na transformação	59
Figura 25 – Transformação de Raia para Ator.....	69
Figura 26 - Tarefa para Caso de Uso	70
Figura 27 - Tarefa de Serviço para Caso de Uso	70
Figura 28 - Tarefa de Serviço para passo de um caso de uso	71
Figura 29 - Descrição do caso de uso Cadastrar Pedido	71

Figura 30 - <i>Gateway</i> Exclusivo para relacionamentos de extensão.....	73
Figura 31 - <i>Gateway</i> Exclusivo com condição padrão para relacionamento de extensão	74
Figura 32 - Descrição caso de uso Analisar Projeto	74
Figura 33 - <i>Gateway</i> Exclusivo para Relacionamento de Inclusão	75
Figura 34 - <i>Gateway</i> Exclusivo para Casos de Uso com especialização	77
Figura 35 - Exemplo de processo de negócio para registro de novo professor	78
Figura 36 - Transformação de <i>gateway</i> inclusivo para relacionamento de extensão.....	79
Figura 37 - Transformação do <i>gateway</i> inclusivo para relacionamento de inclusão	79
Figura 38 - Exemplo de processo de negócio de realização de pedido.....	81
Figura 39 - <i>Gateway</i> paralelo pra casos de uso com relacionamento de inclusão	81
Figura 40 - <i>Gateway</i> Paralelo para Casos de Uso sem relacionamento entre eles.....	83
Figura 41 - Descrição do caso de uso Imprimir Fatura.....	83
Figura 42 - Transformação de diagrama com evento de início temporal	85
Figura 43 - Descrição do caso de uso	85
Figura 44 - Transformação de processo com evento intermediário condicional	87
Figura 45 - Descrição do caso de uso Confirmar Participação em Curso.....	87
Figura 46 - Processo de negócio utilizando evento intermediário anexado a tarefa.....	88
Figura 47 - Transformação de evento temporal anexado a tarefa para casos de uso	88
Figura 48 - Caso de uso Realizar Check-in	89
Figura 49 - Processo de negócio de Ordem de Compra.....	91
Figura 50 - Diagrama de Casos de Uso obtido na transformação.....	91
Figura 51 - Descrição do caso de uso Gerar Ordem de Compra utilizando uma pós-condição	92
Figura 52 - Descrição do caso de uso Enviar Ordem de Compra contendo uma pré-condição.....	92
Figura 53 - Processo de negócio utilizando elemento grupo	95
Figura 54 - Caso de uso originado pelo processo contendo o elemento grupo.....	95
Figura 55 - Descrição do caso de uso Submeter Ordem de Compra	95
Figura 56 - Processo de negócio com grupo e <i>gateway</i>	96
Figura 57 - Diagrama de caso de uso originado da transformação do processo com grupo e <i>gateway</i>	96
Figura 58 - Descrição do caso de uso Cadastrar nova conta.....	97
Figura 59 - Processo de Solicitação de Material Interno	102
Figura 60 - Diagrama de Casos de Uso obtido a partir das tarefas da raia do Enfermeiro Chefe ...	103
Figura 61 - Diagrama de Casos de Uso obtido a partir da raia do Gestor	104
Figura 62 - Descrição do caso de uso Verificar Quantidade de Material Interno.....	104
Figura 63 - Descrição do caso de uso Realizar Pedido de Material Interno	105

Figura 64 - Descrição do caso de uso Enviar Pedido de Material Interno.....	105
Figura 65 - Descrição do caso de uso Avaliar Pedido de Aprovação.....	105
Figura 66 - Descrição do caso de uso Notificar Reprovação por Prazo	106
Figura 67 - Descrição do caso de uso de Informar Dados para Aprovação.....	106
Figura 68 - Descrição do caso de uso de Justificar Reprovação.....	106
Figura 69 - Descrição (após refinamento) do caso de uso Avaliar Pedido de Aprovação.....	107
Figura 70 - Processo de Negócio para Solicitação de Reembolso de Despesas de Viagem.....	109
Figura 71 - Casos de uso do ator Funcionário	110
Figura 72 - Casos de uso do ator Gerente	111
Figura 73 - Casos de uso do ator Financeiro.....	112
Figura 74 - Descrição do caso de uso Cadastrar Pedido de Reembolso de Viagem.....	112
Figura 75 - Descrição do caso de uso Informar Despesas de Veículo.....	113
Figura 76 - Descrição do caso de uso Enviar Email para o Gerente.....	113
Figura 77 - Descrição do caso de uso Analisar Relatório de Despesas	114
Figura 78 - Descrição do caso de uso Informar Dados de Aprovação.....	114
Figura 79 - Descrição do caso de uso Informar Motivos Reprovação.....	114
Figura 80 - Descrição do caso de uso de Autorizar Pagamento.....	115
Figura 81 - Descrição do caso de uso Liberar Pagamento.....	115
Figura 82 - Descrição do caso de uso Informar Motivos de Reprovação do Pagamento	115
Figura 83 - Descrição do caso de uso Efetuar Transferência.....	116

LISTA DE TABELAS

Tabela 1 - Regras de mapeamento de conceitos de negócio para casos de uso.....	52
Tabela 2 - Formato do caso de uso	65
Tabela 3 - Elementos da BPMN analisados.....	67
Tabela 4 - Passos genéricos para aplicação do método	101

LISTA DE SIGLAS

B2B	<i>Business to Business</i>
BPMI	<i>Business Process Management Institute</i>
BPMN	<i>Business Process Modeling Notation</i>
CIM	<i>Computing Independent Model</i>
OMG	<i>Object Management Group</i>
PIM	<i>Platform Independent Model</i>
PSM	<i>Platform Specific Model</i>
RUP	<i>Rational Unified Process</i>
QVT	<i>Query-View Transformation</i>
SGPN	Sistemas de Gerenciamento de Processos de Negócio
UML	<i>Unified Modeling Language</i>
WfMC	<i>Workflow Management Coalition</i>

SUMÁRIO

1.	Introdução	27
1.1	Motivação.....	29
1.2	Questão de pesquisa	29
1.3	Objetivos	30
1.3.1	Objetivo Geral.....	30
1.3.2	Objetivos específicos	30
1.4	Metodologia	30
1.5	Estrutura da Dissertação.....	31
2.	Modelagem de Processo de Negócio	33
2.1	Conceitos básicos sobre modelagem de processos de negócio	33
2.2	Business Process Modeling Notation (BPMN).....	34
2.2.1	Objetos de Fluxo	35
2.2.2	Objetos de Conexão	38
2.2.3	Partições (Swinlanes)	38
2.2.4	Artefatos	39
2.3	Metamodelo da BPMN	39
3.	Modelagem de Sistemas, MDA e Transformações de Modelos.....	41
3.1	Casos de uso e a UML	41
3.2	Modelos de casos de uso	42
3.3	Descrição de Casos de Uso	42
3.4	Relacionamento entre casos de uso.....	43
3.4.1	Relacionamento de extensão.....	44
3.4.2	Relacionamento de Inclusão	45
3.4.3	Relacionamento de Generalização	45
3.5	Metamodelo de casos de uso.....	46
3.6	MDA	48
3.7	Transformações entre modelos	49
4.	Trabalhos Relacionados	51
4.1	Deriving Use Case Diagrams from Business Process Models (2002)	51
4.2	A framework for business model driven development (2004)	52

4.3	A Method of Linking Business Process Modeling with Information System Design Using UML and its Evaluation (2007)	53
4.4	Improving Requirements Analysis through Business Process Modelling: A Participative Approach (2008)	55
4.5	Patrones para la Extracción de Casos de Uso a partir de Procesos de Negocio (2009)	56
4.6	Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach (2010)	58
4.7	Considerações sobre os trabalhos relacionados	60
5	Método para transformação de BPMN em Casos de Uso da UML	63
5.1	Estrutura do caso de uso resultante	64
5.1.1	Formato do detalhamento do caso de uso	64
5.2	Verificação do modelo de negócio a ser transformado	66
5.2.1	Elementos suportados	66
5.3	Transformação dos elementos da BPMN para casos de uso	67
5.3.1	Participante e Raia	68
5.3.2	Tarefa Manual	69
5.3.3	Tarefa de Usuário	69
5.3.4	Tarefa de Serviço	70
5.3.5	Gateway	72
5.3.5.1	Gateway Exclusivo	72
5.3.5.2	Gateway Inclusivo	77
5.3.5.3	Gateway Paralelo	80
5.3.6	Evento	83
5.3.6.1	Evento Inicial	84
5.3.6.2	Evento Intermediário	86
5.3.6.3	Evento Final	89
5.3.7	Artefatos	90
5.3.7.1	Objeto de Dados	90
5.3.7.2	Grupo	93
5.4	Considerações finais sobre o método proposto	98
5.4.1	Análise de diferentes possibilidades de transformação	98
5.4.2	Elementos da descrição dos Casos de Uso	99
5.4.3	Transformações dos eventos	99
5.4.4	Transformação dos artefatos	99

5.4.5	Referência aos metamodelos.....	99
6	Aplicação do método proposto	101
6.1	Passos Genéricos para Aplicação do Método	101
6.2	Aplicação 1 – Processo de solicitação de Material Interno em um hospital....	102
6.2.1	Realização da Transformação	102
6.2.2	Considerações sobre o resultado obtido.....	107
6.3	Aplicação 2 – Processos para pedido de reembolso de despesas de viagem ...	108
6.3.1	Realização da Transformação	109
6.3.2	Considerações sobre o resultado obtido.....	116
7	Considerações Finais	119
7.1	Limitações do Estudo.....	120
7.2	Trabalhos Futuros	120
	Referências Bibliográficas	123

1. INTRODUÇÃO

Esse trabalho tem por finalidade apresentar um método que auxilie na transformação de modelos de processos de negócio expressos usando a *Business Process Modeling Notation* (BPMN) para modelos de sistemas baseados em casos de uso da *Unified Modeling Language* (UML). O método proposto auxilia o analista de sistemas a realizar essa transformação, identificando os casos de uso e seus relacionamentos.

Atualmente as organizações necessitam aperfeiçoar os seus processos de negócio para permanecerem competitivas. Uma das formas de visualizar ou definir esses processos é através da construção de modelos. A modelagem dos processos de negócio ajuda na identificação e entendimento das atividades realizadas por diferentes participantes de uma organização. O processo do negócio consiste em um conjunto de um ou mais procedimentos ou atividades que, coletivamente, realizam um objetivo do negócio dentro de um contexto organizacional [WMC99]. Um modelo de processo de negócio é uma representação visual dos elementos que formam esse processo. Esses elementos podem ser tarefas, participantes, transições, documentos, arquivos, entre outros.

Uma vez identificados e modelados os processos de negócio pode ser necessária a criação de um sistema que automatize determinadas atividades desses processos. O desenvolvimento de um sistema é uma tarefa complexa, sendo que é preciso seguir determinadas etapas antes da sua efetiva programação. As etapas iniciais do desenvolvimento são basicamente a análise de requisitos e a modelagem do sistema. A modelagem do sistema ajuda o analista de sistemas a entender a informação, função e comportamento de um sistema, tornando a análise de requisitos mais fácil e sistemática [PRE06].

Para a modelagem de sistemas a linguagem utilizada como padrão atual na indústria é a UML [UML11]. A UML possui diferentes diagramas que podem ser usados para modelagem dos diversos aspectos do sistema. Dentre esses diagramas pode-se destacar o diagrama de casos de uso. Esse diagrama permite obter uma modelagem da visão dos casos de uso do sistema mostrando os casos de uso, os atores e seus relacionamentos.

A identificação correta dos requisitos de um sistema de informação é um fator determinante para a construção de um sistema que atenda as necessidades dos usuários. Uma das formas de expressar requisitos de sistema é através da utilização de casos de uso [COC05, SOM07]. O conceito de caso de uso proposto por Jaboson [JAC92] e contemplado pela UML [UML11] é utilizado atualmente na indústria de software como principal meio de capturar e documentar os requisitos de um sistema. Um analista de sistemas pode identificar os casos de uso de um sistema

através da análise de diferentes fontes de informação. Entre elas, estão os modelos de processos de negócio relacionados com o sistema que está sendo desenvolvido.

Os modelos de processos de negócio podem servir como fonte de requisitos para o desenvolvimento de um sistema que suporte esses processos. Esses modelos expressam informações fundamentais que devem ser consideradas no sistema. Por isso cada vez mais os desenvolvedores de *software* estão percebendo a necessidade de entender os modelos de processos de negócio propostos pelos analistas de negócio [PHA98, SIN09]. Analisando as informações contidas nos modelos de processo de negócio pode ser possível identificar casos de uso do sistema. Entretanto, não há uma relação direta entre os elementos do modelo no nível do negócio para elementos dos casos de uso, que estão em um nível do sistema. O que se verifica hoje é que existe uma lacuna (um *gap*) [ODE03] entre a modelagem feita no nível do negócio e a modelagem realizada no nível de sistema, sendo necessário realizar uma transformação dos elementos existentes nos modelos de negócio para elementos dos modelos em nível de sistema.

Essa transformação de modelos do nível de negócio para os do nível de sistema vem sendo objeto de diversos estudos na área de Engenharia de Software. Analisando a questão a partir de uma abordagem da Arquitetura Dirigida por Modelos (*Model-driven Architecture*, MDA) [OMG03], verifica-se que os modelos de processos de negócio estão em um nível de Modelo Independente de Computação (*Computing Independent Model*, CIM), enquanto que os modelos de sistemas localizam-se nos níveis de Modelos Independentes de Plataforma (*Platform Independent Model*, PIM) e de Modelos para Plataforma Específica (*Platform Specific Model*, PSM). Sendo assim, a transformação entre um modelo em um nível CIM para um modelo do nível PIM e, posteriormente, de nível PSM, pode ser apoiada usando os princípios de transformação entre modelos propostas pela MDA.

Uma revisão sistemática realizada pelo autor desta pesquisa em 2010 [HER10] com o objetivo de identificar e avaliar abordagens para realizar essa transformação verificou a predominância do uso da BPMN [BPM09] e da UML como notação e linguagem padrão para a modelagem de negócio e sistema, respectivamente. Essa revisão sistemática também apontou que várias propostas existentes não contemplam uma série de elementos da notação BPMN, bem como geram modelos de sistema baseados em casos de uso incompletos.

A finalidade da presente pesquisa é apresentar um método que auxilie o analista de sistemas a modelar o sistema – utilizando casos de uso – a partir de modelos de processos de negócio construídos em BPMN. O método proposto apresenta alternativas de transformação de elementos da BPMN para elementos de casos de uso da UML que podem ser utilizadas pelo analista auxiliando o mesmo no processo de transformação dos modelos.

1.1 MOTIVAÇÃO

As organizações estão cada vez mais buscando agilidade na identificação e definição dos seus processos de negócio, bem como desenvolvendo ferramentas para automatizar e melhorar a execução dos mesmos.

A modelagem dos processos de negócio de uma organização geralmente é realizada por um analista de negócio. Esse profissional se dedica especificamente a identificar, documentar e melhorar os processos de negócio. Uma vez modelados os processos, pode-se partir para uma automatização dos mesmos através do desenvolvimento de um sistema de informação. Esse sistema de informação, por sua vez, geralmente é modelado por outro profissional, o analista de sistema. O analista de sistema é responsável por analisar e documentar os requisitos do sistema, muitas vezes utilizando como base os processos de negócio definidos pelo analista de negócio.

Esses dois profissionais, entretanto, utilizam diferentes notações e linguagens para realizar essas duas tarefas. No contexto da modelagem de negócio o padrão atualmente é o uso da BPMN [HAR10]. Já para modelagem de sistemas o padrão é a utilização da UML [WAT08]. Esses dois profissionais trabalham em níveis de abstrações diferentes e a transformação dos modelos criados no nível de negócio para modelos em nível de sistema se torna difícil justamente pela diferença semântica existente entre os elementos que formam os modelos em cada um desses níveis.

Sendo assim, a proposta de um método que auxilie na transformação de elementos de um modelo em nível de negócio para um modelo em nível de sistema se torna pertinente para a área de Engenharia de Software. O método proposto, neste trabalho, tem seu escopo limitado a transformações de modelos de processo de negócio expressos em BPMN para casos de uso da UML, contemplando tanto os diagramas de casos de uso como os elementos do detalhamento dos mesmos. Essa limitação se justifica pelo fato de a BPMN ser a notação padrão atualmente na indústria para a modelagem em nível de negócio [HAR10] e os casos de uso da UML serem amplamente utilizados pela empresas de desenvolvimento de software para a modelagem de sistemas.

1.2 QUESTÃO DE PESQUISA

Com base no exposto acima a questão de pesquisa apresentada no presente trabalho é *“Como transformar modelos de processo de negócio, feitos em BPMN, para modelos de sistema expressos através de casos de uso da UML?”*.

1.3 OBJETIVOS

A partir da questão de pesquisa proposta é possível definir o objetivo geral e os objetivos específicos do trabalho, apresentados a seguir.

1.3.1 OBJETIVO GERAL

Desenvolver um método que auxilie na transformação de modelos de processos de negócio criados em BPMN para modelos de sistema expressos através de casos de uso da UML.

1.3.2 OBJETIVOS ESPECÍFICOS

OE1 – Analisar as propostas existentes que procuram realizar a transformação entre modelos de negócio e modelos de sistema.

OE2 – Estudar a arquitetura dirigida por modelos (MDA) e metamodelos da BPMN e da UML, identificando os elementos envolvidos nas transformações.

OE3 – Definir os elementos de negócio contemplados na transformação bem como a estrutura do caso de uso resultante da transformação.

OE4 – Criar um método que auxilie na transformação, visando considerar os principais elementos utilizados na BPMN bem como os principais elementos dos casos de uso.

OE5 – Aplicar o método proposto analisando os resultados obtidos.

1.4 METODOLOGIA

Uma revisão sistemática realizada pelo autor desta pesquisa em 2010 indicou que não há uma solução padrão na área para realizar a transformação de modelos de processos de negócio para modelos de sistema. Também se verificou na literatura que apenas os trabalhos mais recentes utilizam a BPMN como notação para a modelagem de negócio, sendo que vários elementos não são abordados, como diferentes *gateways*, eventos e artefatos.

A presente pesquisa caracteriza-se por ser do tipo exploratória, pois estuda assuntos ainda não abordados nas anteriores, ou abordados de forma não aprofundada. A estratégia de pesquisa é do tipo Projeto e Criação (*Design and Creation*), segundo a classificação de Oates [OAT06]. Essa

estratégia tem como foco o desenvolvimento de um novo produto ou artefato, nesse caso o método de transformação. Para sua validação são utilizadas aplicações de exemplo para a solução proposta.

As etapas para realização do trabalho estão descritas abaixo.

Etapa1

Atualizar o referencial teórico relacionado às propostas existentes para transformação de modelos de processo de negócio para modelos de sistema. Nesse sentido, deve ser utilizada como base a revisão sistemática realizada em 2010 [HER10] pelo autor desta pesquisa.

Etapa2

Estudar os metamodelos da BPMN e da UML, assim como as definições acerca de transformações de modelos da MDA, verificando os elementos dos metamodelos envolvidos na transformação.

Etapa 3

Identificar os principais estudos com o mesmo escopo definido, ou seja, que utilizem BPMN e casos de uso da UML. Através desses estudos, verificar melhorias que podem ser realizadas visando à criação de um método que atenda necessidades não tratadas nas propostas atuais.

Etapa 4

Definir as limitações que devem ser consideradas no modelo de processo de negócio utilizado na transformação, bem como a estrutura do caso de uso resultante da transformação.

Etapa 5

Desenvolver o método que auxilie na transformação de um modelo de processo de negócio feito em BPMN para casos de uso da UML, gerando o diagrama de casos de uso bem como o detalhamento parcial dos mesmos.

Etapa 6

Aplicar o método proposto através de exemplos de processos de negócio que contenham os elementos contemplados pelo método analisando o resultado obtido.

1.5 ESTRUTURA DA DISSERTAÇÃO

O texto está organizado em 7 capítulos. O Capítulo 1 apresenta a introdução, motivação, objetivos da pesquisa e metodologia utilizada. O Capítulo 2 apresenta conceitos básicos sobre

modelagem de processos de negócio e descreve os elementos da BPMN assim como seu metamodelo.

O Capítulo 3 discorre sobre modelagem de sistemas utilizando casos de uso da UML, além de descrever conceitos da MDA relacionados à transformação de modelos.

No Capítulo 4 estão apresentados os trabalhos relacionados com a presente dissertação, assim como considerações sobre os mesmos.

O Capítulo 5 apresenta o método de transformação proposto por esta pesquisa.

No Capítulo 6 são descritas aplicações do método proposto. Para isso são utilizados dois exemplos de aplicações do método.

Por fim, o Capítulo 7 apresenta as considerações finais e propostas de trabalhos futuros, seguido das referências bibliográficas utilizadas.

2. MODELAGEM DE PROCESSO DE NEGÓCIO

Esse capítulo apresenta conceitos básicos sobre modelagem de processos de negócio e sobre a BPMN, que é uma notação utilizada para representação desses processos. O capítulo está organizado da seguinte forma: a Seção 2.1 descreve conceitos básicos relacionados à modelagem de processos de negócio; a Seção 2.2 apresenta elementos da BPMN utilizados no presente trabalho; por último, na Seção 2.3, é mostrado o metamodelo da BPMN.

2.1 CONCEITOS BÁSICOS SOBRE MODELAGEM DE PROCESSOS DE NEGÓCIO

A modelagem de processos de negócio permite uma série de benefícios, tais como: uniformização do entendimento da forma de trabalho; análise e melhoria do fluxo de informações; explicitação do conhecimento sobre os processos gerando know-how para a organização; apoio na tomada de decisões [VER96].

Para entender o que é a modelagem de processos de negócio é importante consolidar alguns conceitos básicos, como *processos do negócio*, *gestão de processos do negócio*, *sistemas de gerenciamento de processos do negócio* e *modelo de processos do negócio*.

Segundo Weske [WES07], um processo do negócio consiste em um conjunto de atividades que são executadas coordenadamente em um ambiente organizacional. A execução conjunta dessas atividades visa atingir um objetivo de negócio. Por sua vez, Verdant [VER96] define um processo do negócio como sendo uma seqüência (ou um conjunto parcialmente ordenado) de atividades organizacionais, cuja execução é disparada por algum evento gerando resultados quantificáveis ou visíveis.

A *Workflow Management Coalition* (WfMC) [WMC99] diz que um processo de negócio é um conjunto de um ou mais procedimentos relacionados que coletivamente contribuem para a realização de um objetivo de negócio, normalmente dentro de uma estrutura organizacional definindo papéis e relacionamento.

A gestão de processos de negócio inclui conceitos, métodos e técnicas para permitir o *design*, administração, configuração, implantação e análise dos processos de negócio. A base da gestão de processos de negócio é a explícita representação dos processos de negócio com suas atividades e restrições de execução entre elas. Uma vez definidos esses processos se torna possível a analisar e melhorar os mesmos [WES07].

Para a automatização dos processos do negócio das organizações se faz necessário a utilização de um software de apoio. Esses softwares podem ser chamados de Sistemas de Gerenciamento de Processos do Negócio (SGPN). Um SGPN consiste em um software dirigido a

partir de representações de processos utilizado para coordenar a adoção de processos do negócio definidos [WES07].

Um modelo de processos de negócio consiste em um conjunto de modelos contendo atividades e restrições de execução entre as mesmas. Uma instância de um processo do negócio representa um caso concreto do ambiente operacional de uma empresa formado por instâncias de atividades [WES07].

Uma das finalidades da modelagem de processos de negócio é expressar de forma clara e objetiva as atividades realizadas pelos diferentes envolvidos no processo dentro de uma ou mais organizações.

A utilização de soluções informatizadas para dar suporte aos processos de uma organização é fundamental para que as mesmas se mantenham competitivas [MUT08]. A falta de entendimento das necessidades de negócio é um dos motivos de insucesso de alguns sistemas [SAN02]. Para que os sistemas efetivamente auxiliem a organização é necessário que atendam as necessidades de negócio da mesma [SAN02].

Nesse sentido a modelagem de processos de negócio se torna importante para entendimento dos requisitos de um sistema. A representação de um modelo de processo de negócio pode ser feita através de diagramas desenhados utilizando uma determinada notação como, por exemplo, a BPMN.

2.2 BUSINESS PROCESS MODELING NOTATION (BPMN)

Os modelos de processos de negócio são desenhados utilizando alguma notação específica para esse fim. Algumas abordagens tentaram unificar a forma de desenhar processos de negócio, como o uso de redes de Petri [AAL98] e modelagem de processos baseadas em eventos (*event-driven process chain*) [SCH05]. A UML também já foi utilizada para modelagem de processo de negócio [ERI00, BAS02].

Nos últimos anos a BPMN [BPM09] vem se consolidando como padrão da indústria para modelagem de processos de negócio [HAR10]. A BPMN foi proposta em 2004 pelo *Business Process Management Institute* (BPMI), e logo se tornou a principal notação utilizada para desenho dos processos de negócio.

A BPMN consiste em uma notação gráfica utilizada para criação dos modelos que representam os processos de negócio de uma determinada organização. O principal objetivo da BPMN é prover uma notação que seja facilmente entendida por todos envolvidos no negócio [BPM09].

O processo de negócio é desenhado na BPMN utilizando seus elementos gráficos. Esses elementos estão divididos em quatro categorias: objetos de fluxo; objetos de conexão; partições (*swimlanes*); artefatos [BPM09].

2.2.1 OBJETOS DE FLUXO

Os objetos de fluxo são divididos em três tipos: atividades, desvios (*gateways*) e eventos.

- **Atividade**

Uma atividade descreve um trabalho realizado dentro da organização por alguma pessoa ou setor. A atividade pode ser atômica ou complexa.

- *Tarefa*

Uma atividade atômica é chamada de tarefa. Uma atividade atômica, ou tarefa, representa uma ação realizada que não pode (ou não é interessante) dividi-la em outras. A BPMN apresenta diferentes tipos de tarefas, dentre as quais se destacam: tarefas manuais; tarefas de usuário e tarefas de serviço. Uma tarefa de usuário representa uma ação realizada por um participante que não necessita o uso de uma aplicação, como atender um telefonema. Uma tarefa de usuário representa uma ação realizada com o auxílio de uma aplicação (sistema). A tarefa de serviço representa a execução de um serviço automatizado, como um *WebService*. [WHI09]

- *Subprocesso*

Uma atividade não atômica representa um subprocesso. Um subprocesso é uma atividade composta por uma série de outras atividades. O subprocesso pode ser visto no diagrama como uma atividade única, semelhante a uma tarefa, abstraindo os outros elementos contidos dentro dele.

As atividades são representadas no diagrama BPMN através de um retângulo com as bordas arredondadas. A Figura 1 apresenta os símbolos usados para a atividade do tipo tarefa e atividade do tipo subprocesso.



Figura 1 - Representação de Atividades na BPMN

- **Desvio (*gateway*)**

Os *gateways* são utilizados para controlar o fluxo do processo, podendo esse convergir ou divergir ao longo da sua execução. Um *gateway* é um elemento não-obrigatório no processo. Caso o

fluxo não necessite ser controlado não se faz necessária a utilização de um *gateway*. Na BPMN o mesmo tipo *gateway* é usado tanto para dividir como para unir o fluxo do processo.

A BPMN apresenta cinco tipos de *gateways*: *gateway* exclusivo, *gateway* inclusivo, *gateway* paralelo, *gateway* complexo, *gateway* baseado em eventos. Nesse trabalho são abordados os três primeiros citados.

- *Gateway Exclusivo*

Um *gateway* exclusivo divide o fluxo em dois ou mais fluxos, cada um deles com uma condição específica. As condições apresentadas são avaliadas e no momento em que uma delas é verdadeira o fluxo do processo segue por aquele caminho, ignorando os demais. Dessa forma apenas uma das condições apresentadas no *gateway* exclusivo pode ser verdadeira. Um dos fluxos do *gateway* pode ser definido como *default*. Esse será o fluxo a ser seguido caso nenhuma condição seja verdadeira.

- *Gateway Inclusivo*

O *gateway* inclusivo é semelhante ao exclusivo, sendo que a diferença está no fato de que mais de um fluxo pode ser seguido caso mais de uma condição seja verdadeira. As condições verdadeiras dão origem a fluxos paralelos. Também pode ser definido um caminho *default*, usado quando nenhuma das demais condições é verdadeira.

Uma vez utilizado um *gateway* inclusivo para dividir o fluxo outro *gateway* inclusivo deve ser usado para unir o mesmo novamente. Esse *gateway* inclusivo aguarda o término da execução de todos os fluxos que tiveram condições verdadeiras no *gateway* inclusivo que dividiu o fluxo.

- *Gateway Paralelo*

Um *gateway* paralelo é usado para dividir o fluxo em vários fluxos paralelos e para sincronizar os mesmos posteriormente. Um *gateway* paralelo não avalia nenhuma condição. Todos os fluxos originados pelo *gateway* paralelo são executados de forma paralela. A sincronia desses fluxos é feita por outro *gateway* paralelo, que aguarda que todos os fluxos paralelos sejam encerrados para a continuidade do processo.

A Figura 2 apresenta os símbolos usados para os diferentes tipos de *gateways*.



Figura 2 - Gateways da BPMN

- **Eventos**

Evento representa algo que ocorre no início, meio ou final do processo. Pode representar a chegada de uma mensagem ou um determinado tempo transcorrido. Os eventos podem ser de três tipos: eventos de início, eventos intermediários e eventos de final.

- *Evento de Início*

Eventos de início são usados para indicar o início de um processo. Os eventos de início podem ser de diferentes tipos, de acordo com a causa do mesmo. Nesse trabalho são abordados três tipos de eventos de início: o evento sem causa especificada e o evento de início do tipo tempo (*timer*) ou evento temporal e o evento condicional. Quando nenhuma causa é especificada o evento de início representa apenas o ponto de início do processo e o mesmo é iniciado sem que nenhuma condição seja necessária. Já um evento de início do tipo tempo pode indicar que o mesmo só inicia quando uma data/hora for atingida ou quando um ciclo de tempo for completado. O evento condicional possui uma condição de negócio associada ao mesmo.

- *Evento Intermediário*

Um evento intermediário ocorre entre o início e o final do processo. Pode ser de diferentes tipos, assim como o evento de início. Um evento intermediário do tipo tempo, por exemplo, é usado para indicar tempos de espera no processo.

Um evento intermediário pode estar localizado no fluxo do processo ou anexado a uma tarefa. Nesse último caso o evento só pode ser do tipo exceção ou compensação. Um evento do tipo exceção indica que uma determinada exceção ocorreu na execução da tarefa e um determinado fluxo do processo deve ser executado.

- *Evento de Fim*

Um evento fim indica o final do processo. O evento de fim é opcional, podendo o processo ter nenhum, um, ou vários eventos de final.

A Figura 3 apresenta o símbolo utilizado para os diferentes tipos de eventos citados anteriormente.

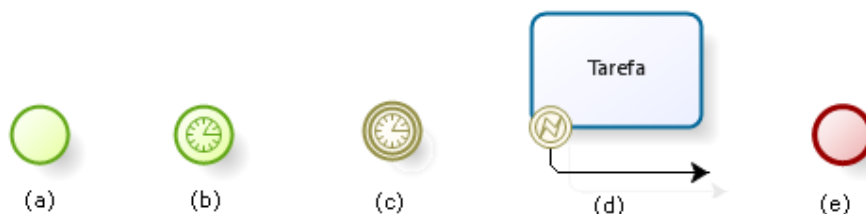


Figura 3 - (a) Evento de início sem causa definida. (b) Evento de início tipo tempo. (c) Evento intermediário do tipo tempo. (d) Evento intermediário de exceção anexado a tarefa. (e) Evento de fim

2.2.2 OBJETOS DE CONEXÃO

Os objetos de conexão são utilizados para conectar os diferentes elementos do modelo. Os objetos de conexão existentes na BPMN são: fluxo de seqüência, fluxo de mensagem e associação.

- *Fluxo de Seqüência*

Um fluxo de seqüência é usado para mostrar a ordem em que as atividades ocorrem no processo.

- *Fluxo de Mensagem*

O fluxo de mensagem é usado para mostrar a troca de mensagens entre participantes do processo. Esse fluxo é usado quando o diagrama possui mais de um *pool*.

- *Associação*

O objeto de associação se utiliza para associar os artefatos aos objetos do fluxo. A Figura 4 a seguir exhibe os três tipos de conexões.

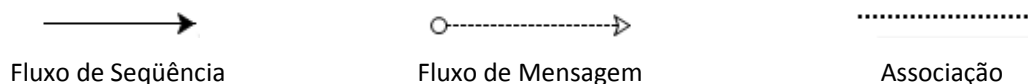


Figura 4 - Objetos de Conexão da BPMN

2.2.3 PARTIÇÕES (SWINLANES)

As partições são usadas para organizar as atividades dentro do modelo. A BPMN possui dois tipos de partições: Participante (*Pool*) e Raia (*Lane*).

- *Participante*

Um *pool* representa um participante do processo. Um *pool* contém um processo de negócio e pode ser usado em modelos de processos *Business to Business* (B2B) [BPM09].

- *Raia*

As raias são partes de um participante. São utilizadas para representar diferentes categorias dentro de um participante. Geralmente são usados para indicar diferentes papéis da organização representada pelo participante.

A Figura 5 apresenta os símbolos usados para o participante e para a raia.

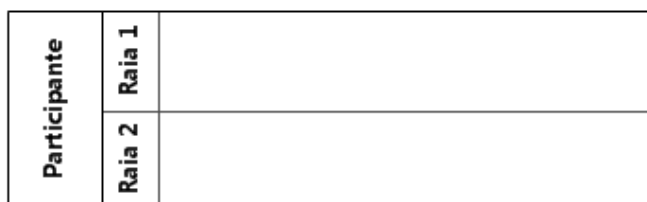


Figura 5 - Partições da BPMN (Participante e Raia)

2.2.4 ARTEFATOS

Os artefatos permitem adicionar informações ao diagrama BPMN permitindo uma maior flexibilidade na construção do modelo. Os artefatos não interferem no fluxo normal do processo, sendo usados apenas para documentar melhor o processo modelado.

A BPMN apresenta alguns tipos pré-definidos de artefatos, como o objeto de dados, anotação e grupo.

- *Objeto de Dados*

Um objeto de dados representa dados gerados ou utilizados por algum elemento do processo. Pode representar um documento, eletrônico ou não. O objeto de dados é ligado ao elemento do processo através de uma linha pontilhada.

- *Anotação*

Anotações são utilizadas para associar um determinado comentário a um elemento do processo. É usado para explicar de forma mais detalhada alguma parte do processo.

- *Grupo*

O artefato grupo é usado na BPMN para agrupar atividades e/ou outros elementos do processo. A BPMN não define uma semântica associada a esse elemento, sendo usado apenas para fins de visualização no diagrama.

A Figura 6 apresenta os símbolos usados para os artefatos pré-definidos na BPMN.

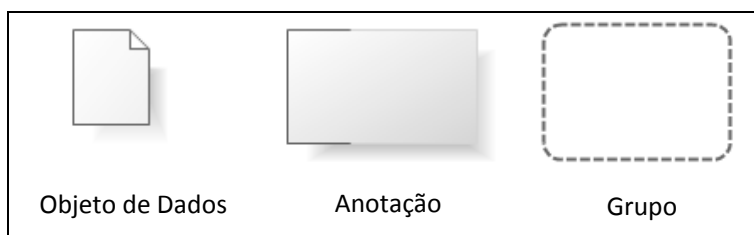


Figura 6 - Artefatos da BPMN

2.3 METAMODELO DA BPMN

Um metamodelo é basicamente um modelo de modelos [OMG03]. Um metamodelo pode ser representado através de um diagrama de classes. Dessa forma ele se torna uma espécie de dicionário visual dos elementos de um modelo. Nessa seção é apresentado um metamodelo da BPMN obtido a partir da documentação da OMG para a BPMN [OMG09b].

O metamodelo apresentado na Figura 7 mostra os principais elementos utilizados para a criação de modelos de processo de negócio e que são pertinentes a esse trabalho. Os elementos em destaque são os utilizados pelo método proposto no Capítulo 5 para transformação de modelos em BPMN para casos de uso da UML.

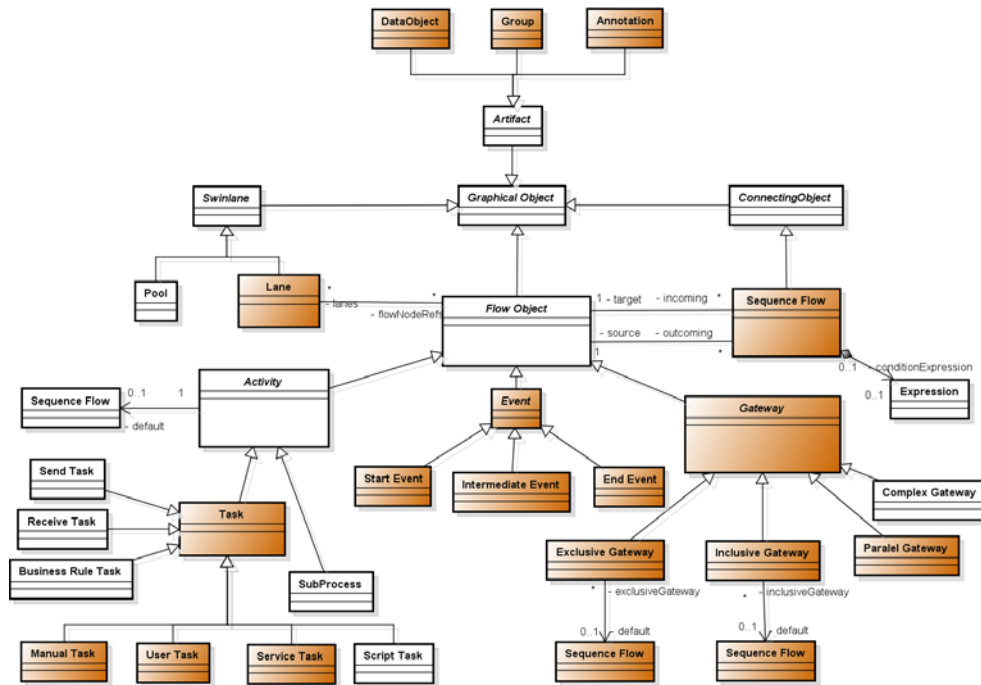


Figura 7 - Metamodelo da BPMN (adaptado de [OMG09b])

Na seção anterior estão apresentados os principais elementos da BPMN divididos em quatro categorias: objetos de fluxo, objetos de conexão, partições e artefatos. No metamodelo essas quatro categorias aparecem representadas como classes abstratas.

A classe *Flow Object* (Objeto de Fluxo) é a base para a criação dos elementos de fluxo que compõem o processo de negócio, tais como tarefas, eventos e *gateways*. A conexão entre os objetos de fluxo se dá no nível da superclasse *Flow Object*. Cada objeto de uma classe concreta derivada dessa classe pode possuir um fluxo de entrada e/ou de saída e esse fluxo está associado a outro objeto de uma classe concreta também derivada de *Flow Object*. A associação ocorre utilizando objetos da classe *Sequence Flow*. Por exemplo, uma *User Task* está associada a um *Exclusive Gateway* no momento em que um objeto da classe *User Task* se associa a um objeto de *Sequence Flow* que está associado a um objeto da classe *Exclusive Gateway*.

Os elementos do tipo artefato são subclasses da classe abstrata *Artifact*, podendo ser objetos das classes *Group*, *Annotation* e *DataObject*. Os objetos de *Annotation* e *DataObject* podem ser conectados como elementos do fluxo através de objetos de conexão do tipo *Association*. Já objetos da classe *Group* não são conectados com objetos do fluxo. Esses objetos são utilizados para destacar visualmente determinadas partes do diagrama e não possuem restrição em relação a *Lanes* e *Pools*.

As classes existentes nos metamodelos auxiliam na identificação dos elementos envolvidos nas transformações de BPMN para UML. Não é escopo do presente trabalho aprofundar as descrições das classes dos metamodelos. Ele é usado para auxílio na definição das transformações.

3. MODELAGEM DE SISTEMAS, MDA E TRANSFORMAÇÕES DE MODELOS

Esse capítulo tem por finalidade apresentar conceitos básicos relacionados à modelagem de sistemas utilizando casos de uso da UML, sobre *Model-driven Architecture* (MDA) e transformações entre modelos. As Seções de 3.1 até 3.5 apresentam uma visão geral sobre casos de uso na UML, modelos, descrição e metamodelos de casos de uso. A Seção 3.6 apresenta a MDA. Já a Seção 3.7 mostra conceitos relacionados com transformações entre modelos utilizando os princípios da MDA.

3.1 CASOS DE USO E A UML

Os casos de uso foram propostos por Jacobson [JAC92] na metodologia de desenvolvimento orientado a objetos. Segundo Jacobson um caso de uso é uma maneira específica de utilizar um sistema usando alguma funcionalidade. Constitui um curso completo da interação que ocorre entre um ator e o sistema [JAC92].

Vários autores apresentam diferentes definições sobre o que é um caso de uso, entretanto a idéia fundamental é a mesma: representar as interações que ocorrem entre o ator, ou atores, e o sistema para atingir um determinado objetivo de interesse do ator.

Um caso de uso engloba um conjunto de cenários em que cada cenário é um encadeamento isolado ao longo do caso de uso [SOM07]. Casos de uso descrevem as interações entre o usuário e o sistema, focando no que o sistema faz para o usuário. O modelo de casos de uso descreve na totalidade o comportamento funcional do sistema [LEF00].

Um caso de uso é uma descrição narrativa do processo do domínio da aplicação. Ele representa um requisito funcional do sistema [BOO05].

Casos de uso são fundamentalmente textuais, embora possam ser escritos em fluxogramas ou outras formas gráficas. Casos de uso são tipicamente oferecidos como uma maneira de capturar e modelar requisitos funcionais conhecidos. Pessoas acham o formato de história mais fácil de compreender do que as longas especificações de requisitos tradicionais. [COC05]

A UML utiliza o conceito de casos de uso para expressar os requisitos de sistema. Na definição da UML um caso de uso é uma descrição de um conjunto de seqüências de ações, inclusive suas variantes, que um sistema realiza para produzir um resultado de valor observável para o ator [BOO05].

A utilização de caso de uso tomou um grande impulso a partir do *Rational Unified Process* (RUP) [KRU00]. O RUP é um framework para desenvolvimento de software dirigido por casos de uso. Os requisitos funcionais do sistema são mapeados para casos de uso. O RUP utiliza as definições da UML relativas aos casos de uso.

3.2 MODELOS DE CASOS DE USO

Um modelo de casos de uso é uma abstração visual dos casos de uso do sistema e seus relacionamentos. Os componentes de um modelo de casos de uso são [JAC92]:

- Ator: representa um papel que tipicamente solicita ações e recebe reações do sistema. Cada ator pode participar de vários casos de uso.
- Caso de Uso: representa um documento narrativo que descreve a seqüência de eventos feitos por um ator do sistema.
- Sistema: o sistema que está sendo modelado.

A UML apresenta o diagrama de caso de uso para representar o modelo de casos de uso. Esse diagrama é formado pelos casos de uso com seus atores e os relacionamentos entre os casos de uso e entre os atores.

A Figura 1 apresenta os símbolos usados para representar atores e casos de uso no diagrama da UML:



Figura 8 - Símbolos para atores e casos de uso usados no diagrama da UML

3.3 DESCRIÇÃO DE CASOS DE USO

A descrição de um caso de uso consiste no detalhamento textual do mesmo utilizando um determinado formato. A descrição de um caso de uso deve considerar informações do tipo [LAR04]:

- como e quando o caso de uso se inicia;
- as interações com os atores envolvidos;
- as seqüências normais de execução;
- as seqüências alternativas de execução.

O RUP apresenta um formato resumido para o detalhamento de casos de uso. Esse formato contém os elementos abaixo:

- *Nome*: nome do caso de uso.
- *Breve descrição*: breve descrição do caso de uso.
- *Fluxo básico de eventos*: contém a descrição textual (geralmente um passo a passo numerado) da interação básica entre o ator e o sistema para atingir um objetivo.

- *Fluxos alternativos*: contém a descrição textual dos diversos fluxos alternativos ao fluxo básico definido.
- *Pré-condições*: descrição textual que define as condições necessárias no sistema para que o caso de uso inicie.
- *Pós-condições*: descrição textual que define possíveis estados que o sistema poderá se encontrar depois do término do caso de uso.
- *Pontos de Extensão*: uma lista de locais dentro do fluxo de eventos (básico ou alternativo) do caso de uso cujo comportamento adicional é inserido usando relacionamentos de extensão.
- *Requisitos especiais*: descrição textual de algum requisito não funcional, como requisitos de confiabilidade, desempenho ou suportabilidade.

Outro formato de detalhamento de casos de uso é apresentado por Cockburn [COC05]. Seguem alguns dos elementos desse formato:

- *Número*: número do caso de uso.
- *Nome*: nome do caso de uso.
- *Objetivo no contexto*: objetivo do caso de uso dentro do contexto.
- *Escopo*: escopo do caso de uso.
- *Nível*: resumido, tarefa primária, subfunção.
- *Pré-condições*: estado esperado para início do caso de uso.
- *Ator primário*: ator primário do caso de uso.
- *Acionador*: ação do sistema que dispara o caso de uso, podendo ser um evento de tempo.
- *Cenário de sucesso principal*: passos do cenário desde o gatilho até o término do objetivo. Os passos são numerados.
- *Extensões*: extensões dos passos do cenário principal, podendo ser outra ação ou apontar para outro caso de uso.

3.4 RELACIONAMENTO ENTRE CASOS DE USO

A definição da UML [UML11] apresenta três tipos de relacionamentos entre casos de uso: a Extensão, a Inclusão e a Generalização.

3.4.1 RELACIONAMENTO DE EXTENSÃO

Um relacionamento de extensão entre casos de uso indica que um caso de uso tem seu comportamento estendido por outro caso de uso. O caso de uso base pode permanecer isolado, mas em determinadas situações seu comportamento pode ser estendido pelo comportamento de outro caso de uso. Essa extensão do comportamento pode ocorrer em diferentes pontos do caso de uso base, que são chamados de pontos de extensão [BOO05].

O relacionamento de extensão é utilizado para modelagem de uma parte de um caso de uso que pode ser considerada como opcional do sistema, separando assim o comportamento opcional do obrigatório. Esse relacionamento pode ser utilizado para modelar um subfluxo em separado. Também pode ser usado para modelar vários fluxos que podem ser inseridos em certo ponto de acordo com uma interação explícita com um ator. A extensão é representada como uma dependência na UML, utilizando o estereótipo <<extend>>, conforme ilustrado na Figura 9 [BOO05].

Nessa figura o caso de uso Enviar Pedido Parcial é disparado no passo do caso de uso base em que é feita a verificação se os produtos estão prontos. Caso não estejam todos prontos (condição descrita no detalhamento textual do caso de uso) é disparado o caso de uso Enviar Pedido Parcial.

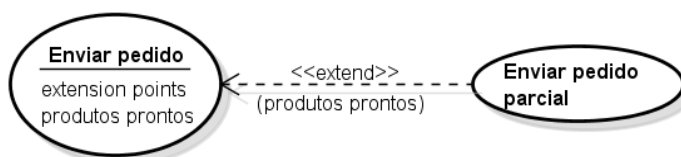


Figura 9 - Exemplo de relacionamento de extensão [BOO05]

Na documentação do RUP o relacionamento de extensão é definido como um relacionamento que estabelece a conexão entre um caso de uso de extensão e um caso de uso base. Esse relacionamento é utilizado para mostrar que uma parte do caso de uso pode ser opcional (ou possivelmente opcional). A extensão é condicional, dessa forma a sua execução depende da interação ocorrida no caso de uso base. As condições para que o relacionamento de extensão ocorra devem ser descritas na descrição do caso de uso em uma seção para os relacionamentos de extensão.

A descrição do relacionamento de extensão deve ter referências aos pontos de extensão do caso de uso base ao qual se refere assim como a condição para que o relacionamento ocorra. A extensão é descrita no caso de uso base [KRU00].

A descrição do relacionamento de extensão da Figura 9 pode ser realizada da seguinte forma:

Ponto de Extensão: Produtos Prontos

Essa extensão depois do passo *n* (*n* é o passo onde é feita a verificação se os produtos estão prontos)

Condição: Produtos não estão totalmente disponíveis

3.4.2 RELACIONAMENTO DE INCLUSÃO

Um relacionamento de inclusão entre casos de uso indica que o caso de uso base incorpora explicitamente o comportamento de outro caso de uso em um local específico do caso de uso base. Esse relacionamento é utilizado para evitar descrever o mesmo fluxo de eventos várias vezes, colocando esse comportamento em um caso de uso próprio [BOO05]. A execução do caso de uso incluído é obrigatória. Esse caso de uso é sempre disparado pelo caso de uso base.

O relacionamento de inclusão é representado como uma dependência utilizando o estereótipo <<include>>, conforme a Figura 10. Nesse exemplo o caso de uso Rastrear Pedido inclui em algum ponto do seu fluxo o caso de uso Validar Usuário.



Figura 10 - Exemplo de relacionamento de inclusão [BOO05]

Para especificar a localização no fluxo de eventos do caso de uso base na qual é incluído o comportamento do outro caso de uso, deve simplesmente ser escrito *include* seguido do nome do outro caso de uso [BOO05].

3.4.3 RELACIONAMENTO DE GENERALIZAÇÃO

A generalização entre casos de uso é semelhante à generalização entre classes. Um caso de uso filho herda o comportamento do caso de uso pai, sendo que o caso de uso filho deve acrescentar ou até sobrescrever o comportamento do caso de uso pai [BOO05].

Um relacionamento de generalização entre casos de uso define que o caso de uso filho contém todos os atributos, relacionamentos, sequência de comportamento e pontos de extensão definidos no caso de uso pai [UML11].

A representação da generalização entre casos de uso é feita através de uma linha cheia com uma seta aberta, da mesma forma que é feita entre classes [BOO05]. A Figura 11 mostra um exemplo. Nesse exemplo o caso de uso pai é o Validar Usuário. Essa validação pode ser realizada de duas formas, através da senha ou através da análise biométrica (pela retina, por exemplo). O comportamento inicial da validação é o mesmo independente do método usado para validar o usuário.

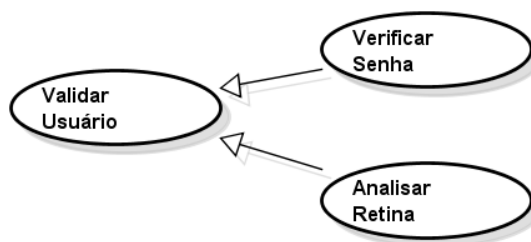


Figura 11 - Exemplo de relacionamento de generalização [BOO05]

3.5 METAMODELO DE CASOS DE USO

Um metamodelo é uma espécie de dicionário visual dos elementos de um modelo e pode ser expresso utilizando um diagrama de classes. A OMG apresenta um metamodelo para os diagramas da UML, inclusive para o diagrama de casos de uso [UML11, WEI07]. Entretanto, o metamodelo apresentado pela OMG não contempla itens relativos ao detalhamento dos casos de uso. O metamodelo da OMG apresenta elementos que identificam os casos de uso, atores e relacionamento entre casos de uso.

A Figura 12 apresenta o metamodelo da UML para casos de uso. Esse metamodelo não apresenta elementos referentes ao detalhamento dos casos de uso, ficando restrito apenas ao diagrama de casos de uso.

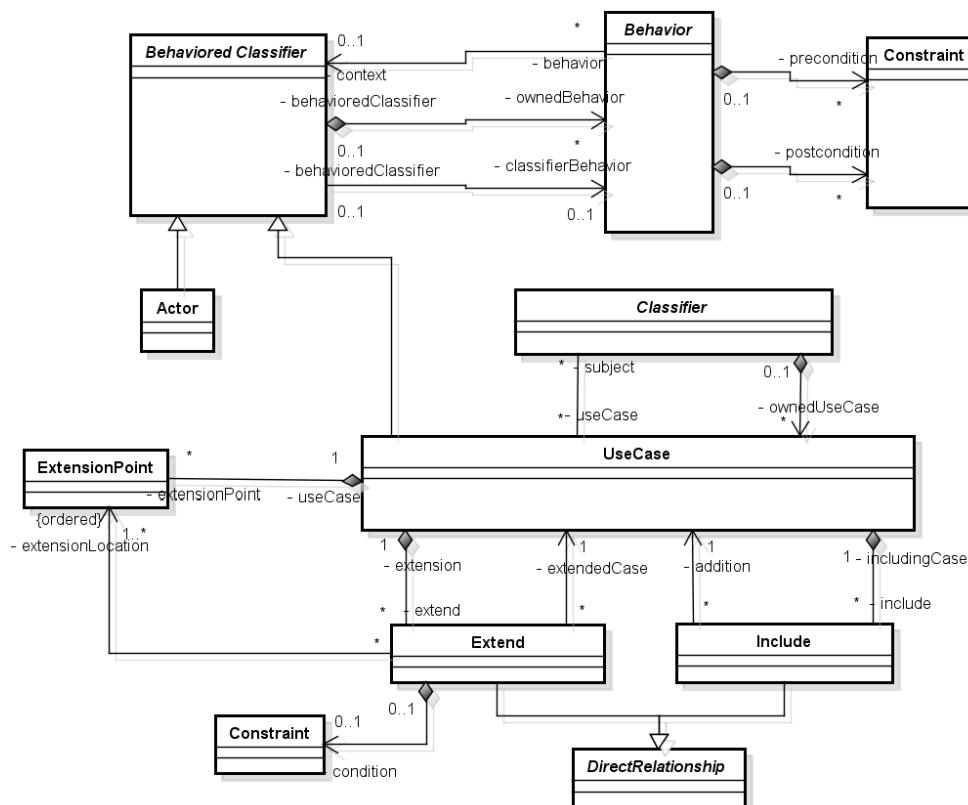


Figura 12 – Metamodelo da UML para casos de uso [UML11]

A principal classe é a *UseCase*. Essa classe é filha de outra classe abstrata chamada de *BehavioredClassifier*. A classe *Actor* também é filha da classe *BehavioredClassifier*, que também é filha da classe *Classifier* (não está representada na figura). A relação entre o ator e o caso de uso ocorre no nível do relacionamento definido entre os classificadores da UML.

O metamodelo da UML é complexo e as definições de suas classes e relacionamentos são revisadas a cada nova versão da linguagem. Não é escopo desse trabalho uma explicação detalhada dos elementos desse metamodelo. O metamodelo foi utilizado para analisar como as transformações entre elementos de diferentes modelos poderiam ser realizadas e quais classes dos metamodelos estão envolvidas em cada transformação.

No trabalho de Somé [SOMé09] é apresentado um metamodelo para o detalhamento textual de casos de uso. Esse metamodelo contempla uma série de elementos utilizados na descrição do caso de uso, como passos do fluxo, acionadores, pré e pós-condições, entre outros, estendendo o metamodelo definido pela *Object Management Group* (OMG).

O trabalho de Somé é uma extensão do metamodelo da UML contendo elementos relativos à descrição dos casos de uso. Um caso de uso possui elementos como fluxo de eventos principal, fluxos alternativos, além de outros campos que podem ser definidos dependendo do formato de descrição utilizado. Esse metamodelo procura formalizar a representação desses elementos. A Figura 13 apresenta algumas das classes principais do metamodelo.

A utilização de um metamodelo que contenha elementos relativos à descrição do caso de uso possibilita uma modelagem mais padronizada do detalhamento do mesmo, uma vez que os elementos a serem utilizados estão definidos no metamodelo.

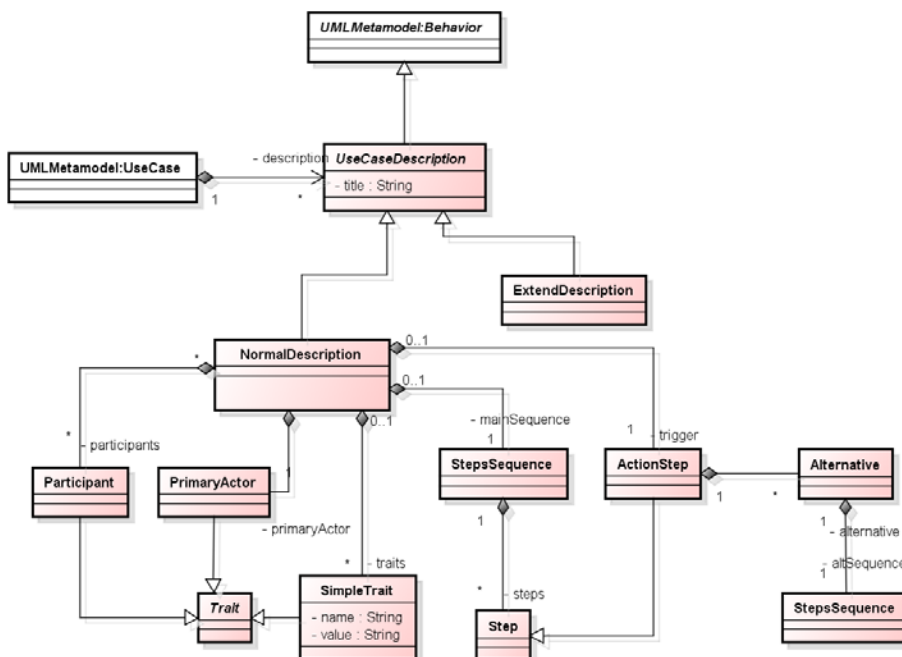


Figura 13 - Metamodelo (resumido) de descrição de caso de uso [SOMé09]

A principal classe desse metamodelo é a *UseCaseDescription*. Essa classe é abstrata, tendo como subclasses concretas a *NormalDescription* e a *ExtendDescription*. Pela definição do metamodelo cada caso de uso pode ter várias descrições. Somé argumenta que isso permite que cada caso de uso possua várias descrições associadas cada uma de acordo com um determinado nível de detalhamento.

A classe *NormalDescription* possui uma seqüência de passos (classe *StepsSequence*) que é composta por vários passos, que podem ser do tipo *ActionStep*. Cada objeto da classe *ActionStep* pode ter uma seqüência de passos alternativos, através da associação com objetos da classe *Alternative*.

Esse metamodelo é utilizado no presente trabalho com o objetivo de indicar quais elementos dos metamodelos estão envolvidos nas transformações realizadas entre elementos dos modelos de processos de negócio em BPMN para elementos da descrição de casos de uso. Não é escopo do presente trabalho detalhar os elementos do metamodelo da parte textual, mas sim utilizar o mesmo como um complemento ao metamodelo da UML para auxiliar na identificação das transformações apresentadas no Capítulo 5.

3.6 MDA

A *Model-Driven Architecture* (MDA), ou Arquitetura Dirigida por Modelos, consiste em uma abordagem para desenvolvimento de software que coloca a criação de modelos como centro do processo de desenvolvimento. O princípio geral da MDA é separar a especificação de operações do sistema de detalhes de implementação em uma determinada plataforma [OMG03].

Os softwares são desenvolvidos a partir de modelos criados dentro dos diversos níveis de abstração. Esses níveis de abstração vão desde o mais alto nível, que independe da plataforma ou tecnologia até o nível mais baixo que seria o próprio código-fonte. Essa passagem de um nível para outro é feita através de transformações entre modelos, assunto discutido com mais detalhes na Seção 3.7.

A MDA divide os modelos em três níveis de abstração [HAR04]: o *Computation Independent Model* (CIM), o *Platform Independent Model* (PIM) e o *Platform Specific Model* (PSM).

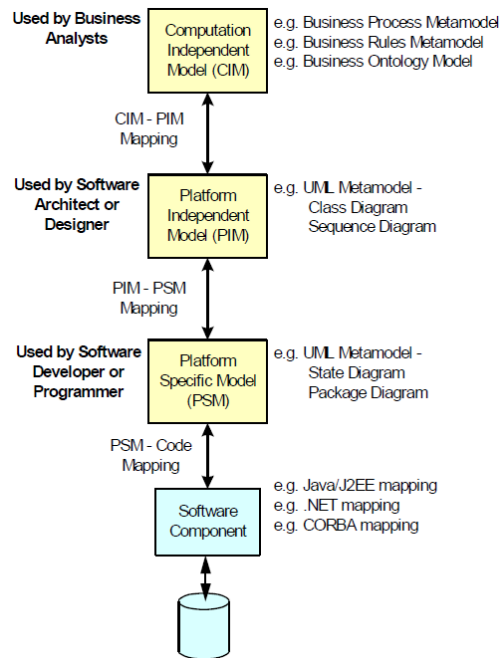


Figura 14 - Níveis da MDA [HAR04]

No nível CIM não são mostradas informações estruturais dos sistemas. Nesse nível estão contidos os modelos de processos de negócio. Esses modelos têm como objetivo modelar o processo de negócio sem levar em consideração restrições de tecnologia. Esses modelos geralmente são criados pelos analistas de negócio. Algumas partes de um modelo CIM podem ser suportadas por um sistema. Entretanto, o CIM independe de plataforma e para transformar um modelo CIM para um PIM é necessário identificar quais partes do CIM são suportadas por um sistema.

Os modelos de sistema representados através da UML com o uso de diagramas de casos de uso são exemplos de modelos no nível PIM. Eles são independentes de plataforma. Entretanto, são direcionados a identificar funcionalidades específicas do sistema não entrando em detalhes de como essas são implementadas.

Já diagramas de seqüência da UML constituem exemplos de modelos no nível PSM, pois contém detalhes específicos da plataforma de implementação utilizada. A Figura 14 mostra os diferentes níveis e os tipos de modelos pertencentes a cada um deles.

3.7 TRANSFORMAÇÕES ENTRE MODELOS

A transformação entre modelos é o processo de conversão de um modelo em outro modelo do mesmo sistema [OMG03]. Essa transformação é feita através da definição de regras de mapeamento entre os modelos. Essas regras são definidas para elementos dos metamodelos de cada um dos modelos criados. A partir da definição dessas regras um modelo criado utilizando um metamodelo pode ser transformado em outro modelo baseado em um metamodelo diferente.

No trabalho de Mens [MEN06], é proposta uma taxonomia para transformação entre modelos. Essa taxonomia classifica as transformações entre modelos de acordo com determinadas características, dentre as quais podem se destacar:

- Número de modelos de origem e destino

Um modelo de origem pode ser transformado em um ou mais modelos. O contrário também é verdade, ou seja, vários modelos de origem podem resultar em apenas um modelo destino.

- Transformação endógena versus exógena

Uma transformação endógena acontece dentro de um mesmo metamodelo. Por exemplo, um modelo representado em UML transformado em outro modelo em UML constitui uma transformação endógena. A transformação exógena ocorre quando os metamodelos são diferentes. Um exemplo é a transformação de um modelo em BPMN para um modelo em UML.

- Transformação horizontal versus vertical

Quando um modelo é transformado em outro dentro do mesmo nível de abstração essa transformação é chamada de horizontal. Por sua vez, a transformação vertical acontece quando um modelo em um nível de abstração é transformado em outro modelo em um nível de abstração maior ou menor.

- Nível de automação

O nível de automação faz uma distinção entre as transformações que são automáticas e as que necessitam de um trabalho manual para realizar a mesma. As automáticas são passíveis de implementação de rotinas para realizar as mesmas.

- Transformação sintática e/ou semântica

A transformação sintática leva em consideração apenas elementos sintáticos da transformação, não verificando a semântica da mesma. A transformação semântica é mais complexa, pois o novo modelo deve levar em conta a semântica dos elementos do modelo original.

Para a transformação de um modelo em um nível de abstração para um modelo em outro nível de abstração (transformação vertical) é necessária a análise dos elementos em nível de metamodelos. Os metamodelos da BPMN e de casos de uso da UML apresentados nas seções anteriores são utilizados para identificação das transformações entre os modelos de processo de negócio em BPMN (nível CIM) para modelos de casos de uso da UML (nível PIM). A transformação de modelos de nível PIM para PSM não é abordada nesse trabalho, mas pode ser assunto de um trabalho futuro.

4. TRABALHOS RELACIONADOS

Esse capítulo apresenta alguns trabalhos relacionados com a transformação de modelos de processo de negócio para modelos de sistema. Esses trabalhos foram obtidos a partir de uma revisão sistemática realizada pelo autor do presente trabalho em 2010 e depois atualizada em 2011.

Na análise dos trabalhos encontrados, realizada na revisão sistemática, verifica-se que não existe um método, estratégia ou abordagem que seja padrão atualmente. O que existe são diferentes propostas que tentam resolver o problema de transformar um modelo em um nível de abstração de negócio para um modelo em um nível de abstração de sistema.

Em relação às linguagens/notações utilizadas para criação dos modelos verifica-se que em trabalhos mais recentes existe um predomínio do uso da BPMN para modelagem dos processos de negócio [LIE04, ROD10, VAR08]. Outros trabalhos utilizam uma notação de modelagem de processo de negócio mais genérica, como é o caso de Okawa [OKA07] e Dijkman [DIJ02], sendo que esse último utiliza um diagrama de atividades. Já para a modelagem do sistema as propostas utilizam predominantemente os casos de uso da UML [DIJ02, BAE07, LIE04, ROD10, ODE03, LUN03, AZE08, EST03, BER09]. Além do diagrama de casos de uso também são utilizados diagramas de classes em alguns trabalhos [OKA07, ROD10, DAM08].

As próximas seções apresentam com mais detalhes alguns desses trabalhos citados anteriormente. Os trabalhos detalhados são os que procuram identificar casos de uso a partir de processos de negócio, descrevendo como realizar essa tarefa. Apenas um dos trabalhos citados [OKA07] não identifica casos de uso, mas sim classes de projeto software. Esse trabalho se torna interessante de ser estudado porque mostra outra forma de identificar elementos no nível de sistema a partir de modelos de processos no nível de negócio.

4.1 DERIVING USE CASE DIAGRAMS FROM BUSINESS PROCESS MODELS (2002)

O trabalho de Dijkman [DIJ02], publicado em 2002, é um dos primeiros a abordar a transformação de processos de negócio para casos de uso, sendo utilizado como referência pela maioria dos estudos sobre o assunto.

A proposta do trabalho é apresentar uma técnica para simplificar a captura de requisitos com casos de uso buscando derivar casos de uso a partir de processos de negócio. Nesse trabalho é apresentado o conceito de *Role* (Papel) e *Step* (Passo), que é utilizado por outros autores [LIE04, VAR08, BER09]. Um Passo consiste em um conjunto de atividades realizadas sem desvios pelo mesmo Papel do processo de negócio.

Também é de Dijkman a ideia de diferenciar as tarefas do processo de negócio em manuais ou executadas pelo sistema, essa última podendo ser com ou sem intervenção de um usuário. Os processos de negócio não são feitos em BPMN, mas sim em diagrama de atividades da UML.

A técnica consiste basicamente em um mapeamento entre os elementos do diagrama de processo de negócio e os casos de uso, definidos na Tabela 1.

Elementos de processo de negócio	Elementos do caso de uso
Papel	Ator
Passo	Caso de Uso
Associação entre Papel e Passo	Associação entre Ator e Caso de Uso
Tarefa	Interação
Tarefa em um Passo	Interação dentro de um Caso de Uso
Transição entre Tarefas em um mesmo Passo	Interações ordenadas dentro de um mesmo Caso de Uso
Restrição na transição	Restrição na interação
Caminhos de um desvio	Extensão de Caso de Uso ou caminho alternativo na descrição.

Tabela 1 - Regras de mapeamento de conceitos de negócio para casos de uso

O mapeamento é realizado com base nas regras definidas. Posteriormente é feita uma validação do resultado obtido com o objetivo de eliminar possíveis inconsistências.

4.2 A FRAMEWORK FOR BUSINESS MODEL DRIVEN DEVELOPMENT (2004)

Uma proposta de transformação de modelos BPMN para diagramas de casos de uso pode ser vista no trabalho de Liew [LIE04]. Nesse trabalho o autor faz uso da estrutura de *annotations* disponível na BPMN. Usando *annotations* é possível adicionar uma informação textual a qualquer elemento do modelo. Nesse artigo o autor propõe um mapeamento entre elementos do modelo de negócio para elementos de um diagrama de casos de uso.

Para atender a todos os conceitos de negócio necessários é proposta a inclusão de uma *annotation* que identifique o Ator do caso de uso. O Ator poderia ser identificado pela raia, mas o uso de uma *annotation*, denominada *Role* (Papel), permite uma maior flexibilidade para associar qualquer elemento do modelo a um determinado ator.

Essa proposta é basicamente uma extensão do trabalho de Dijkman, mas utilizando a BPMN ao invés de uma notação genérica de modelagem de negócio. Os conceitos de *Step* (Passo) e *Role* (Papel) são os mesmos. Uma contribuição desse trabalho é a geração de diagramas de colaboração da UML. Esses são gerados a partir das *annotations* utilizadas no modelo em BPMN.

Um exemplo de mapeamento de modelo de processo de negócio para diagrama de caso de uso usando esse método pode ser visualizado na Figura 15 e na Figura 16.

Considerando que as atividades representadas na Figura 15 já estão agrupadas de acordo com o conceito de Passo e que o Papel Atendente foi definido e ligado aos Passos apresentados no diagrama, o diagrama de caso de uso resultante é formado por quatro casos de uso: Identificar

Forma de Pagamento, Receber Dinheiro ou Cheque, Passar o Cartão de Crédito e Empacotar Produto. O ator é o Papel Atendente. O diagrama de casos de uso resultante está apresentado na Figura 16.

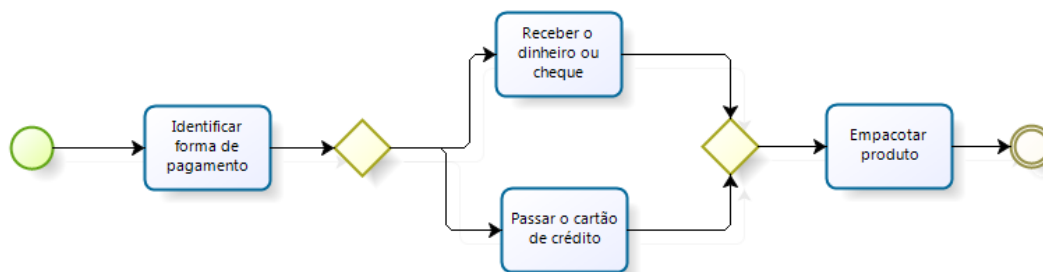


Figura 15 - Processo de negócio (traduzido do original) [LIE04]

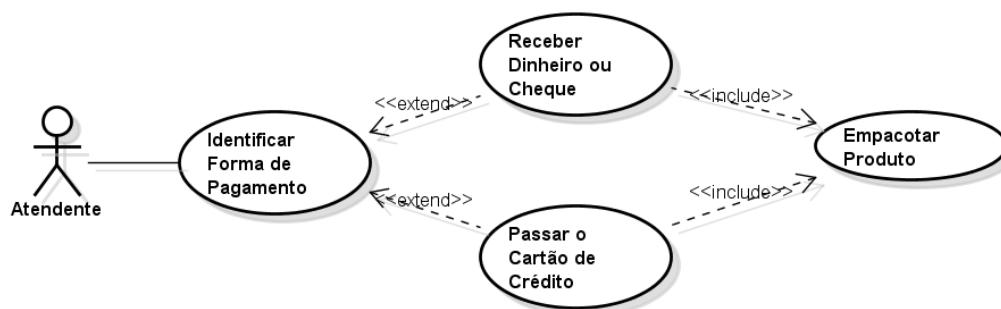


Figura 16 - Diagrama de caso de uso resultante (traduzido do original) [LIE04]

4.3 A METHOD OF LINKING BUSINESS PROCESS MODELING WITH INFORMATION SYSTEM DESIGN USING UML AND ITS EVALUATION (2007)

Em Okawa [OKA07] é apresentada uma proposta que utiliza uma notação genérica de BPM para modelagem de processo de negócio e diagramas de classes para modelagem das classes de projeto de sistema. Apesar de não utilizar casos de uso é interessante a análise desse trabalho porque o mesmo busca identificar diretamente classes de projeto de software em diagramas de processo de negócio, ao invés dos casos de uso.

O autor divide o processo todo em quatro estágios. No primeiro é feita uma análise dos processos de negócio atuais e criados modelos que representam esses processos. Uma simulação de cada processo de negócio é realizada em um segundo momento. Posteriormente são realizadas melhorias e correções nos processos analisados. Por último, é feita a transformação de cada processo definido para diagrama de classes da UML.

A transformação dos modelos de processo de negócio para diagrama de classes é realizada seguindo uma determinada política:

- 1) Associar cada processo de negócio definido a uma classe única do diagrama de classes.

- 2) Obter os dados necessários para a criação e execução do processo de negócio e tratar os mesmos como candidatos a atributos no diagrama de classes.
- 3) Tratar os elementos do modelo que descrevem fenômenos (ou atividades) como candidatos a métodos da classe.
- 4) Quando alguma especificação adicional parecer necessária nos passos 2 e 3 citados acima, adicionar outros atributos e métodos as classes.
- 5) Após os passos acima, verificar elementos comuns as classes que poderiam ser compartilhados. Nesse caso preparar uma classe para esses elementos compartilhados.

A Figura 17 apresenta um exemplo do processo de transformação. A partir do workflow que representa o processo de negócio é obtida a classe e os candidatos a atributos e métodos da classe. Após uma análise complementar é definida a classe resultante.

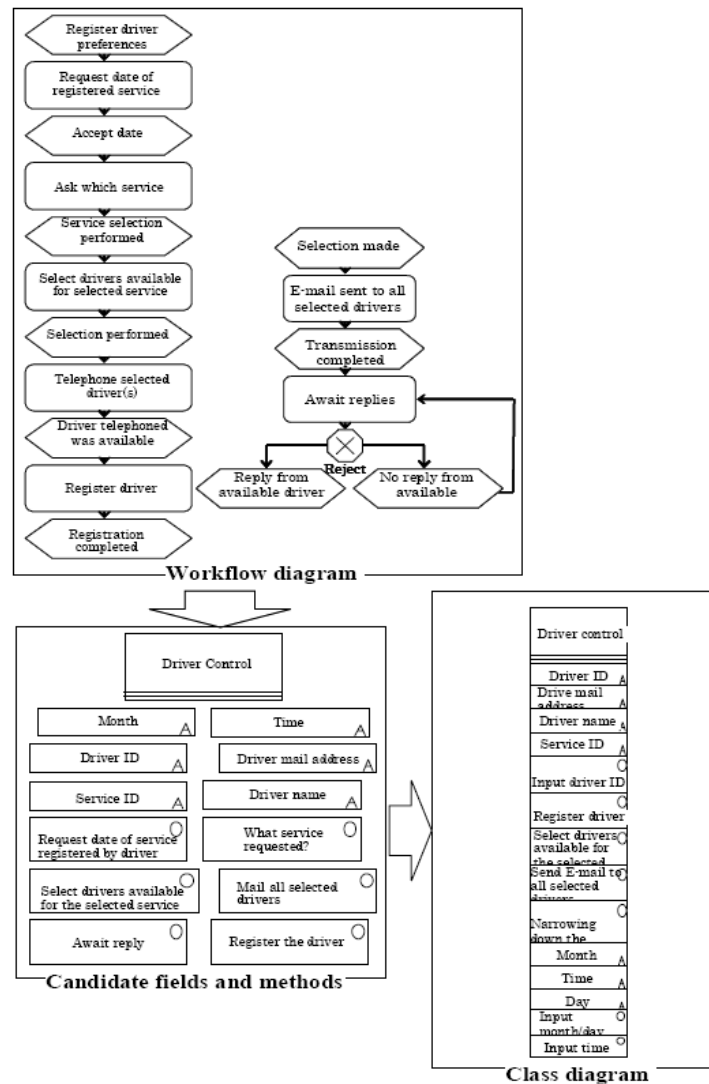


Figura 17 - Diagrama de classes obtido a partir do processo de negócio [OKA07]

4.4 IMPROVING REQUIREMENTS ANALYSIS THROUGH BUSINESS PROCESS MODELLING: A PARTICIPATIVE APPROACH (2008)

O estudo de Vara [VAR08] utiliza rótulos nos elementos de um diagrama BPMN para identificar e mapear esses elementos para uma estrutura semelhante a casos de uso. Mais precisamente ele utiliza um *template* para o detalhamento de um caso de uso baseado no conceito de *Task and Descriptions* (Tarefa e Descrição), proposto por Lauesen [LAU03]. Esse *template* é semelhante a uma descrição de caso de uso, mas com algumas informações adicionais. Dentre as informações contidas no *template* estão: nome da tarefa, responsável pela execução, gatilhos, pré-condições e pós-condições, dados de entrada/saída, detalhamento da interação entre usuário e o sistema. Além do procedimento de rotular os diagramas BPMN o autor também sugere a criação de documentos complementares para representação de informações que não podem ser exibidas graficamente, como regras de negócio. Esses documentos servem como complementos para as *Task and Description* geradas.

Os elementos do diagrama em BPMN são rotulados como: “O”, que indica um elemento fora do sistema e que não fará parte do mesmo; “IS”, indicando um item que fará parte do sistema e será controlado pelo mesmo sem intervenção humana; “U”, que identifica algo que será executado pelo usuário, sendo que essa execução implica em uma interação do usuário com o sistema. A figura 8 contém um exemplo de um diagrama BPMN com os rótulos definidos.

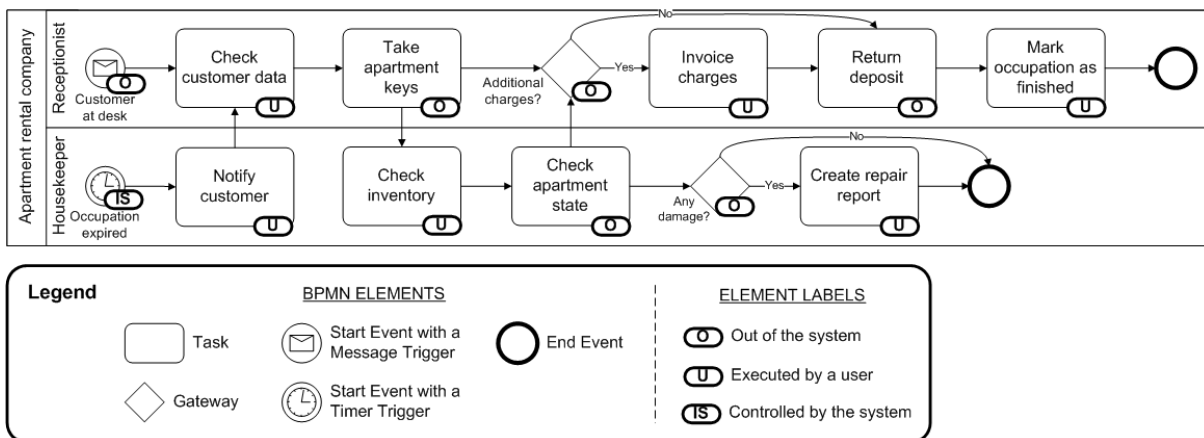


Figura 18 - Processo de negócio com rótulos [VAR08]

No método proposto as informações contidas no *template* gerado são oriundas do diagrama BPMN e nas documentações complementares. Em linhas gerais, a transformação de um modelo de negócio expresso em BPMN para um *template Task and Description* baseado em casos de uso utilizando essa técnica segue os seguintes passos:

- 1) Cada tarefa do diagrama BPMN rotulada com “U” é transformada em um *Task and Description* com o mesmo nome.

- 2) O participante da tarefa definido na raia do diagrama BPMN se torna o role do Task and Description.
- 3) Os eventos do tipo “IS” que precedem a tarefa são transformados em gatilhos.
- 4) Os *gateways* marcados como “IS” que precedem e sucedem a tarefa se tornam pré-condições e pós-condições, respectivamente.
- 5) Os objetos de dados no diagrama BPMN tornam-se os dados de entrada/saída.
- 6) A interação entre o usuário e sistema é definida partir do comportamento do usuário para executar a respectiva tarefa, contido nas documentações complementares.
- 7) As regras de negócio também são obtidas a partir dos documentos adicionais que descrevem cada tarefa.

A Figura 19 apresenta a *Task and Description* da tarefa *Notify Customer* do processo de negócio da Figura 18.

Business Process: Check out			
Task: Notify Customer		Role: Housekeeper	
Triggers			
• Occupation expired			
Preconditions			
-			
Postconditions			
Updated information stored in System			
Input		Output	
Data Object	State	Data Object	State
Occupation	Expired	Occupation	Expiration Notified
Customer	-	-	-
User intention		System responsibility	
		1. Show the apartments whose occupation has expired	
2. Select an apartment		3. Show the occupation and the customer data	
4. Phone the customer			
5. Notify the occupation expiration			
5. Mark the occupation expiration as notified			
		6. Record the notification of occupation expiration	
Business Rules			
1. Clients are to check out by 10.00 a.m. on the last day of their stay at the latest, and they are obliged to have vacated the apartment by this time.			

Figura 19 - Descrição da tarefa *Notify Customer* obtida transformando o processo de negócio [VAR08]

4.5 PATRONES PARA LA EXTRACCIÓN DE CASOS DE USO A PARTIR DE PROCESOS DE NEGOCIO (2009)

No trabalho de Javier [BAC09] são propostos padrões para extração de casos de uso a partir de modelos de processos de negócio em BPMN. Esse trabalho tem como principal contribuição a identificação de padrões para identificar tanto os casos de uso como seus relacionamentos (inclusão e extensão). Esses padrões abordam os diferentes tipos de *gateways* e quais relacionamentos entre casos de uso eles originam.

Esse trabalho utiliza o conceito de *Step* (Passo), proposto por Dijkman [DIJ02], assim como rotula as tarefas do BPMN entre Automáticas, Manuais ou Suportadas. As tarefas automáticas e suportadas podem se tornar casos de uso do sistema.

O processo para identificar os casos de uso ocorre da seguinte forma:

- 1) Os atores são identificados através das raiais.
- 2) Os passos (série de tarefas executadas sem interrupção) são identificados e transformados em casos de uso.
- 3) A partir desse ponto são identificados os relacionamentos entre os casos de uso através dos padrões apresentados, dentre os quais se podem destacar:
 - a. Casos de uso com *gateways* inclusivos/exclusivos dão origem a relacionamentos de inclusão ou extensão. O caminho padrão do *gateway* se torna um caso de uso de inclusão enquanto os outros se tornam casos de uso de extensão.
 - b. Casos de uso unidos por *gateways* paralelos se tornam casos de uso separados.
 - c. O caso de uso que ocorre após um *gateway* exclusivo utilizado para unir novamente o fluxo se torna um caso de uso de inclusão dos casos de uso dentro do *gateway*.
 - d. Se uma tarefa possui uma exceção associada a ela no diagrama BPMN então essa exceção dá origem a um novo caso de uso com um relacionamento de extensão.
- 4) Após o término da identificação dos casos de uso e dos seus relacionamentos o analista de requisitos deve refinar o resultado obtido.

A Figura 20 apresenta os casos de uso obtidos a partir do processo de negócio mostrado na Figura 21 utilizando os padrões sugeridos.

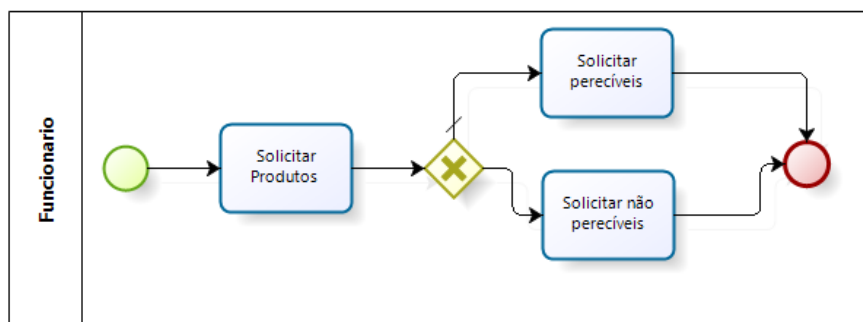


Figura 20 - Exemplo de processo de negócio em BPMN

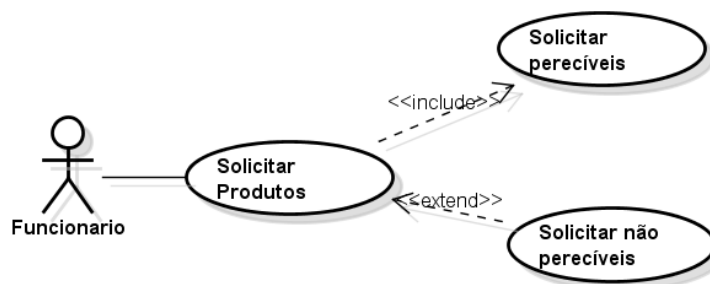


Figura 21 - Casos de uso obtidos a partir do processo de negócio

4.6 SEMI-FORMAL TRANSFORMATION OF SECURE BUSINESS PROCESSES INTO ANALYSIS CLASS AND USE CASE MODELS: AN MDA APPROACH (2010)

A proposta apresentada por Rodriguez [ROD10] parte de um modelo inicial em BPMN e transforma esse modelo em um diagrama de classes e em diagramas de casos de uso, tendo como foco requisitos de segurança em processos de negócio. Nesse trabalho são utilizadas regras bem definidas para a transformação entre os modelos dos diferentes níveis. Em um primeiro momento é realizada uma transformação entre o modelo BPMN para um diagrama de atividades da UML. Essa transformação ocorre em um mesmo nível de abstração. Posteriormente, o diagrama de atividades gerado dá origem a um diagrama de classes e a diagramas de casos de uso. Os diagramas gerados são refinados para incluir requisitos de segurança pré-definidos através de um framework proposto.

A transformação entre o diagrama em BPMN para o diagrama de classes e para o diagrama de casos de uso é relativamente simples. Conforme mencionado anteriormente, a primeira transformação realizada é entre um diagrama BPMN e um diagrama de atividades. Cada *Pool* ou *Lane* do diagrama em BPMN é transformado em uma Partição de Atividade no diagrama de atividades. Cada tarefa da BPMN é mapeada para uma Ação; cada fluxo de mensagem é mapeado para um fluxo de objeto; os objetos de dados são mapeados para *datastores* e o evento inicial é mapeado para um nodo de início.

Uma vez gerado o diagrama de atividades esse é transformado em um diagrama de classes e em um diagrama de casos de uso. Cada partição de atividade da origem a uma classe no diagrama de classes. Todas as ações dentro da partição são colocadas como operações das classes. Os *datastores* também dão origem a classes. Nos casos em que existem subpartições, essas são transformadas em classes que se relacionam com a classe da partição (que contém essa subpartição) através de um relacionamento de generalização.

Observando o diagrama BPMN da Figura 22, de acordo com o método proposto o diagrama de classes gerado teria uma classe expedição com quatro métodos, cada um deles correspondendo a uma das tarefas do BPMN. Deve-se reforçar que antes da geração do diagrama de classes se faz

necessária a geração do diagrama de atividades, que nesse caso conteria uma partição de atividade (expedição) e quatro ações (cada uma das quatro tarefas).

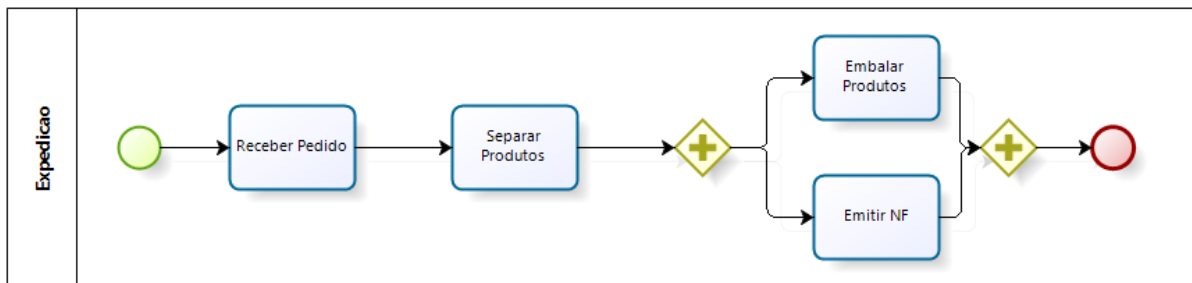


Figura 22 - Exemplo de processo de negócio em BPMN

A Figura 23 apresenta o diagrama de classe (com uma classe apenas) gerado a partir do modelo de processo de negócio da Figura 22.

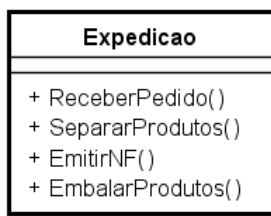


Figura 23 - Exemplo de classe gerada a partir da transformação

A geração do diagrama de casos de uso também ocorre a partir do diagrama de atividades obtido do BPMN, conforme mencionado anteriormente. Para fazer a transformação do diagrama de atividades para casos de uso as seguintes regras são utilizadas: cada partição de atividade dá origem a um ator com o mesmo nome; cada ação origina um caso de uso, que é relacionado à partição em que está contido. No caso de existirem subpartições, essas dão origem a atores com relação de especialização. As subpartições se tornam atores especializados enquanto que a partição principal se torna ator generalizado.

Seguindo as regras citadas acima o diagrama de caso de uso gerado a partir do modelo da Figura 22 pode ser conferido observando a Figura 24.

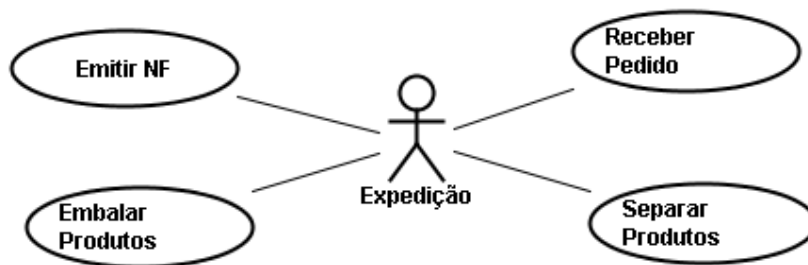


Figura 24 - Diagrama de casos de uso obtido na transformação

4.7 CONSIDERAÇÕES SOBRE OS TRABALHOS RELACIONADOS

Os trabalhos apresentados nas seções anteriores mostram um esforço da comunidade científica em propor soluções que busquem aproximar os modelos em nível de sistema dos modelos em nível de negócio.

No trabalho de Dijkman [DIJ02] é utilizado um diagrama de atividades da UML para modelar o processo de negócio. Dijkman é um dos precursores nesse tema de identificação de casos de uso a partir de processos de negócio. Os conceitos propostos são utilizados em outros trabalhos [VAR08, BER09, LIE04].

Um dos problemas do trabalho de Dijkman é justamente a notação utilizada para a modelagem do processo de negócio. Atualmente a BPMN oferece maior poder de expressão para elementos em nível de negócio do que um simples diagrama de atividades da UML.

O trabalho de Liew [LIE04] é uma evolução do trabalho anterior de Dijkman à medida que utiliza a BPMN para desenhar o negócio. Entretanto, vários elementos da BPMN não são considerados, como os diferentes tipos de tarefas, eventos e *gateways*. Na geração dos casos de uso apenas o diagrama e interações do fluxo básico são geradas. É necessária uma intervenção no diagrama BPMN para identificar os *Roles* (que dão origem aos atores) e os *Steps* (que dão origem aos casos de uso), assim como a marcação do tipo da tarefa, manual, automática ou suportada.

A transição entre o nível de negócio para o nível de sistema também é assunto do trabalho de Okawa [OKA07]. A diferença é que o objetivo não é a geração de casos de uso de sistema, mas sim das classes de projeto de software. O princípio utilizado é que cada processo de negócio pode representar uma classe e as tarefas que ocorrem dentro do processo tornam-se candidatas a métodos dessa classe. Essa transformação direta entre um artefato de negócio e um artefato em nível de projeto de sistema pode gerar um diagrama de classes de projeto que não reflete a realidade necessária para implementação do sistema. Por isso que o autor reforça que os elementos gerados nas classes são apenas candidatos a propriedades e métodos, devendo ser validados pelo analista ou projetista.

O trabalho de Vara [VAR08] acrescenta informações relacionadas à descrição das interações que ocorrem no nível dos casos de uso. Na realidade é usado um artefato que estende as informações de um caso de uso, sendo esse artefato chamado de *Task and Description* [LAU03]. A *Task and Description* é um template de descrição de caso de uso contendo elementos adicionais, como dados de entrada/saída, regras de negócio, gatilho, entre outros. Apesar de esse método gerar uma descrição mais detalhada ele acaba não tendo a expressão visual que se obtém ao utilizar os diagramas de casos de uso e seus relacionamentos.

Em Berrocal [BER09] é apresentado um conjunto de padrões para extração de casos de uso em modelos de processos de negócio em BPMN. As idéias utilizadas são basicamente as propostas

por Dijkman. A contribuição desse trabalho está em tratar diferentes tipos de *gateways* e também a ocorrência de exceções ligadas a tarefas. Apesar de fazer uso da notação BPMN os diferentes tipos de tarefas possíveis na BPMN não são abordados, assim como os diferentes tipos de eventos. A descrição dos casos de uso gerados também não é trabalhada. Os padrões propostos são limitados e acabam gerando diagramas de casos de uso com relacionamentos incorretos. É o que ocorre quando um caminho padrão de um *gateway* exclusivo origina um caso de uso com relacionamento de inclusão (Figura 21) com o caso de uso que precede o *gateway*. Esse relacionamento está incorreto porque implicaria dizer que o caso de uso incluído é sempre executado. No *gateway* exclusivo apesar de se ter um caminho padrão não necessariamente esse caminho é sempre percorrido. Ele será o percorrido apenas quando nenhuma condição associada a outro caminho for verdadeira ou quando a sua própria condição for verdadeira. Um relacionamento de inclusão não expressa esse tipo de situação.

O último trabalho estudado é o de Rodriguez [ROD10]. Esse trabalho apresenta um método que parte de um diagrama BPMN e dá origem tanto a diagrama de classes como diagrama de casos de uso, que depois são refinados utilizando requisitos específicos de segurança. Inclusive são apresentadas definições formais das transformações realizadas, sendo utilizada uma linguagem de transformação entre modelos, a *Query View Transformation (QVT)* [OMG05]. A transformação ocorre em duas etapas. Primeiro o modelo em BPMN é transformado em um diagrama de atividades da UML. Em seguida o diagrama de atividades é transformado em diagrama de classes e diagrama de casos de uso. Os autores não deixam claro porque é feita essa transformação para o diagrama de atividades ao invés de gerar diretamente os diagramas de classes e casos de uso. Uma contribuição desse trabalho está na formalização das transformações permitindo uma automatização das mesmas, bem como no uso dos princípios da MDA [OMG03] para transformações entre modelos de diferentes níveis de abstração. Entretanto, os modelos gerados não são completos. No caso do diagrama de classes não é mencionado como os atributos são obtidos. Nos diagramas de casos de uso não são tratados itens de descrição dos casos de uso assim como relacionamento entre os que foram obtidos a partir do modelo inicial em BPMN. Os diferentes tipos de tarefas da BPMN bem como os eventos também não são abordados. O método proposto tem escopo limitado a requisitos de segurança sendo que todos os modelos gerados na transformação são refinados com informações complementares.

Uma conclusão que se pode obter a partir dos estudos relacionados é que não há uma proposta consolidada que permite gerar um artefato em nível de sistema a partir de um artefato em nível de negócio. O que se pode perceber é que os esforços estão mais evoluídos na geração de casos de uso de sistema [DIJ02, LIE04, VAR08, BAC09, ROD10] e que a notação utilizada nos

trabalhos mais recentes para a modelagem de processo de negócio é a BPMN [LIE04, VAR08, BAC09, ROD10].

O fato de não haver uma proposta consolidada para realizar essa transformação indica que ainda existem evoluções que podem ser feitas nesse assunto, dentre as quais se destacam:

- Tratamento de diferentes tipos de tarefas da BPMN;
- Tratamento dos eventos de um diagrama BPMN;
- Diferentes possibilidades de transformação dos *gateways*;
- Geração da descrição dos casos de uso (ou parte dela);

A proposta apresentada no próximo capítulo tem como objetivo apresentar um método que atenda as evoluções mencionadas acima.

5 MÉTODO PARA TRANSFORMAÇÃO DE BPMN EM CASOS DE USO DA UML

O objetivo desse capítulo é apresentar um método de transformação de elementos de modelo de processos de negócio usando diagrama BPMN para elementos do modelo de casos de uso, esse último formado pelo diagrama e descrições iniciais dos casos de uso. Para que isso seja possível é necessário que o modelo de processos de negócio expresso em BPMN represente o negócio considerando tarefas que devem ser executadas por um sistema de informação a ser modelado. O método consiste basicamente em analisar as diferentes alternativas possíveis de transformação dos elementos da BPMN para elementos do modelo de casos de uso.

O método proposto apresenta um caminho que deve ser percorrido para que se atinja o resultado final. O resultado final é o modelo de casos de uso do sistema. O método consiste em três atividades:

- i) Verificação do modelo de processos de negócio de origem;
- ii) Transformação dos elementos da BPMN para casos de uso;
- iii) Refinamento das descrições dos casos de uso.

A verificação do modelo de origem consiste em analisar se o modelo de processos de negócio utilizado como origem permite a identificação de elementos de sistema, ou seja, se o modelo pode ser usado como base para identificação de requisitos de um sistema.

Um processo de negócio que define somente tarefas manuais sem prever a utilização de um sistema não possibilita identificar quais elementos estariam relacionados com o sistema. A BPMN permite diferenciar tarefas manuais com tarefas relacionadas com um sistema. O modelo utilizado como origem na transformação deve possuir essa distinção entre os tipos de tarefas.

A transformação dos elementos da BPMN para casos de uso é a principal atividade do método proposto. Essa atividade consiste em analisar cada elemento da BPMN verificando como o mesmo pode ser transformado em um elemento de caso de uso do sistema. Por exemplo, raias da BPMN podem dar origem a atores de casos de uso, assim como tarefas de sistema podem dar origem aos casos de uso.

A última atividade consiste em um ajuste das transformações realizadas anteriormente, principalmente na descrição dos casos de uso gerados. Por exemplo, a transformação pode dar origem a um caso de uso contendo apenas a identificação do seu ator e uma pré-condição. Nesse caso é necessário um refinamento dessa descrição. A atividade de refinamento não ocorre necessariamente após a atividade ii. Elas podem ocorrer concomitantemente para cada transformação realizada.

As condições para se transformar o modelo de processo de negócio assim como a descrição das transformações estão explicadas no restante do capítulo, que está organizado da seguinte forma:

a Seção 5.1 apresenta o formato de caso de uso que é obtido como resultado das transformações; a Seção 5.2 apresenta considerações sobre a validação do modelo de processo de negócio utilizado como origem; a Seção 5.3 apresenta as definições das transformações entre os elementos de negócio e os de sistema, assim como observações relativas ao refinamento dos casos de uso; finalmente a Seção 5.4 apresenta as considerações finais sobre o método.

5.1 ESTRUTURA DO CASO DE USO RESULTANTE

Conforme mencionado no Capítulo 3, sobre modelagem de sistemas, os casos de uso são utilizados para expressar de forma mais detalhada os requisitos de um sistema. Um caso de uso é uma descrição textual e estruturada da interação que ocorre entre um (ou mais) ator(es) e o sistema para atingir um determinado objetivo de interesse desse ator ou desses atores. Segundo Booch [BOO05], um caso de uso especifica o comportamento de um sistema e é uma descrição de um conjunto de ações, incluindo suas variantes, que produz um resultado observável para um ator. Larman [LAR04] explica que o modelo de casos de uso é formado pelas descrições dos casos de uso e pode incluir um diagrama de casos de uso da UML.

O diagrama de casos de uso é utilizado na UML para visualização dos casos de uso com seus respectivos relacionamentos. É um artefato que fornece uma visão geral dos requisitos do sistema e como esses requisitos estão associados aos atores do sistema. Já o detalhamento de um caso de uso é um documento textual contendo a descrição detalhada das ações que ocorrem para que um ator atinja um determinado objetivo definido pelo caso de uso.

Por ser textual a forma de se descrever um caso de uso varia muito, assim como variam os elementos considerados em cada formato de descrição. Cockburn [COC05] comenta que o mesmo caso de uso escrito por uma pessoa pode ser diferente do escrito por outra pessoa. O estilo de escrita depende da finalidade do caso de uso. Um caso de uso válido em um projeto pode não ser adequado em outro.

Por esse motivo é necessário definir qual o formato do caso de uso a ser trabalhado, ou mais especificamente, quais os elementos que compõem o caso de uso. O formato de caso de uso utilizado nesse trabalho está definido na seção a seguir.

5.1.1 FORMATO DO DETALHAMENTO DO CASO DE USO

Essa seção descreve o formato de caso de uso obtido a partir das transformações. Esse formato está baseado nos formatos propostos pelo RUP e Cockburn [COC05], descritos no Capítulo 3.

O princípio básico das transformações é extrair do diagrama de processo de negócio os elementos do caso de uso. Entretanto, apenas alguns elementos podem ser obtidos analisando o diagrama BPMN. O restante da descrição deve ser completado posteriormente pelo profissional que está fazendo a análise dos casos de uso, geralmente um analista de sistemas. Por esse motivo o formato utilizado para a descrição é uma simplificação dos formatos do RUP e de Cockburn, utilizando campos dos dois formatos, cujo conteúdo desses campos pode ser obtido a partir de informações contidas nos modelos de processo de negócio. O elemento Acionador é específico do formato de Cockburn. O elemento Lista de Atores é uma simplificação do Ator principal e demais atores. Os demais campos são comuns aos formatos do RUP, de Cockburn e de outros autores [JAC92, KRU00].

A Tabela 2 apresenta os elementos do formato proposto.

Elemento	Descrição
Nome	Nome do caso de uso.
Descrição	Breve descrição do caso de uso.
Lista de Atores	Lista com os atores que participam do caso de uso.
Pré-condições	Condições que devem ser verdadeiras no sistema para que o caso de uso inicie.
Acionador	Ação que dispara o caso de uso, podendo ser um evento de tempo.
Fluxo básico de eventos	Descrição textual, passo a passo e numerada que define a interação básica, ou o cenário típico de sucesso, entre os atores e o sistema para atingir um objetivo.
Fluxos Alternativos	Fluxos alternativos ao fluxo básico definido. Devem ter nomes, passos numerados e a indicação de quando eles ocorrem.
Pontos de Extensão	Lista de locais dentro do fluxo de eventos (básico ou alternativo) cujo comportamento adicional é inserido usando relacionamentos de extensão. Geralmente está associado a uma condição para que o relacionamento aconteça.
Pós-condições	Descreve o que deve ser verdadeiro após a execução bem sucedida do caso de uso.

Tabela 2 - Formato do caso de uso

O acionador indica uma ação que ocorre e que é responsável pelo início do caso de uso. Pode ser um evento de tempo, como um horário programado do dia para execução de um caso de uso.

O fluxo básico contém uma lista de passos que representa a troca de informações entre o ator, ou atores, e o sistema. Essa lista de passos deve ser numerada [CON01].

Existe apenas um fluxo básico para cada caso de uso. Os outros fluxos possíveis devem ser colocados no elemento Fluxos Alternativos. Cada fluxo alternativo deve possuir um nome e ter seus passos numerados. Os próprios fluxos alternativos devem ser separados e numerados [CON01], por exemplo, FA1 – <nome do fluxo alternativo 1>, <FA2 – nome do fluxo alternativo 2>. Também deve ser indicada a situação em que cada fluxo pode ocorrer. Por exemplo: FA1 – Cliente não cadastrado. (Esse fluxo ocorre quando o sistema não encontra o cliente informado no passo <número do passo>).

Os pontos de extensão são locais dentro dos fluxos que indicam que um comportamento adicional deve ser inserido utilizando outro caso de uso. Os pontos de extensão geralmente estão associados a condições que determinam quando a extensão acontece. Essas condições também devem ser descritas no caso de uso.

5.2 VERIFICAÇÃO DO MODELO DE NEGÓCIO A SER TRANSFORMADO

Um modelo de processos de negócio representado em um diagrama BPMN pode ser utilizado para modelar processos de negócio de uma organização para diversas finalidades. Por exemplo, a organização deseja mapear, apenas para fins de documentação, os processos de negócio atualmente utilizados. Outro exemplo pode ser de uma organização que deseja mapear os processos atuais e a partir desses fazer uma análise de melhoria nesses processos.

O método apresentado nesse trabalho procura identificar casos de uso de sistema em processos de negócio. Sendo assim, para que o mesmo seja aplicado é necessário que o processo de negócio representado identifique quais tarefas serão objetos de um sistema a ser desenvolvido. O método proposto auxilia o analista de sistemas a modelar os casos de uso de sistema existentes no modelo de processo de negócio definido.

5.2.1 ELEMENTOS SUPORTADOS

O método proposto nesse trabalho tem sua aplicação limitada a processos que ocorram dentro de uma mesma organização. Na definição da BPMN esses processos são chamados de *Private (Internal) Business Process* [BPM09], ou processos internos de negócio. Esses processos contêm apenas uma organização participante e a troca de informações ocorre somente dentro dessa organização.

A BPMN apresenta uma série de elementos gráficos que podem ser utilizados na modelagem dos processos. Alguns desses elementos são comuns de aparecerem desde processos simples até mais complexos. A análise das transformações realizada nesse trabalho aborda os elementos da BPMN que são utilizados para modelar processos internos de negócio. Os elementos das quatro categorias (objetos de fluxo, objetos de conexão, partições e artefatos) são considerados na transformação.

Apenas os sub-processos, fluxos de mensagem e anotações não possuem uma análise de transformação. O sub-processo não é considerado explicitamente. Entretanto a transformação pode ser feita considerando os elementos dentro desse sub-processo como se fosse um processo em separado. Já o fluxo de mensagem não é abordado porque os processos considerados ocorrem dentro de uma mesma organização, não sendo necessária a troca de mensagem entre diferentes

participantes do negócio (*pools*). A anotação nada mais é que um mecanismo que permitir adicionar uma informação ao modelo. Nesse trabalho não foi atribuído um uso específico para esse elemento.

Cada elemento representa uma classe do metamodelo da BPMN, conforme comentado no Capítulo 2. Para cada um desses elementos o método apresenta alternativas de transformação. A Tabela 3 contém os elementos da BPMN que são suportados pelo método proposto. Nada impede que o diagrama contenha outros elementos, entretanto esses não são considerados nas alternativas de transformação.

Elemento da BPMN	Descrição
Raia (Lane)	As raias são utilizadas para representar diferentes categorias dentro de uma organização. Pode representar um papel ou um setor que é responsável pelas tarefas dentro da raia. As raias são partes de uma partição (pool). Um pool representa uma organização.
Participante (Pool)	Um pool representa um participante do processo. Representa uma organização. Processos com mais de um pool são processos considerados Business to Business (B2B).
Tarefa (Task)	A tarefa é uma atividade atômica e representa uma ação realizada que não pode (ou não é interessante) dividi-la em outras. Os tipos de tarefa abordados no método de transformação são: tarefas manuais, tarefas de usuário, tarefas de serviço.
<i>Gateways</i>	Elementos utilizados para controlar o fluxo do processo. O método aborda três tipos de <i>gateways</i> : exclusivo, inclusivo e paralelo.
Eventos	Evento é algo que ocorre no início, meio ou final do processo.
Fluxo de Sequência	Mostra a ordem em que as tarefas ocorrem no processo.
Associação	Permite associar artefatos aos objetos de fluxo do processo.
Artefato Objeto de Dados	Representa dados usados ou gerados por elementos do processo. Nesse trabalho são usados apenas diagramas com objetos de dados ligados a tarefas.
Artefato Grupo	Permite agrupar atividades. A BPMN não define um uso específico desse elemento, servindo o mesmo apenas para agrupamento visual de elementos. Entretanto, nesse trabalho é atribuída uma semântica específica para esse artefato, conforme está explicado na seção que define as transformações dos elementos.

Tabela 3 - Elementos da BPMN analisados

Apesar de o elemento Participante (*Pool*) constar na tabela de elementos suportados, o método não considera a troca de mensagens entre os participantes. Cada Participante é analisado separadamente como se fossem processos diferentes.

5.3 TRANSFORMAÇÃO DOS ELEMENTOS DA BPMN PARA CASOS DE USO

Essa seção apresenta como ocorrem as transformações entre elementos de um modelo de processos de negócio em BPMN para elementos de um modelo de sistema utilizando casos de uso. Para cada elemento da BPMN presente na lista de elementos suportados é analisado como esse pode ser transformado em um elemento de um caso de uso.

Segundo os princípios de transformação de modelos da MDA [OMG03] essa transformação de elementos de um modelo em um nível de abstração de negócio para elementos em nível de abstração de sistema é considerada uma transformação vertical. A partir de um modelo de origem em um nível de abstração específico é verificado como os elementos desse modelo são mapeados para elementos de outro modelo em outro nível de abstração. Para cada elemento do metamodelo de origem é verificado como esse elemento pode ser mapeado para um elemento do metamodelo de destino.

As próximas seções apresentam as possibilidades de transformação de cada elemento da BPMN para elementos de casos de uso da UML.

5.3.1 PARTICIPANTE E RAIA

O escopo desse trabalho está limitado a diagramas representando processos dentro de uma mesma organização. Esses diagramas possuem apenas um participante (*pool*). O participante representa uma entidade de negócio, por exemplo, a organização cujo processo está sendo analisado. Também pode representar o processo que está sendo modelado.

A raia é utilizada para organizar e categorizar as atividades do processo. A BPMN não condiciona o uso da raia a um significado específico, ficando a forma de utilização da mesma a critério do modelador.

Apesar de, por definição, a raia não indicar necessariamente quem realiza a tarefa, nesse trabalho é adotado como semântica para a raia a indicação da pessoa (ou grupo de pessoas), setor da empresa ou mesmo outro sistema que é responsável por executar as tarefas contidas dentro da raia.

Essa premissa é necessária para que seja possível transformar a raia em um ator na modelagem utilizando casos de uso. A idéia de transformar raia em ator não é novidade. Uma abordagem semelhante a essa pode ser encontrada em outros trabalhos relacionados [ROD10, ODE03]. Nesses trabalhos são utilizadas as raias de um diagrama de atividades da UML para dar origem aos atores dos casos de uso.

Conforme definido uma raia indica a pessoa, grupo, setor ou mesmo sistema externo responsável pela execução de uma determinada tarefa. Esse conceito é análogo ao de um ator no contexto de casos de uso da UML. Sendo assim é possível definir um mapeamento em que uma raia da BPMN é transformada em um ator no modelo de casos de uso. A Figura 25 ilustra essa transformação.

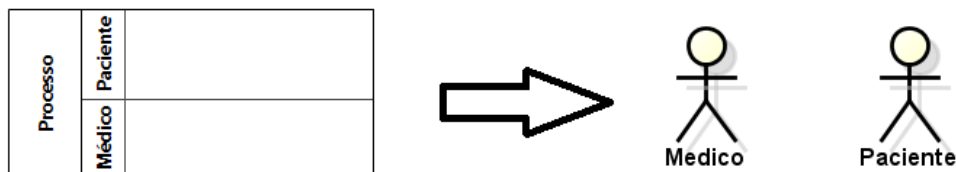


Figura 25 – Transformação de Raia para Ator

Nos metamodelos a transformação ocorre entre a classe *Lane* e a classe *Actor*.

A BPMN permite a utilização de raias aninhadas. Essa disposição das raias, uma dentro da outra, pode indicar generalizações entre as mesmas. Utilizando a semântica de raia definida e mapeando as mesmas para atores de casos de uso a existência de uma raia dentro de outra indica um relacionamento de generalização de atores.

5.3.2 TAREFA MANUAL

Uma tarefa manual representa tarefas executadas no processo sem o apoio de nenhum software. Sendo assim esse tipo de atividade não tem um correspondente no sistema a ser modelado. Não há nenhum caso de uso ou elemento dentro dos casos de uso para associar com uma tarefa manual. Dessa forma não é definida nenhuma transformação de tarefas manuais de um BPMN para casos de uso.

5.3.3 TAREFA DE USUÁRIO

Representa uma tarefa executada por uma pessoa utilizando algum recurso do sistema. A tarefa representa um objetivo a ser atingido no momento da realização da mesma. Na modelagem de sistemas um caso de uso representa um objetivo do ator no sistema. Dessa forma é possível fazer uma relação entre tarefa de usuário e caso de uso do sistema. Tanto a tarefa de usuário na BPMN quanto o caso de uso do sistema na UML representam objetivos a serem atingidos a partir da realização da tarefa ou do caso de uso, respectivamente.

Utilizando esse princípio é possível realizar a transformação de tarefas de usuário da BPMN para casos de uso da UML. O ator do caso de uso é o ator identificado através da raia onde a tarefa está contida, conforme explicado na seção que aborda a transformação da raia.

Nos metamodelos essa transformação ocorre entre a classe *User Task* do BPMN para *Use Case* da UML. O atributo nome da tarefa da classe *User Task* do BPMN se torna o nome do caso de uso. Já a associação entre caso de uso e ator deriva da associação entre a *User Task* e a *Lane* que ocorre a partir da classe *Flow Object* (representada no metamodelo do Capítulo 3).

Um exemplo de uma tarefa que representa um caso de uso é Cadastrar Pedido. Essa tarefa quando definida como tarefa de usuário da BPMN representa um objetivo do usuário representado na raia ao utilizar o sistema. O ator desse caso de uso está identificado através da raia que contém essa tarefa. A Figura 26 ilustra esse exemplo.

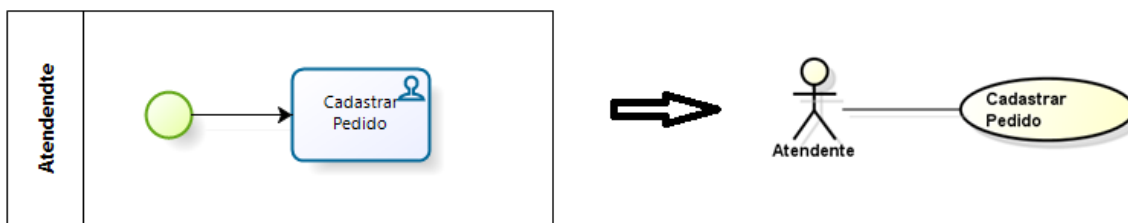


Figura 26 - Tarefa para Caso de Uso

Existe outra possibilidade em que a tarefa de usuário não é transformada diretamente em um caso de uso, mas sim em um passo do caso de uso. Isso ocorre quando a tarefa está dentro de um elemento Grupo. Essa transformação é abordada detalhadamente mais adiante.

5.3.4 TAREFA DE SERVIÇO

A diferença entre tarefa de serviço e tarefa de usuário da BPMN está no fato de que a tarefa de serviço é executada por um sistema ou *webservice* sem a intervenção do usuário.

A transformação de uma tarefa de serviço pode seguir a mesma abordagem utilizada na tarefa de usuário, ou seja, essa tarefa dá origem a um caso de uso e o ator do caso de uso é a raia que contém a tarefa. Entretanto em uma tarefa de serviço não há intervenção do usuário, não existindo uma interação entre o ator e o sistema. O ator nesse caso representa apenas o interessado no objetivo do caso de uso ou mesmo quem inicia o caso de uso, mas não interage diretamente com o mesmo. A Figura 27 mostra um exemplo dessa transformação. Nesse exemplo o diagrama BPMN contém duas tarefas. Uma tarefa de usuário Cadastrar Pedido e outra tarefa de serviço Atualizar Ranking de Pedidos. Essa tarefa de serviço representa uma rotina do sistema que atualiza informações da situação dos pedidos nas filiais. A transformação dessa tarefa em caso de uso se torna interessante para identificar quem é o interessado na realização disso ou mesmo quem inicia esse caso de uso.

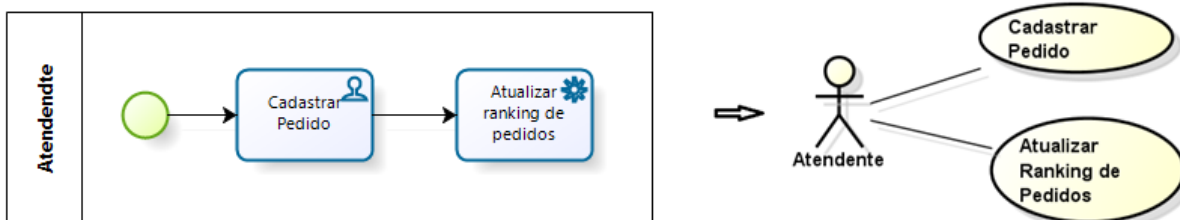


Figura 27 - Tarefa de Serviço para Caso de Uso

Nessa situação a transformação nos elementos do metamodelo é a mesma que ocorre para tarefa de usuário, mas nesse caso se utiliza a classe *Service Task*.

Outra forma de transformar tarefa de serviço em elementos de casos de uso é fazer com que a tarefa seja um passo de um caso de uso. Nessa situação a tarefa se torna um passo, executado pelo sistema, dentro do caso de uso originado por outra tarefa (nesse caso de usuário) que precede essa tarefa de serviço. O exemplo da Figura 28 ilustra essa situação.

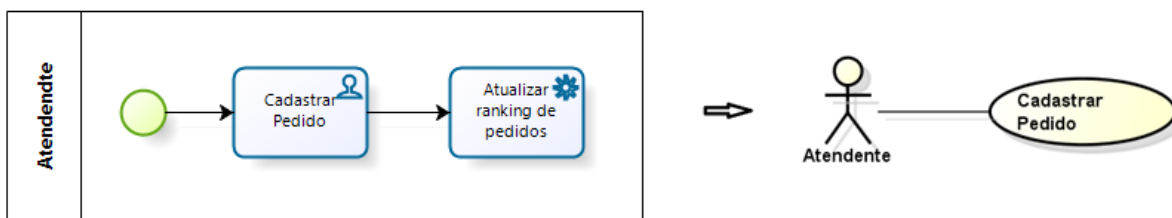


Figura 28 - Tarefa de Serviço para passo de um caso de uso

Nesse exemplo a tarefa não aparece no diagrama de casos de uso. Ela é mapeada para um passo do fluxo básico da descrição do caso de uso originado pela tarefa de usuário que precede a tarefa de serviço, no exemplo a tarefa Cadastrar Pedido. O passo do fluxo básico indica a execução da tarefa pelo sistema, por exemplo, “Sistema atualiza ranking de pedidos”. O resultado dessa transformação pode ser conferido na Figura 29 que mostra a descrição obtida para o caso de uso Cadastrar Pedido. Em termos de metamodelo a transformação ocorre entre as classes *Service Task* da BPMN e *Step* do metamodelo de casos de uso.

Nome do Caso de Uso:	Cadastrar Pedido
Descrição:	
Lista de Atores:	Atendente
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	...
Fluxos alternativos:	Sistema atualiza ranking de pedidos.
Pontos de Extensão:	
Pós-condições:	

Figura 29 - Descrição do caso de uso Cadastrar Pedido

É importante observar que a descrição obtida não é completa. Essa descrição contém apenas os elementos que podem ser identificados a partir do diagrama BPMN. As descrições devem ser refinadas com base em outras informações necessárias a esse refinamento. Não é objetivo das transformações propostas nesse trabalho obter a descrição completa de um caso de uso, mas sim auxiliar na identificação desses casos de uso e alguns elementos da sua descrição quando esses estão presentes nos diagramas BPMN.

Existe ainda a situação em que essa tarefa de serviço é algo executado por um sistema externo ao sistema que está sendo analisado. Nessa situação a recomendação é colocar essa tarefa

de serviço em outra raia sendo que essa raia deve representar o sistema externo. Dessa forma a tarefa de serviço se torna um caso de uso que tem como ator o sistema externo.

Com base nas situações descritas acima verifica-se que a transformação do elemento tarefa de serviço da BPMN para elementos de caso de uso da UML possui as seguintes possibilidades:

- a) A tarefa de serviço é transformada em um caso de uso. O ator do caso de uso é a raia que contém a tarefa.
- b) A tarefa de serviço é transformada em um passo de um caso de uso originado por uma tarefa de usuário que precede a tarefa de serviço.
- c) A tarefa de serviço se torna um caso de uso que tem como ator um sistema externo.

A primeira alternativa é mais adequada quando se deseja mostrar no diagrama de casos de uso os interessados na realização do objetivo da tarefa de serviço. A segunda alternativa pode ser usada quando não é interessante representar essa tarefa como um caso de uso separado, sendo a mesma um simples passo de outro caso de uso. A terceira é usada quando a tarefa é realizada por um sistema externo.

5.3.5 GATEWAY

Gateways são elementos utilizados para determinar o caminho dos fluxos de seqüência definidos. Um diagrama de casos de uso não apresenta a seqüência de execução dos casos de uso, por esse motivo a transformação de um *gateway* em BPMN para elementos de caso de uso não é direta, sendo necessária uma verificação de qual o *gateway* usado e o contexto das tarefas relacionadas com ele no processo modelado.

5.3.5.1 GATEWAY EXCLUSIVO

Um *gateway* exclusivo é usado em partes do diagrama onde existe a necessidade de seguir um determinado caminho do fluxo quando uma determinada condição for atendida. Essa condição determina qual será o fluxo de seqüência a ser seguido. Apenas um dos caminhos do *gateway* pode ser seguido. A transformação de um *gateway* exclusivo para um caso de uso pode ocorrer de diversas formas.

Um *gateway* exclusivo pode dar origem a relacionamentos de extensão, inclusão e especialização entre os casos de uso originados pelas tarefas envolvidas no *gateway*. Outra possibilidade é o *gateway* dar origem a passos de determinado caso de uso. Essa última alternativa é abordada mais adiante na transformação do elemento Grupo.

Possibilidade 1 – Gateway Exclusivo resulta em relacionamentos de extensão de casos de uso

Uma das formas de transformar *gateway* exclusivo em casos de uso é utilizando relacionamentos de extensão entre os casos de uso originados a partir das tarefas do *gateway*. Essa alternativa é utilizada por Liew [LIE04] para transformar qualquer tipo de *gateway*.

A tarefa de usuário que precede o *gateway* dá origem a um caso de uso normal (ou caso de uso base). As tarefas de usuário que estão dentro do *gateway* (após a bifurcação) se tornam casos de uso de extensão do caso de uso normal. As condições indicadas no *gateway* são utilizadas nos pontos de extensão. A Figura 30 apresenta essa transformação.

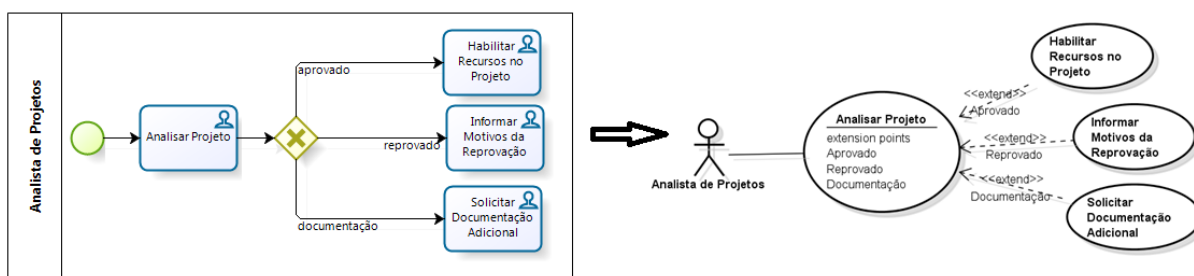


Figura 30 - Gateway Exclusivo para relacionamentos de extensão

Uma restrição dessa abordagem é que a ocorrência de um ponto de extensão não é obrigatória em um caso de uso. Um caso de uso pode ter vários pontos de extensão, porém em um determinado cenário (execução de um caso de uso) nenhum dos pontos de extensão é utilizado. Nessa situação sempre um dos casos de uso de extensão seria executado. Uma forma de contornar isso é colocar uma condição padrão (*default*) no *gateway* e esse caminho padrão se torna parte do fluxo básico do caso de uso normal originado pela tarefa que precede o *gateway*. As demais condições se tornam pontos de extensão e as tarefas dentro do *gateway* dão origem a casos de uso de extensão que estendem o caso de uso normal.

Por exemplo, uma tarefa de usuário Analisar Projeto precede um *gateway* exclusivo que tem três caminhos, um com a tarefa de Habilitar Recursos no Projeto que ocorre quando o projeto é aprovado e é o caminho padrão, outro Informar Motivos da Reprovação que ocorre quando o projeto é reprovado e outro Solicitar Documentação Adicional que ocorre quando o responsável necessita de outros documentos para definir a situação. A tarefa Analisar Projeto se torna um caso de uso normal cujo fluxo básico contém os passos relativos à aprovação do projeto, que é o caminho padrão. Esse caso de uso possui relacionamentos de extensão, um para o caso de uso Informar Motivos da Reprovação e outro para o caso de uso Solicitar Documentação Adicional. Os pontos de extensão são os locais no fluxo básico do caso de uso normal nos quais o comportamento é estendido para os outros casos de uso. Cada ponto de extensão está associado a uma condição, que é

a condição obtida a partir do *gateway*. O caso de uso normal (ou caso de uso base) se torna o acionador dos casos de uso estendidos. Esse exemplo está ilustrado na Figura 31.

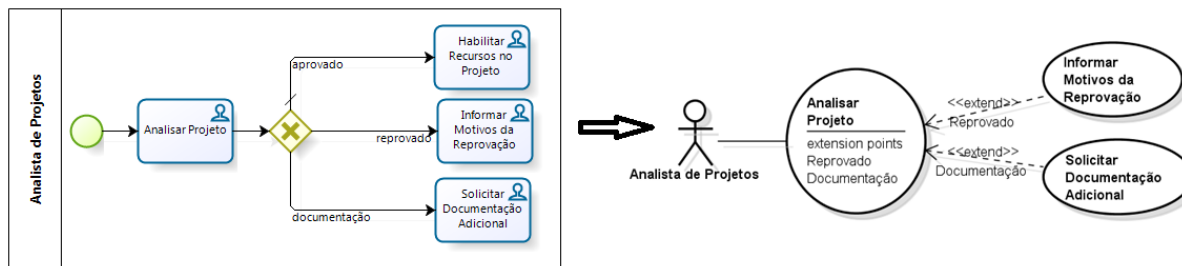


Figura 31 - Gateway Exclusivo com condição padrão para relacionamento de extensão

A descrição do caso de uso Analisar Projeto considerando a aprovação dentro do fluxo básico pode ser conferida na Figura 32.

Nome do Caso de Uso:	Analisar Projeto
Descrição:	
Lista de Atores:	Analista de Projetos
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	... Passo <i>m</i> Analista de Projetos verifica dados do projeto Passo <i>n</i> Analista de Projetos aprova projeto
Fluxos alternativos:	
Pontos de Extensão:	Ponto de Extensão 1 - Reprovado No passo “Analista de Projetos verifica dados do projeto” se o projeto for reprovado o caso de uso Informar Motivos da Reprovação deve ser executado. Ponto de Extensão 2 –Documentação No passo “Analista de Projetos verifica dados do projeto” se for necessário documentação adicional o caso de uso Solicitar Documentação Adicional deve ser executado.
Pós-condições:	

Figura 32 - Descrição caso de uso Analisar Projeto

Utilizando esse recurso de colocar um dos caminhos do *gateway* no fluxo básico se resolve a questão relacionada com a não obrigatoriedade de execução de um dos relacionamentos de extensão.

Entretanto, há outra situação. Assim como não é obrigatória a execução de um dos relacionamentos de extensão também não há impeditivos para que todos ou alguns deles sejam executados. Essa possibilidade de execução de um ou mais pontos de extensão não reflete o que é representado em um *gateway* exclusivo. Analisando apenas o diagrama de caso de uso não fica indicado que apenas um dos pontos de extensão pode ser executado. Voltando ao exemplo anterior, da Figura 30, se analisar apenas o diagrama de caso de uso nada indica que apenas um dos casos de uso de extensão pode ser executado. Para isso seria necessário indicar que esses casos de uso de extensão são mutuamente exclusivos. Uma forma de fazer isso é colocar um comentário no diagrama de casos de uso ou então uma restrição no detalhamento do caso de uso normal.

No nível de metamodelos essa transformação ocorre entre as classes *Exclusive Gateway* da BPMN, com as classes *Use Case*, *Extend* e *ExtensionPoint* da UML.

Possibilidade 2 – *Gateway* Exclusivo resulta em relacionamentos de inclusão de casos de uso

Outra possibilidade de transformação de *gateway* exclusivo em casos de uso pode ser utilizando relacionamentos de inclusão entre casos de uso. Essa alternativa aparece no trabalho de Dijkman [DIJ02].

Nessa transformação a tarefa que precede o *gateway* origina um caso de uso que é incluído pelos casos de uso originados a partir das tarefas que estão dentro do *gateway*. Os casos de uso originados por tarefas dentro do *gateway* incluem o comportamento do caso de uso originado pela tarefa que precede o *gateway*. Dessa forma fica expressada a obrigatoriedade de execução do caso de uso que precede o *gateway* assim como o seu reaproveitamento por outros casos de uso, que é o que justifica um relacionamento de inclusão.

No exemplo citado anteriormente, de aprovação do projeto, a tarefa Analisar Projeto se torna um caso de uso que é incluído pelos casos de uso Habilitar Recursos no Projeto, Informar Motivos da Reprovação e Solicitar Documentação Adicional, conforme a Figura 33.

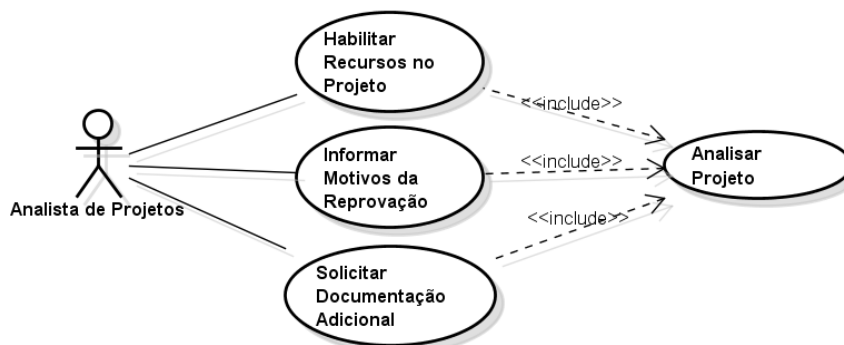


Figura 33 - *Gateway* Exclusivo para Relacionamento de Inclusão

Essa alternativa gera um diagrama de caso de uso que mostra que para executar determinados casos de uso é obrigatória a execução de outro caso de uso, o Analisar Projeto. Isso reflete o que está no *gateway*. Entretanto, se perde um pouco da semântica contida no diagrama BPMN. Um *gateway* exclusivo deixa claro que apenas um dos caminhos deve ser seguido em cada situação. Se analisar o diagrama de casos de uso gerado essa informação não fica clara. O ator é associado a três casos de uso, mas não fica indicado que cada um deles só é utilizado em uma determinada situação que depende justamente de algo que ocorre no caso de uso de Analisar Projeto (caso de uso que foi incluído). O diagrama resultante não expressa a necessidade das condições para que ocorram determinados casos de uso.

Em relação à descrição dos casos de uso, aqueles originados a partir de tarefas dentro do *gateway* devem ter como primeiro passo do fluxo básico a indicação da inclusão do caso de uso originado a partir da tarefa que precede o *gateway*. No exemplo usado os casos de uso Habilitar Recursos no Projeto, Informar Motivos da Reprovação e Solicitar Documentação Adicional devem ter como primeiro passo do seu fluxo básico a indicação de que deve ser executado o caso de uso Analisar Projeto. Já o caso de uso Analisar Projeto tem como acionador os três casos de uso anteriores.

Nos metamodelos a transformação utiliza a classe *Exclusive Gateway* da BPMN e as classes *Use Case* e *Include* da UML.

Possibilidade 3 – Gateway Exclusivo resulta em relacionamentos de especialização de casos de uso

Uma terceira possibilidade de transformação é utilizando relacionamentos de generalização entre casos de uso. O caso de uso originado pela tarefa que precede o *gateway* dá origem a um caso de uso pai e os casos de uso dentro do *gateway* se tornam casos de uso especializados. O caso de uso pai é uma generalização dos casos de uso que estão dentro do *gateway*. Todos os passos do caso de uso pai são comuns e necessários para a ocorrência de qualquer um dos casos de uso filho. Essa abordagem também permite expressar que apenas um dos casos de uso filho é executado em cada cenário.

Usando o exemplo anterior o caso de uso Aprovar Projeto se torna o caso de uso generalizado, ou caso de uso pai. Os casos de uso originados pelas tarefas dentro do *gateway*, Habilitar Recursos no Projeto, Informar Motivos da Reprovação e Solicitar Documentação Adicional se tornam casos de uso filho desse caso de uso pai, conforme observado na Figura 34. O caso de uso pai se torna o acionador dos casos de uso filhos.

Essa abordagem consegue expressar a semântica de exclusividade de um caminho do *gateway* através do relacionamento de generalização de casos de uso. A descrição do caso de uso pai é feita com fluxos alternativos, um para cada situação especializada, sendo que em cada cenário somente uma, ou nenhuma delas deve ocorrer.

Essa alternativa atende a diversas situações em que o *gateway* exclusivo é usado, justamente porque os casos de uso especializados são mutuamente exclusivos em cada cenário de execução do caso de uso pai.

Nos metamodelos as transformações ocorrem nas classes *Exclusive Gateway* da BPMN e na classe *Use Case* da UML, utilizando o relacionamento de especialização.

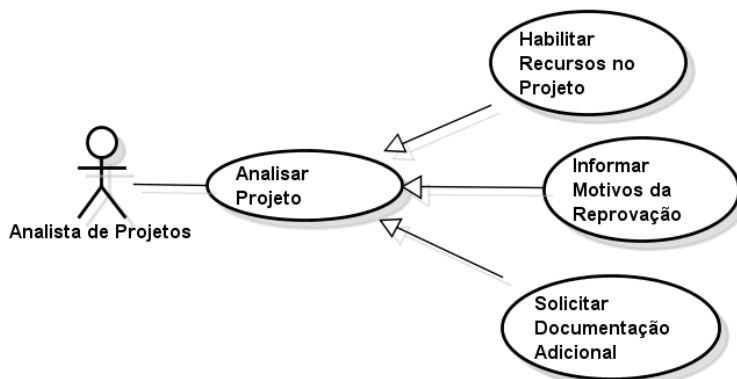


Figura 34 - Gateway Exclusivo para Casos de Uso com especialização

5.3.5.2 GATEWAY INCLUSIVO

Gateways inclusivos se diferem dos exclusivos por permitirem que mais de um caminho seja seguido a partir das condições definidas no *gateway*. Mais de uma condição do *gateway* pode ser verdadeira, assim como deve ter pelo menos uma condição que seja verdadeira para que o processo siga o seu fluxo.

As alternativas de transformação do *gateway* inclusivo para casos de uso são semelhantes às utilizadas para o *gateway* exclusivo. Podem ser originados casos de uso com relacionamento de extensão ou casos de uso com relacionamento de inclusão. Entretanto, a transformação para casos de uso com relacionamento de especialização não é possível, conforme está explicado adiante.

Possibilidade 1 – Gateway Inclusivo resulta em relacionamentos de extensão de casos de uso

Um caso de uso pode ter vários pontos de extensão, que geralmente possuem condições que determinam que um caso de uso de extensão seja disparado. A existência das extensões não implica dizer que elas sempre são executadas. Pode ocorrer um cenário onde todos os pontos de extensão são verdadeiros, assim como outro cenário em que nenhum é verdadeiro e também situações onde alguns são verdadeiros e outros não. Uma situação semelhante ocorre no *gateway* inclusivo. Todas as condições podem ser verdadeiras ou apenas algumas delas. Entretanto, pelo menos uma tem que ser definida como padrão (*default*) para que o fluxo siga adiante.

Para se transformar um *gateway* inclusivo em um caso de uso com relacionamentos de extensão deve-se criar um caso de uso normal a partir da tarefa que precede o *gateway*. As condições expressas no *gateway* se tornam condições dos pontos de extensão do caso do caso de uso normal. As tarefas dentro do *gateway* se tornam casos de uso de extensão que são disparados a partir do ponto de extensão definido.

Se o *gateway* possui um fluxo padrão definido então as tarefas desse fluxo se tornam passos do fluxo básico do caso de uso originado pela tarefa que precede o *gateway*. As demais tarefas dos outros fluxos do *gateway* se tornam os casos de uso de extensão. Os casos de uso de extensão têm como elemento acionador o caso de uso que eles estendem. O raciocínio é o mesmo utilizado na transformação do *gateway* exclusivo.

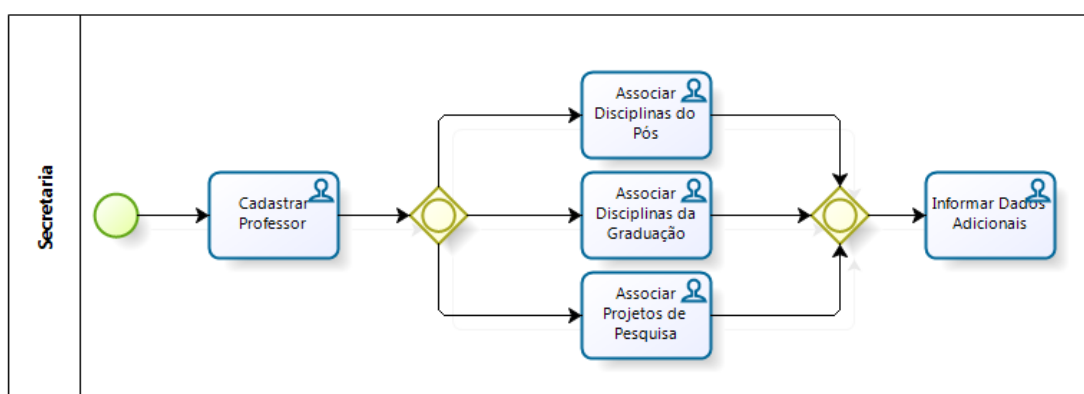


Figura 35 - Exemplo de processo de negócio para registro de novo professor

A Figura 35 apresenta um exemplo de diagrama BPMN que mostra o processo de inclusão de um novo professor em uma universidade.

Esse processo tem um *gateway* inclusivo que determina tarefas que devem ser executadas dependendo do tipo de professor cadastrado. Se o professor for de graduação então ele deve ter as disciplinas da graduação associadas a ele. Se ele for de pós-graduação então deve ter as disciplinas da pós-graduação. E se ele for pesquisador devem ser associados projetos de pesquisa a ele. Pode ocorrer que ele seja apenas professor da graduação, ou da pós ou pesquisador, mas também pode ocorrer que ele tenha as três atribuições ou duas delas. Essa situação é expressa com um *gateway* inclusivo.

Extraindo casos de uso obtém-se o diagrama mostrado na Figura 36, onde os casos de uso de extensão são as tarefas dentro do *gateway* e o caso de uso normal é a tarefa que precede o *gateway*. Os pontos de extensão estão ligados às condições definidas no *gateway*, que nesse exemplo podem ser *flags* indicando se o professor é de graduação e/ou pós e/ou pesquisador.

O caso de uso originado após o *gateway* inclusivo utilizado para unir o fluxo é um caso de uso normal associado ao respectivo ator, que pode ter como pré-condição o resultado do caso de uso originado pela tarefa que precede o *gateway* inclusivo utilizado para dividir o fluxo.

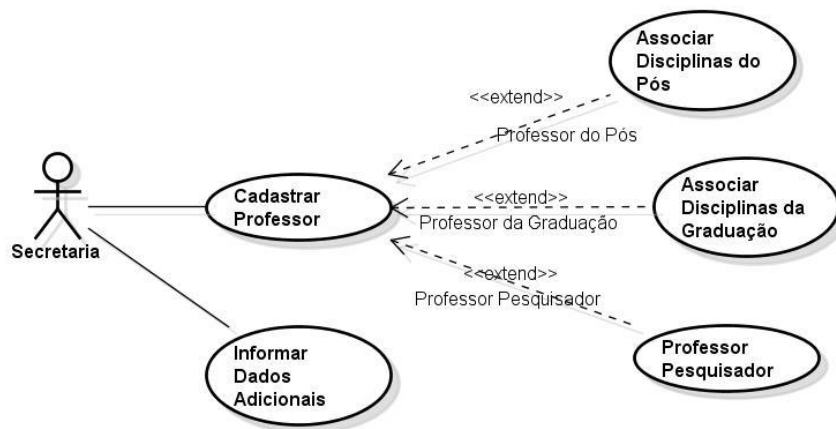


Figura 36 - Transformação de *gateway* inclusivo para relacionamento de extensão

As classes do metamodelos envolvidas nessa transformação são as classes *Inclusive Gateway* da BPMN, *Use Case* da UML, *Extended* e *ExtensionPoint* da UML.

Possibilidade 2 – *Gateway* Inclusivo resulta em relacionamentos de inclusão de casos de uso

A estratégia é a mesma utilizada na transformação do *gateway* exclusivo para casos de uso com relacionamentos de inclusão. Cada caso de uso originado dentro do *gateway* se torna um caso de uso que inclui o comportamento do caso de uso originado pela tarefa que precede o *gateway*.

O problema dessa abordagem é que não fica expressa no diagrama a existência de condições necessárias para que os casos de uso originados a partir das tarefas dentro do *gateway* ocorram. A Figura 37 mostra o resultado da transformação do processo de negócio de registro de professor utilizando essa abordagem.

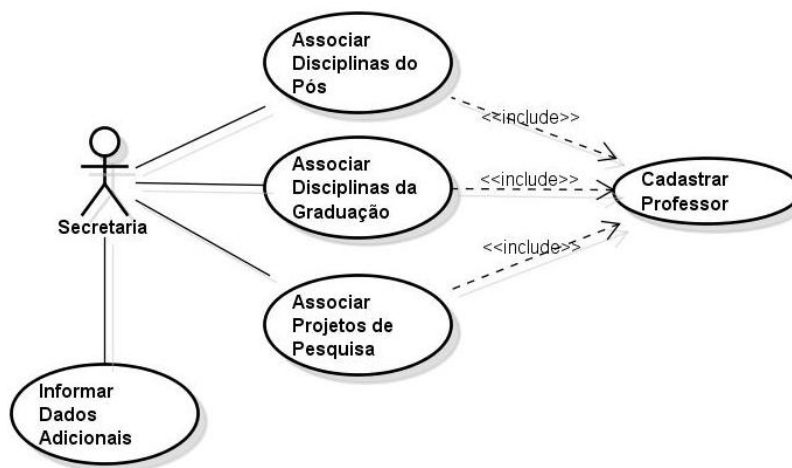


Figura 37 - Transformação do *gateway* inclusivo para relacionamento de inclusão

Observando a transformação realizada percebe-se que não fica clara a existência de condições para que cada um dos casos de uso associados ao ator *Secretaria* ocorra. Analisando

apenas para o diagrama pode-se concluir que Associar Disciplinas do Pós, Associar Disciplinas da Graduação e Associar Projetos de Pesquisa não têm nenhuma condição necessária para serem utilizados.

No nível dos metamodelos as transformações ocorrem utilizando as classes *Inclusive Gateway* da BPMN, *Use Case* e *Include* da UML.

Já a transformação de um *gateway* inclusivo para casos de uso com relacionamento de generalização de casos de uso não é possível, ao contrário do que acontece no *gateway* exclusivo. Os casos de uso filho são mutuamente exclusivos a cada execução do caso de uso pai e isso não configura a realidade expressa no *gateway* de inclusão. No exemplo anterior se fosse gerado um caso de uso pai a partir da tarefa Cadastrar Professor e cada tarefa de cada fluxo do *gateway* originasse casos de uso filhos então implicaria dizer que a cada execução do caso de uso Cadastrar Professor apenas um dos casos de uso filho seria executado, o que estaria inconsistente com o processo de negócio que deu origem aos casos de uso.

5.3.5.3 GATEWAY PARALELO

Um *gateway* paralelo é utilizado para criar fluxos paralelos sem verificar nenhuma condição. Dessa forma o processo segue pelos vários fluxos definidos e depois é sincronizado utilizando novamente o *gateway* paralelo para sincronização. A tarefa que sucede o *gateway* utilizado para sincronização só é executada quando todas as tarefas paralelas são executadas.

Ao contrário do que ocorre com os *gateways* exclusivos e inclusivos a transformação dos elementos de um *gateway* paralelo para casos de uso não fica adequada utilizando relacionamentos de extensão de casos de uso. Isso porque a execução das tarefas contidas no *gateway* não depende de nenhuma condição, sendo executada de forma paralela e independente. Nesse caso não existiriam condições associadas aos pontos de extensão para relacionar o caso de uso normal com os casos de uso de extensão.

A transformação de um *gateway* paralelo para casos de uso pode ocorrer de duas formas. Uma delas é através de relacionamentos de inclusão entre os casos de uso originados. A outra é simplesmente a geração de casos de uso independentes, ou seja, sem relacionamentos no diagrama.

Possibilidade 1 – Gateway Paralelo resulta em relacionamentos de inclusão de casos de uso

O uso de um relacionamento de inclusão de casos de uso implica na obrigatoriedade de realização do caso de uso incluído para que o caso de uso normal atinja seu objetivo. Em um *gateway* paralelo, uma vez que o *gateway* divide o fluxo é necessária a execução de todas as tarefas dentro do *gateway*. Partindo dessa comparação é possível transformar a tarefa que precede o

gateway em um caso de uso normal e as tarefas dentro do *gateway* se tornam casos de uso que são incluídos no caso de uso normal.

A Figura 38 mostra um exemplo de processo de negócio referente à realização de um pedido. O vendedor registra o pedido. A partir desse ponto ocorrem duas tarefas em paralelas e independentes: Imprimir Fatura e Gerar Recibo de Entrega. Uma vez finalizadas as tarefas é autorizado o envio do pedido.

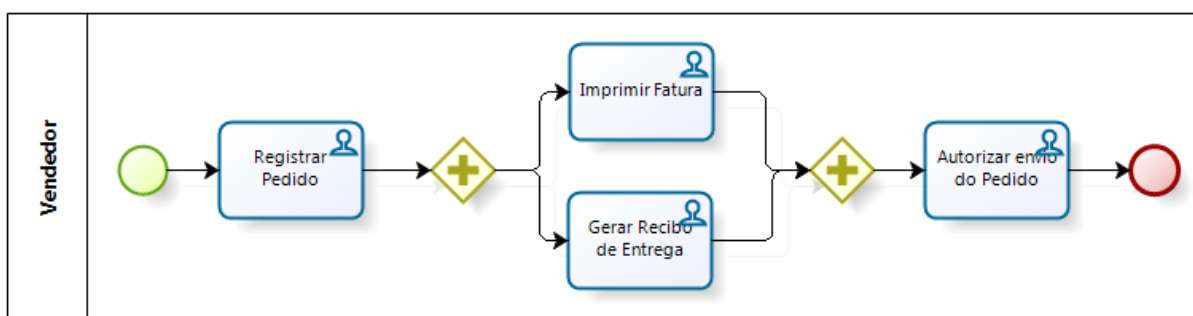


Figura 38 - Exemplo de processo de negócio de realização de pedido

Transformando esse processo de negócio em casos de uso com relacionamento de inclusão obtém-se um caso de uso normal Registrar Pedido e dois casos de uso de inclusão: Imprimir Fatura e Gerar Recibo de Entrega, ligados ao caso de uso Registrar Pedido. O caso de uso que ocorre após o *gateway* se torna um caso de uso normal sem relação com os demais, tendo apenas como pré-condição o resultado do caso de uso registrar pedido, conforme pode ser observado na Figura 39.

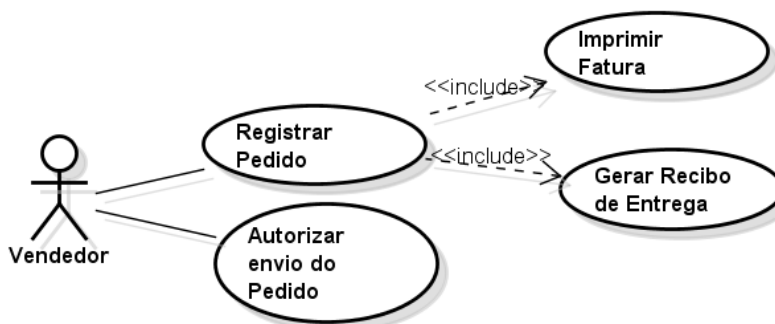


Figura 39 - Gateway paralelo pra casos de uso com relacionamento de inclusão

Essa transformação também pode variar. Nada impede que o caso de uso originado pela tarefa que ocorre depois do *gateway* inclua os casos de uso originados pelas tarefas dentro do *gateway*. Nessa situação o caso de uso Autorizar Envio do Pedido incluiria os casos de uso Imprimir Fatura e Gerar Ordem de Compra. Já o caso de uso Registrar Pedido ficaria como um caso de uso normal sem relacionamento com os outros no diagrama.

Nos metamodelos as transformações ocorrem nas classes *Paralel Gateway* da BPMN, *Use Case* e *Include* da UML.

Utilizando essa abordagem se consegue expressar no diagrama que um caso de uso necessita de outros para que atinja o seu objetivo, refletindo o que ocorre no *gateway* inclusivo, onde as tarefas dentro do *gateway* devem ocorrer obrigatoriamente para que o fluxo siga. Entretanto, a finalidade de um relacionamento de inclusão de um caso de uso não é apenas para indicar obrigatoriedade de execução de outro caso de uso. Ao se utilizar a inclusão de casos de uso deve ser considerada a reutilização dos casos de uso incluídos. Essa situação não ocorreu no exemplo anterior. Quando isso não ocorre o diagrama de caso de uso originado fica inconsistente com a semântica de relacionamento de inclusão da UML, pois existe um caso de uso incluído em outro, mas que não é reaproveitado por outro caso de uso. A solução para essa situação seria não utilizar os relacionamentos de inclusão, conforme explicado a seguir.

Possibilidade 2 – Gateway Paralelo resulta em Casos de Uso independentes

Essa alternativa é a mais simples. As relações existentes no *gateway* não dão origem a relacionamentos entre os casos de uso. Cada tarefa relacionada com o *gateway* se torna um caso de uso independente. Entretanto, é possível obter algumas informações do *gateway* que podem ser utilizadas na descrição dos casos de uso originados. Por exemplo, os casos de uso originados pelas tarefas dentro do *gateway* possuem como pré-condição o resultado do caso de uso que precede o *gateway*. Já o caso de uso que ocorre depois do *gateway* de sincronização tem como pré-condição o resultado dos casos de uso que estão dentro do *gateway*.

Realizando essa transformação no processo de negócio de registro de pedidos obtêm-se quatro casos de uso:

- Registrar Pedido
- Imprimir Fatura
- Gerar Recibo de Entrega
- Autorizar Envio de Pedido

O caso de uso Imprimir Fatura tem como pré-condição o resultado do caso de uso Registrar Pedido, que pode ser “Pedido Registrado”, ou a geração de algum artefato associado à tarefa do BPMN (a transformação de artefatos para elementos de casos de uso é abordada mais adiante). Nesse caso estaria configurada uma pré-condição para a realização do caso de uso Imprimir Fatura ou mesmo do Gerar Recibo de Entrega. A Figura 40 mostra o diagrama resultante dessa transformação. No diagrama as pré-condições não são exibidas. Esses elementos aparecem apenas na descrição dos casos de uso.

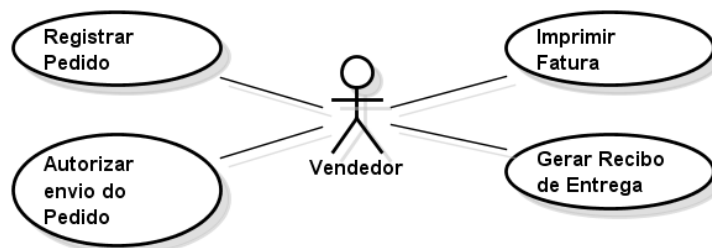


Figura 40 - Gateway Paralelo para Casos de Uso sem relacionamento entre eles

As pré-condições não são utilizadas para expressar seqüenciamento entre casos de uso. A utilização da pré-condição indica uma situação (um estado) que deve ser verdadeiro para que o caso de uso seja executado. No exemplo apresentado a pré-condição simplesmente indica a necessidade da existência de algo para que ele seja executado.

A Figura 41 abaixo mostra como exemplo a descrição do caso de uso Imprimir Fatura.

Nome do Caso de Uso:	Imprimir Fatura
Descrição:	
Lista de Atores:	Vendedor
Pré-condições:	Pedido registrado
Acionador:	
Fluxo básico de eventos:	
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 41 - Descrição do caso de uso Imprimir Fatura

5.3.6 EVENTO

Evento na definição da BPMN é algo que ocorre durante o processo. Existem três tipos de eventos: evento inicial, evento intermediário e evento final. Esses três tipos podem se dividir em outros tipos dependendo da sua causa ou do seu resultado. Os eventos podem ocorrer de diferentes formas em diferentes locais do processo e por esse motivo a transformação dos mesmos para elementos de casos de uso pode acontecer de várias maneiras.

Esse trabalho apresenta alternativas de como os eventos podem dar origem a elementos de casos de uso. São abordados eventos com causa temporal e condicional. Esses dois tipos de causa podem ser mapeados para elementos de casos de uso conforme é mostrado a seguir. Outros tipos de causa de eventos como mensagem, cancelamento, *link*, entre outros, estão fora do escopo desse trabalho por serem usados para processos com mais de um *pool* ou porque, em uma primeira análise, não foram encontradas relações com os elementos de casos de uso. Essa análise dos demais tipos de causas de eventos pode ser tema de um trabalho futuro.

5.3.6.1 EVENTO INICIAL

Um evento de início em um diagrama BPMN indica onde inicia a execução do fluxo do processo. Um diagrama pode ter apenas um evento de início. Esses eventos são acionados por diferentes causas. Cada evento inicial pode ser de um tipo, que depende da causa que acionou esse evento. Um evento de início sem causa definida não tem transformação para elemento de casos de uso, pois não acrescenta nenhuma informação específica a algum caso de uso obtido a partir de uma tarefa do processo. Por exemplo, um evento de início sem definição nenhuma ligado a uma tarefa de usuário não é levado em consideração na transformação dessa tarefa para casos de uso, pois não se tem nenhuma outra informação associada a esse evento. Ele é um elemento único do modelo em nível de negócio, usado apenas para indicar um ponto de início do processo.

Entretanto se for adicionada uma causa a esse evento essa causa pode ser usada na geração do caso de uso obtido a partir da tarefa associada a esse evento de início. Um evento de início com causa temporal, por exemplo, pode ser usado para identificar o acionador de um caso de uso. Segundo Cockburn [COC05] um acionador de um caso de uso especifica o evento que faz o caso de uso começar, podendo ser um evento de tempo. Nessa situação a restrição temporal associada ao evento é utilizada como acionador do caso de uso originado. O ator do caso de uso é o interessado no objetivo do mesmo, mas o acionador do caso de uso é o próprio tempo.

A forma de se descrever um evento de tempo em casos de uso é objeto de discussão na área. Alguns autores [COC05, RAT02] consideram o evento de tempo como um acionador do caso de uso, sendo o ator o principal interessado, mas não o tempo. Outros autores [AMB05, MAC01] colocam o tempo como ator do caso de uso, podendo esse ser um ator secundário. A definição da UML não esclarece a questão, definindo apenas que ator pode ser um usuário ou um sistema externo. No trabalho de Bari [BAR02] é apresentado um estudo que estende o conceito de ator da UML definindo um novo tipo de ator, denominado ator temporal.

No presente trabalho é utilizado um formato de caso de uso que faz a distinção entre acionador e ator de caso de uso. Devido a isso o mais apropriado é colocar o evento de tempo como acionador do caso de uso, sendo que o ator permanece o que foi identificado através da raia que contém a tarefa ligada ao evento e que deu origem ao caso de uso.

A Figura 42 apresenta um exemplo dessa transformação. Nesse exemplo existe um evento de início temporal que é responsável por disparar uma tarefa todos os dias às oito horas da manhã. No diagrama de casos de uso gerado não aparece nenhuma informação relativa ao evento. Porém, a descrição desse caso de uso contém como elemento acionador a causa temporal do evento de início, conforme exibido na Figura 43.

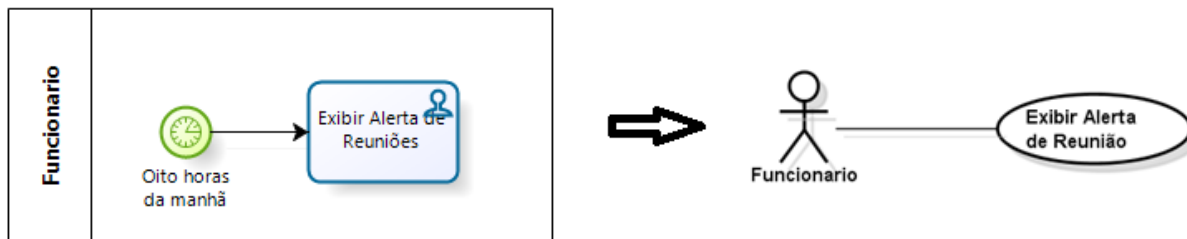


Figura 42 - Transformação de diagrama com evento de início temporal

Nome do Caso de Uso:	Exibir Alerta de Reunião
Descrição:	
Lista de Atores:	Funcionário
Pré-condições:	
Acionador:	Caso de uso iniciado pelo sistema todos os dias às oito horas da manhã.
Fluxo básico de eventos:	
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 43 - Descrição do caso de uso

Outro tipo de causa para um evento de início pode ser uma condição ou regra de negócio definida. Nesse caso existe um evento de início que para ser disparado necessita que uma determinada condição definida seja verdadeira. No instante que a condição definida é verdadeira a tarefa ligada a esse evento de início começa sua execução. Se a tarefa ligada ao evento de início é transformada em um caso de uso e a execução dessa tarefa está ligada a uma condição ou regra de negócio definida pode-se afirmar que o caso de uso também se inicia somente quando essa condição é verdadeira. Entretanto somente a especificação da condição não deixa claro quem é que aciona o caso de uso.

Quando a causa é temporal, como no exemplo anterior, está indicado que o próprio sistema aciona o caso de uso. Quando a causa é outra condição qualquer não fica claro quem é o acionador. Nessa situação o mais adequado é transformar essa condição definida em uma pré-condição do caso de uso. Uma pré-condição não inicia o caso de uso, ela apenas indica um estado que deve ser verdadeiro para que o caso de uso se inicie. Nessa situação o evento é a pré-condição e não o acionador do caso de uso.

Um exemplo de evento de início condicional é “Cliente com cadastro vencido”. Transformando a tarefa ligada a esse evento para um caso de uso não fica claro quem aciona o caso de uso. Entretanto, a pré-condição para início do mesmo está estabelecida. Nesse exemplo o caso de uso possui uma pré-condição que define que o cadastro do cliente deve estar vencido.

Em relação aos metamodelos a transformação utiliza as classes *Start Event* da BPMN e a classe *Use Case* e *Constraint* (para as pré-condições) da UML.

5.3.6.2 EVENTO INTERMEDIÁRIO

Eventos intermediários ocorrem entre o início e final de um processo. Uma das finalidades de se utilizar eventos intermediários é para modelar situações de espera no processo. Também são utilizados para tratar exceções relacionadas a alguma tarefa. Nesse trabalho são abordados eventos intermediários sem causa definida, eventos intermediários temporais e eventos intermediários condicionais.

Semelhantemente ao que ocorre com eventos iniciais sem causa definida um evento intermediário sem causa não possui uma relação com os elementos de casos de uso. Um evento assim colocado no diagrama tem uma função meramente ilustrativa do negócio, não tendo conteúdo associado ao mesmo que possa ser utilizado na transformação para sistema.

Eventos intermediários possuem uma particularidade em relação a eventos iniciais e finais: eles podem ocorrer no fluxo do processo ou anexados a uma tarefa. Quando ocorrem no fluxo do processo eles são precedidos por algum outro elemento de fluxo da BPMN. E quando aparecem anexados a uma tarefa indicam algo que pode ocorrer durante a execução da tarefa.

A localização do evento – no fluxo do processo ou anexado a uma tarefa – é importante para a análise de como ele pode ser transformado em elemento de caso de uso. Por esse motivo a definição dessas transformações está dividida conforme essas situações. Em cada uma delas o evento pode ser temporal ou condicional, conforme explicado abaixo.

Situação 1 – Eventos intermediários no fluxo do processo

Um evento intermediário temporal que ocorre no fluxo do processo geralmente é utilizado para indicar pontos de espera do processo, ou seja, indicam a espera para que uma determinada restrição temporal definida aconteça para que o processo siga seu fluxo. Quando esse evento temporal precede uma tarefa fica especificado que essa tarefa só é disparada no instante em que a condição temporal definida no evento é verdadeira. Se a tarefa que depende do evento temporal para ser executada for transformada em um caso de uso pode-se assumir que o acionador desse caso de uso será o evento temporal, da mesma forma que ocorre com eventos temporais de início. O princípio é o mesmo utilizado nos eventos de início, ou seja, eventos temporais (tanto início como intermediários) que precedem tarefas se tornam acionadores dos casos de uso originados por essas tarefas, quando essas dão origem a casos de uso.

Outro tipo de evento intermediário é aquele que só ocorre a partir de uma condição qualquer especificada. Essa condição pode ser uma regra de negócio associada com alguma informação do processo. Esse evento intermediário condicional pode preceder uma tarefa. Nessa situação a tarefa só ocorre quando a condição associada ao evento é verdadeira. Se a tarefa que tem seu início condicionado por um evento desse tipo se tornar um caso de uso do sistema, então o evento

condicional é a pré-condição desse caso de uso. O raciocínio utilizado é o mesmo para eventos de início do tipo condicional. Resumindo, eventos condicionais de início ou intermediários (contidos no fluxo) que precedem uma tarefa se tornam pré-condições do caso de uso originado por essa tarefa.

A Figura 44 apresenta um exemplo de transformação de um processo contendo um evento intermediário condicional. O processo de negócio tem uma tarefa que é a solicitação da inscrição em um curso. Uma vez feita a inscrição, o processo aguarda através de um evento intermediário condicional a confirmação do curso. Após a confirmação do curso o aluno pode confirmar sua participação. Os eventos e exceções desse processo não estão representados, pois a finalidade do exemplo é mostrar apenas a ocorrência do evento intermediário condicional.

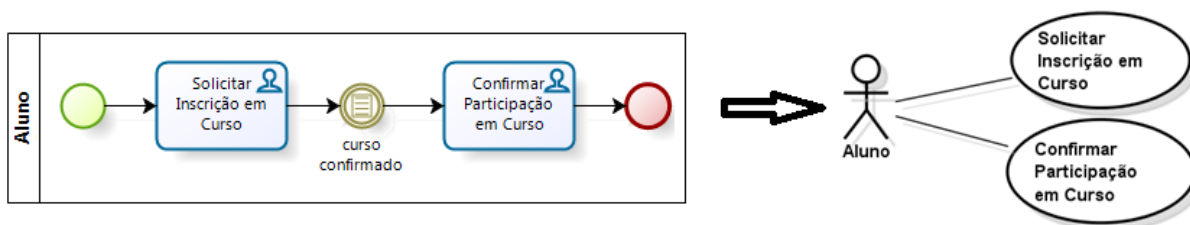


Figura 44 - Transformação de processo com evento intermediário condicional

Em relação à descrição dos casos de uso obtidos na transformação a pré-condição do caso de uso Confirmar Participação em Curso tem como valor a condição definida no evento intermediário, ou seja, curso confirmado. A Figura 45 abaixo apresenta a descrição obtida.

Nome do Caso de Uso:	Confirmar Participação em Curso
Descrição:	
Lista de Atores:	Aluno
Pré-condições:	Curso confirmado
Acionador:	
Fluxo básico de eventos:	
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 45 - Descrição do caso de uso Confirmar Participação em Curso

Situação 2 – Eventos intermediários anexados a uma tarefa

Essa segunda situação aborda eventos intermediários, tanto temporais como condicionais, que estão anexados a uma tarefa do processo. Quando um evento intermediário anexado a uma tarefa é disparado ele interrompe a execução da tarefa e o fluxo segue para o elemento ligado a esse evento. A ocorrência desse evento desvia o fluxo de execução para outro elemento, que pode ser outra tarefa.

Se uma tarefa que possui um evento anexado se torna um caso de uso, então esse evento pode dar origem a um fluxo alternativo no caso de uso gerado ou mesmo a um relacionamento de extensão. Essa última situação ocorre quando o evento está ligado à outra tarefa que também é transformada em caso de uso. Esse relacionamento de extensão está ligado a uma condição que é a condição de ocorrência do evento. Ou seja, o ponto de extensão do caso de uso base possui uma condição para ocorrer que é justamente a condição definida no evento. Essa condição pode ser tanto temporal como de outro tipo.

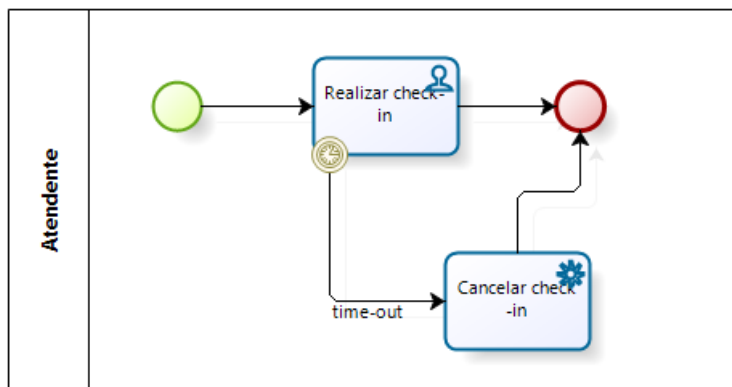


Figura 46 - Processo de negócio utilizando evento intermediário anexado a tarefa

O processo apresentado na Figura 46 é utilizado como exemplo para a transformação de evento anexado a tarefa. Nesse processo existe uma tarefa de usuário denominada Realizar Check-in. Essa tarefa tem um tempo limite para ser executada. Uma vez atingido esse tempo é disparada outra tarefa, dessa vez de serviço, que cancela o check-in iniciado pela tarefa anterior. Fazendo a transformação desse processo para casos de uso é possível obter dois casos de uso, um relacionado à tarefa de Realizar Check-in e outro com a tarefa de Cancelar Check-in. Esse segundo caso de uso pode ser definido como uma extensão do primeiro. Esse relacionamento de extensão entre os dois casos de uso depende de uma condição, que é a condição temporal definida no evento da BPMN. A Figura 47 mostra o resultado dessa transformação.



Figura 47 - Transformação de evento temporal anexado a tarefa para casos de uso

No exemplo anterior a tarefa de serviço foi transformada em um caso de uso separado. Essa tarefa pode também ser transformada em passos do caso de uso, conforme explicado na seção sobre

transformação de tarefa de serviço. Nesse caso o resultado da transformação seria de apenas um caso de uso, mas contendo na sua descrição um fluxo alternativo descrevendo o comportamento relacionado com o cancelamento do *check-in*, conforme exibido na Figura 48.

Nome do Caso de Uso:	Realizar Check-in
Descrição:	
Lista de Atores:	Atendente
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	
Fluxos alternativos:	FA1. Ocorrência de Time-Out Esse fluxo inicia no momento em que o tempo limite para execução do caso de uso é encerrado. 1. Sistema cancela check-in ...
Pontos de Extensão:	
Pós-condições:	

Figura 48 - Caso de uso Realizar Check-in

No nível dos metamodelos a transformação de eventos intermediário utiliza as classes *Intermediate Event* da BPMN e as classes *Use Case*, *Extended*, *Constraint* e *Step* da UML.

5.3.6.3 EVENTO FINAL

O evento de final como o próprio nome define especifica o local onde o processo é encerrado. Após um evento de final nenhum outro fluxo é seguido e nenhuma outra tarefa é realizada. Um processo de negócio pode ter mais de um evento de final.

Ao contrário dos eventos de início e intermediário, um evento de final não pode ser do tipo condicional nem do tipo tempo. Isso porque o processo se encerra nesse evento e esses dois tipos estão associados a condições que especificam quando o evento é disparado e conseqüentemente quando o fluxo do processo segue sua execução. Como o evento de final define o fim do processo não faz sentido associar alguma condição a esse evento porque não há fluxo de processo depois dele.

O evento de final mais comum de ser utilizado é o evento do tipo mensagem. Esse evento envia uma mensagem para outro participante (*pool*) do processo. Como nesse trabalho as transformações definidas estão limitadas a elementos dentro de um mesmo participante então não há definição de transformação de evento final do tipo mensagem para elementos do modelo de sistema utilizando casos de uso. Pela mesma razão os eventos de início e intermediário do tipo mensagem não possuem transformação definida. A análise de como poderia ser feita a essa transformação, assim como de outros tipos de eventos da BPMN podem ser objetos de estudos futuros.

5.3.7 ARTEFATOS

Os artefatos são elementos adicionais que podem ser incorporados ao diagrama, mas que não alteram o fluxo do processo. São utilizados para auxiliar no entendimento do processo modelado adicionando informações para torná-lo mais claro.

Nesse trabalho são analisadas as possíveis transformações de dois tipos de artefatos: objeto de dados e grupo.

5.3.7.1 OBJETO DE DADOS

Os objetos de dados representam uma informação que pode ser usada como entrada ou saída de uma tarefa. Esses objetos de dados podem estar associados a um estado. A tarefa pode utilizar um artefato de dados já existente ou então gerar um novo ou ainda alterar o seu estado.

A transformação de um objeto de dados para elemento de casos de uso pode ocorrer de três formas:

1. Artefato de dados utilizado como entrada para uma tarefa se torna uma pré-condição do caso de uso gerado por essa tarefa;
2. A ação de geração de um artefato de dados utilizado como saída de uma tarefa ou mesmo a mudança do estado de um já existente se torna um passo do fluxo básico ou alternativo do caso de uso gerado pela tarefa associada ao artefato.
3. Quando a ação de geração do artefato ou a mudança de estado do mesmo é algo que sempre ocorre independentemente do que aconteça no caso de uso gerado pela tarefa associada ao artefato, então a ação de geração desse artefato se torna uma pós-condição do caso de uso.

A diferença entra a situação 2 e 3 descritas acima está na geração (ou alteração de estado) incondicional do artefato. Isso porque uma pós-condição de um caso de uso é algo que ocorre em qualquer cenário do caso de uso. Então para a geração do artefato ser uma pós-condição é necessário que ela sempre ocorra em qualquer situação. Caso contrário, a geração desse artefato ou alteração de estado do mesmo deve ser mapeada para um passo do fluxo básico ou alternativo do caso de uso gerado.

No nível dos metamodelos a transformação utiliza a classe *DataObject* da BPMN e as classes *Constraint* (para pré e pós condições) e *Step* (para os passos) da UML.

O exemplo a seguir mostra um exemplo de transformação de artefato de dados para elementos de casos de uso. O exemplo utiliza um processo de criação e envio de ordem de compras, conforme pode ser observado na Figura 49.

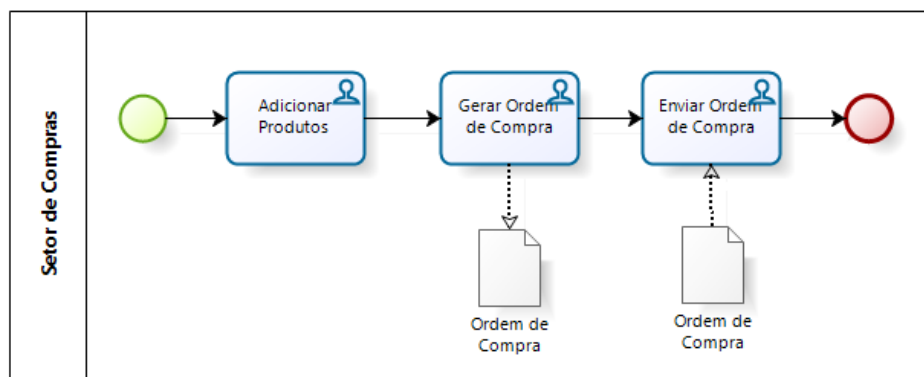


Figura 49 - Processo de negócio de Ordem de Compra

O processo possui três tarefas, uma relacionada com a inclusão de produtos, outra com a geração da ordem de compra propriamente dita e outra com o envio da ordem de compra. Cada tarefa desse processo é uma tarefa de usuário, logo pode ser transformada em caso de uso do sistema. O ator dos casos de uso gerados é o Setor de Compras, devidamente identificado pela raia que contém as tarefas. A tarefa Gerar Ordem de Compra está associada ao artefato objeto de dados chamado Ordem de Compra. Esse objeto aparece como saída dessa tarefa. Fazendo a transformação dessa tarefa obtém-se um caso de uso chamado Gerar Ordem de Compra que tem como pós-condição a criação do objeto Ordem de Compra. Essa transformação só é válida se esse caso de uso gerado sempre der origem a ordem de compra. Caso contrário o mais adequado é que a criação do objeto de dados apareça na descrição do fluxo básico ou alternativo.

A tarefa Enviar Ordem de Compra tem como objeto de entrada uma Ordem de Compra. Essa tarefa é transformada em um caso de uso com o mesmo nome tendo como pré-condição uma Ordem de Compra. Sendo assim a transformação resulta em três casos de uso, um deles com uma pós-condição e outro com uma pré-condição, conforme pode ser observado na Figura 50 que mostra o diagrama de casos de uso e na Figura 51 e Figura 52 que mostram as descrições de dois dos casos de uso.

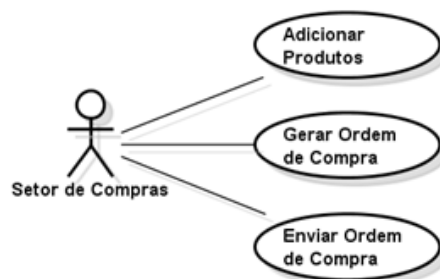


Figura 50 - Diagrama de Casos de Uso obtido na transformação

A figura abaixo apresenta a descrição do caso de uso Gerar Ordem de Compra indicando a pós-condição relacionada à criação do artefato de dados pelo sistema. Posteriormente, quando esse caso de uso tiver seu fluxo de eventos definido e possuir algum cenário em que a geração da ordem

de compra não ocorra, então a mesma deve deixar de ser uma pós-condição e se tornar um passo executado pelo sistema no fluxo de eventos.

Nome do Caso de Uso:	Gerar Ordem de Compra
Descrição:	
Lista de Atores:	Setor de Compras
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	Ordem de Compra gerada pelo sistema.

Figura 51 - Descrição do caso de uso Gerar Ordem de Compra utilizando uma pós-condição

A Figura 52 mostra a descrição do caso de uso Enviar Ordem de Compra. Para que esse caso de uso seja iniciado pelo sistema é necessário que exista uma ordem de compra. Por isso a existência de uma ordem de compra é uma pré-condição a esse caso de uso.

Nome do Caso de Uso:	Enviar Ordem de Compra
Descrição:	
Lista de Atores:	Setor de Compras
Pré-condições:	Ordem de Compra existente
Acionador:	
Fluxo básico de eventos:	
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 52 - Descrição do caso de uso Enviar Ordem de Compra contendo uma pré-condição

O exemplo anterior é interessante para se mostrar a diferença que pode ocorrer quando se visualiza o processo de negócio da BPMN e o conseqüente modelo de casos de uso originado.

O processo de geração de ordem de compra contido no diagrama BPMN expressa uma finalidade por si só, que é a geração dessa ordem de compra. Para isso são executadas três tarefas, sendo que uma depende da outra para ser realizada.

Ao transformar essas tarefas em casos de uso do sistema a informação relativa ao seqüenciamento dessas tarefas é perdida. O motivo é que o objetivo do diagrama de casos de uso não é expressar a seqüência de execução dos mesmos (apesar de que quando se utiliza relacionamento de extensão e inclusão uma seqüência entre os casos de uso relacionados pode ficar subentendida). Entretanto, casos de uso que não apresentam relação entre si não possuem informação sobre a seqüência de execução deles no sistema. No detalhamento dos casos de uso essa seqüência também não é descrita. As pré-condições e pós-condições não devem ser utilizadas para criar seqüência de execução dos casos de uso [KRU00]. Cada caso de uso é independente do outro, possuindo seu próprio objetivo que deve ser a geração de um valor observável para o ator.

O que pode ocorrer é que um caso de uso tenha como resultado algum objeto do sistema que seja necessário para a execução de outro caso de uso. Nessa situação estaria configurado que um determinado caso de uso só poderá ocorrer quando alguma instância (ocorrência completa) de outro

caso de uso tiver ocorrido, pelo menos uma vez. Ou seja, um caso de uso tem como pré-condição a existência de algo que só a ocorrência de outro caso de uso poderia ter originado, conforme ocorreu no exemplo da geração de ordem de compra. Entretanto, basta que apenas tenha ocorrido uma execução do caso de uso anterior para que o outro seja executado quando necessário. No exemplo anterior é necessário que apenas tenha sido gerada uma ordem de compra qualquer para que o caso de uso de Enviar Ordem de Compra tenha sua pré-condição atendida. A pré-condição do Enviar Ordem de Compra não é específica de uma determinada ordem de compra gerada por outro caso de uso, mas sim de qualquer ordem de compra. Ao se detalhar o caso de uso Enviar Ordem de Compra um dos passos do mesmo seria o de selecionar a ordem de compra a ser enviada. Nesse momento o usuário selecionaria a ordem de compra desejada e o sistema enviaria a mesma.

A explicação acima mostra que, ao transformar cada tarefa de um processo de negócio para casos de uso independentes a seqüência de execução dessas tarefas é perdida, porque cada tarefa é vista como tendo um objetivo próprio.

Entretanto, podem ocorrer situações em que uma simples tarefa do processo de negócio não configura um caso de uso próprio. Podem acontecer situações em que seja mais interessante a transformação de várias tarefas para um único caso de uso ao invés de cada tarefa para um caso de uso. Dessa forma cada tarefa pode se tornar um passo do caso de uso. Para isso é necessário a identificação de quais tarefas (ou outros elementos do BPMN) dão origem a apenas um caso de uso e qual é o objetivo específico desse conjunto de elementos. Uma forma de permitir essa transformação de vários elementos de um diagrama BPMN para apenas um caso de uso é utilizando o artefato Grupo, disponível na BPMN.

5.3.7.2 GRUPO

A transformação de tarefas de um diagrama BPMN para casos de uso independentes acaba não permitindo representar a seqüência de execução dessas tarefas, conforme foi explicado na seção anterior. As pré-condições e pós-condições não devem ser utilizadas para essa finalidade, segundo a própria documentação de casos de uso do RUP.

Essa limitação se torna um problema quando cada tarefa do diagrama em BPMN não expressa por si só um caso de uso independente. Podem ocorrer situações em que as tarefas representam passos de um caso de uso e não o caso de uso inteiro. No trabalho de Dijkman [DIJ02] é proposto o conceito de Passo (*step*) em um diagrama BPMN. Um passo é um agrupamento de tarefas relacionadas seqüencialmente e que são executadas sem interrupção. Na proposta desse autor cada passo é transformado em um caso de uso e as tarefas do passo são as interações que ocorrem entre o ator e o sistema. No exemplo da geração de ordem de compra as três tarefas representam um passo, logo elas seriam apenas um caso de uso, utilizando esse conceito.

A proposta do conceito de passo é interessante e resolve em parte a questão de que várias tarefas representam apenas um caso de uso. Entretanto, podem ocorrer situações em que várias tarefas podem ou não representar apenas um caso de uso, além de que outros elementos associados a essas tarefas (como *gateways*) também podem fazer parte de apenas um caso de uso. Ou seja, é importante ser possível agrupar tarefas e outros elementos indicando que esse agrupamento tem uma finalidade específica (objetivo) e que o mesmo é transformado em um único caso de uso. As tarefas contidas dentro dele bem como outros elementos se tornam interações do caso de uso. O princípio é o mesmo utilizado para o conceito de passo, mas deixando o responsável pela modelagem com maior liberdade para definir o que faz parte ou não do caso de uso.

Para possibilitar isso é utilizado o elemento Grupo da BPMN, que permite agrupar elementos do diagrama. A BPMN não atribui nenhuma semântica específica para esse artefato sendo ele utilizado apenas no agrupamento visual dos elementos do diagrama. A proposta utilizada no presente trabalho atribui uma semântica própria para o Grupo. A utilização dele no diagrama indica que os elementos agrupados são transformados em um único caso de uso e as interações do caso de uso são obtidas a partir dos elementos dentro do grupo. A transformação leva em consideração todos os elementos abordados nas outras transformações citadas anteriormente, exceto os eventos. O elemento fluxo de seqüência também é utilizado nessa transformação.

A transformação gera um caso de uso cujo nome é o nome do elemento Grupo do diagrama. As tarefas de usuário dentro do grupo tornam-se passos executados pelo ator. O ator é obtido a partir da raia que contém esse grupo. Caso o grupo tenha elementos em mais de uma raia, cada raia diferente se torna um ator do caso de uso. As tarefas de serviço tornam-se passos do sistema. A seqüência dos passos é definida pela seqüência das tarefas no diagrama, ou seja, pelos fluxos de seqüência existentes no BPMN. A ocorrência de *gateways* (exclusivo, inclusivo, paralelo) dentro do grupo indica a existência de fluxo alternativos do caso de uso. O caminho padrão dá origem a passos do fluxo básico. Os demais caminhos do *gateway* que estejam dentro do grupo dão origem a passos de fluxos alternativos. Os artefatos de dados utilizados como entrada de uma tarefa se tornam passos do sistema que representam uma leitura desse artefato. Os artefatos de dados utilizados como saída de uma tarefa dentro do grupo se tornam passos do sistema que representam a geração desse artefato.

Em relação aos metamodelos várias classes são utilizadas, tanto no metamodelo da BPMN como da UML. No metamodelo da BPMN são usadas a classe *Group* e todas as outras relativas a elementos que estejam dentro do grupo. Na UML são usadas as classes *UseCase*, *Constraint*, *Step* e *Alternative* (para fluxos alternativos).

O exemplo a seguir (Figura 53) apresenta o processo de negócio da Ordem de Compra utilizado no exemplo do artefato de dados contendo todas as tarefas dentro de um grupo. O nome do

grupo é Submeter Ordem de Compra. A idéia é mostrar que o processo de geração e envio da ordem de compra é percebido como um objetivo único, que é Submeter Ordem de Compra. A transformação resulta em apenas um caso de uso, conforme a Figura 54.

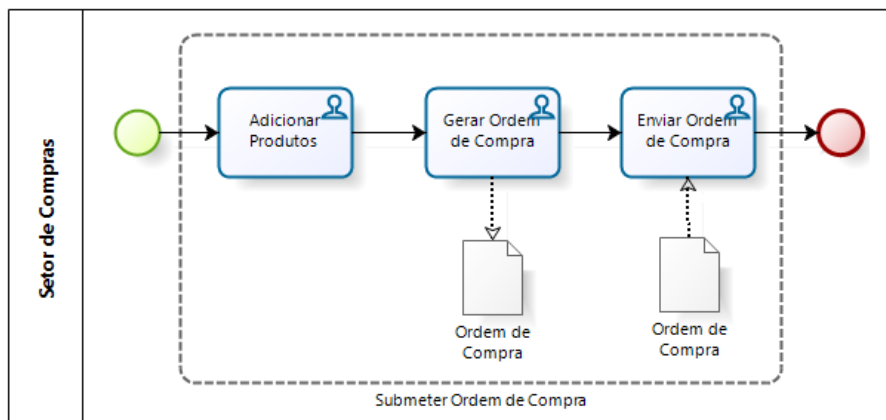


Figura 53 - Processo de negócio utilizando elemento grupo

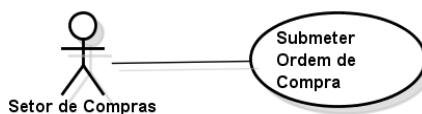


Figura 54 - Caso de uso originado pelo processo contendo o elemento grupo

A figura acima mostra o diagrama de caso de uso originado. Como o agrupamento usado contém todas as tarefas dentro do processo foi originado apenas um caso de uso. As tarefas que ocorrem dentro do grupo se tornam passos do caso de uso gerado. A Figura 55 apresenta um exemplo descrição do caso de uso gerado pela transformação.

A vantagem de se agrupar as tarefas é que se torna possível transformar várias tarefas em um único caso de uso. A descrição dos passos pode variar dependendo dos elementos contidos no grupo. No caso da entrada ou saída de artefato de dados o passo é realizado pelo sistema como resultado de algum passo anterior realizado pelo ator. A descrição gerada serve como base para que seja refinada pelo analista de sistemas. A descrição obtida por si só não é completa.

Nome do Caso de Uso:	Submeter Ordem de Compra
Descrição:	
Lista de Atores:	Setor de Compras
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	<ol style="list-style-type: none"> 1. Setor de Compras adiciona produtos 2. Setor de Compras gera a ordem de compra 3. Sistema grava a ordem de compra gerada 4. Sistema carrega ordem de compra 5. Setor de Compras envia ordem de compra
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 55 - Descrição do caso de uso Submeter Ordem de Compra

O exemplo mostrado é simples e utiliza apenas tarefas de usuário seqüenciais, não mostrando nenhum *gateway*. A seguir é utilizado um exemplo mais complexo para demonstrar a transformação de elementos dentro do grupo.

Exemplo de transformação de processos contendo *gateway* dentro de grupo

O exemplo a seguir mostra a transformação para casos de uso de um processo de negócio contendo um elemento do tipo *gateway* dentro de um grupo. O grupo é transformado em um caso de uso único e os elementos dentro do *gateway* se tornam as interações desse caso de uso.

A Figura 56 apresenta o processo de negócio. Esse processo apresenta, de forma resumida, a criação de uma nova conta em uma instituição financeira. Primeiro a documentação é reunida, em seguida é iniciada a abertura da conta informando os dados cadastrais. Somente se o cliente for especial então é informado um limite de crédito. No final os dados são enviados para a matriz da empresa. Esse envio é feito por uma rotina do sistema que gera um arquivo e envia o mesmo para o sistema principal.

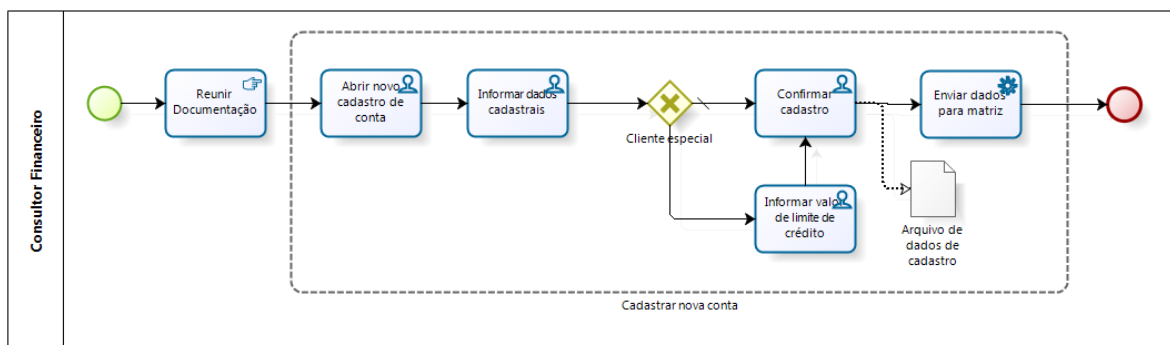


Figura 56 - Processo de negócio com grupo e *gateway*

Transformando esse processo de negócio para caso de uso se obtém apenas um caso de uso. Isso porque todas as tarefas de usuário ou sistema estão dentro de um grupo. A tarefa fora do grupo é uma tarefa manual, que não é passível de transformação, conforme já explicado. Sendo assim o diagrama de casos de uso obtido possui apenas um caso de uso. A Figura 57 mostra esse resultado.



Figura 57 - Diagrama de caso de uso originado da transformação do processo com grupo e *gateway*

Os elementos contidos no grupo dão origem a elementos da descrição do caso de uso. A seguir é apresentado um passo a passo explicando a transformação completa e a conseqüente geração da descrição do caso de uso Cadastrar Nova Conta.

- 1) A raia do processo de negócio dá origem a um ator de sistema com o mesmo nome, nesse caso Consultor Financeiro.

- 2) Cada grupo contido no diagrama é transformado em um caso de uso do sistema. O nome desse caso de uso é o nome do grupo (Cadastrar Nova Conta). O ator é o que foi originado pela raia, ou seja, Consultor Financeiro.
- 3) Dentro do grupo as duas primeiras tarefas dão origem aos dois primeiros passos do ator no fluxo básico do caso de uso.
- 4) As tarefas do fluxo padrão do *gateway* também dão origem a passos do ator no fluxo básico.
- 5) O fluxo alternativo do *gateway* dá origem a um fluxo alternativo do caso de uso.
- 6) A geração do artefato de dados é transformada em um passo do sistema.
- 7) A seqüência dos passos é obtida de acordo com a seqüência das tarefas.

A descrição do caso de uso obtida a partir da transformação pode ser observada na Figura 58.

Nome do Caso de Uso:	Cadastrar nova conta
Descrição:	
Lista de Atores:	Consultor Financeiro
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	<ol style="list-style-type: none"> 1. Consultor Financeiro abre novo cadastro de conta. 2. Consultor Financeiro informa dados cadastrais. 3. <u>Sistema verifica tipo de cliente</u> 4. Consultor Financeiro confirma cadastro. 5. Sistema gera arquivo de dados de cadastro. 6. Sistema envia arquivo de dados para matriz.
Fluxos alternativos:	<p>FA1 – Cliente especial</p> <p>No passo 3 o sistema identifica que o cliente é especial.</p> <ol style="list-style-type: none"> 1. Consultor Financeiro informa valor de limite de crédito. <p>Caso de uso retorna ao passo 4.</p>
Pontos de Extensão:	
Pós-condições:	

Figura 58 - Descrição do caso de uso Cadastrar nova conta

Conforme pode ser observado, a descrição gerada é mais completa se comparada com as outras apresentadas anteriormente. Isso ocorre porque dentro do grupo estão diversos elementos e esses elementos fornecem as informações necessárias à descrição.

De qualquer forma a descrição obtida não é definitiva. Assim como nas outras situações apresentadas anteriormente ela deve ser refinada pelo analista de sistemas de forma a se tornar completa e expressar com maior precisão as interações do caso de uso. Nesse exemplo o passo 3 teve que ser acrescentado para que o restante da descrição ficasse consistente. Esse passo tem relação com o *gateway*. O *gateway* expressa uma decisão que deve ser tomada. Entretanto não fica claro quem toma a decisão, se é o ator ou se é o sistema. Nesse exemplo o mais adequado é colocar o passo da decisão relacionado ao sistema, mas pode ser que em outra situação o mais adequado seja uma decisão pelo ator. O método apresentado não especifica quando a decisão do *gateway* é passo do ator ou do sistema. Ele apenas indica que isso deve ser observado na descrição, ficando a cargo do analista de sistemas essa definição.

5.4 CONSIDERAÇÕES FINAIS SOBRE O MÉTODO PROPOSTO

Esse capítulo apresentou um método que tem como objetivo transformar um modelo de processo de negócio descrito em BPMN para um modelo de casos de uso de sistema. Wazlawick [WAZ09] define um método como sendo um caminho para atingir um objetivo. No método apresentado o caminho consiste em três passos: verificar se o modelo de processo de negócio pode ser transformado; analisar as possíveis transformações; refinar as descrições obtidas.

A verificação do modelo analisa se o mesmo contém os elementos necessários para que a transformação seja realizada. A análise das transformações é a principal contribuição do método proposto. Para cada elemento da BPMN considerado no trabalho é feita uma análise de como esse elemento pode ser transformado em um caso de uso (ou parte de um caso de uso). Já o refinamento das descrições é uma tarefa de complementação do resultado obtido pelas transformações. Cada caso de uso gerado não possui uma descrição completa, mas sim uma descrição inicial contendo os elementos que puderam ser identificados no BPMN. O restante do detalhamento do caso de uso deve ser feito pelo analista de sistemas.

Conceitualmente o método realiza uma transformação de um modelo em um nível de abstração de negócio para um nível de abstração de sistema. A transformação entre modelos deve considerar os elementos do metamodelo de cada um dos modelos utilizados [OMG03]. Por isso para cada transformação explicada é colocado qual o elemento do metamodelo relacionado.

Com base na classificação de transformação de modelos [MEN06] apresentada no Capítulo 3 o método realiza uma transformação vertical, exógena, não automática e semântica. A transformação é vertical, pois utiliza modelos em diferentes níveis de abstração. É exógena porque são modelos com diferentes metamodelos. Não é automática uma vez que necessita de intervenção para definir as alternativas utilizadas em cada situação. É semântica porque busca relação de conceitos entre os elementos dos modelos utilizados.

O método apresenta alguns diferenciais em relação a outros trabalhos que também abordam esse tema, conforme apresentado abaixo.

5.4.1 ANÁLISE DE DIFERENTES POSSIBILIDADES DE TRANSFORMAÇÃO

Ao contrário de outros trabalhos [ROD10, LIE04, DIJ02] que na maioria das situações apresentam apenas uma possibilidade de transformação de elementos, nesse trabalho são apresentadas várias alternativas para cada elemento. Cada alternativa é explicada ao longo da descrição do método e ao aplicá-lo essas explicações devem ser consideradas.

5.4.2 ELEMENTOS DA DESCRIÇÃO DOS CASOS DE USO

As transformações apresentadas consideram diversos elementos da descrição dos casos de uso, gerando não apenas o diagrama de casos de uso, mas descrições com pré-condições, pós-condições, acionador, fluxos, entre outros elementos. Apesar da descrição não ser completa, conforme já explicado, ela serve como base para que o analista de sistemas complete a mesma com informações que não estão no processo de negócio apresentado no diagrama BPMN.

5.4.3 TRANSFORMAÇÕES DOS EVENTOS

Os eventos da BPMN são considerados nas definições das possíveis transformações. Em outros trabalhos relacionados o tratamento de eventos iniciais, intermediários e finais não é considerado. Apenas o trabalho de Javier [BER09] faz menção a exceções que podem ocorrer ligadas a tarefas, que seriam os eventos intermediários anexados a uma tarefa e que esses originam relacionamentos de extensão.

5.4.4 TRANSFORMAÇÃO DOS ARTEFATOS

Também é abordada a transformação dos artefatos Objeto de Dados e Grupo. Esse último quando utilizado no diagrama BPMN muda a maneira que os demais elementos são transformados. A utilização do Grupo permite uma maior liberdade ao modelador para definir partes do diagrama que originam apenas um caso de uso.

5.4.5 REFERÊNCIA AOS METAMODELOS

As transformações apresentadas são definidas analisando, mesmo que superficialmente, os elementos dos metamodelos da BPMN e da UML. Esses estão comentados nos Capítulos 2 e 3. Cada possibilidade de transformação analisada indica quais as classes dos metamodelos que são utilizadas. Essa referência aos metamodelos pode servir de base para uma futura implementação de uma ferramenta que automatize essas transformações.

6 APLICAÇÃO DO MÉTODO PROPOSTO

O objetivo desse capítulo é apresentar aplicações práticas do método proposto em modelos de processo de negócio relacionados a situações práticas. Para isso são utilizados dois exemplos. O primeiro relacionado com um processo de negócio de um hospital da cidade de Porto Alegre. O segundo exemplo é um processo de negócio para solicitação de reembolso de despesas de viagem em uma organização.

Para cada processo apresentado é feito um comentário geral sobre o mesmo e em seguida é aplicado o método de transformação. O modelo de casos de uso resultante da transformação contém o diagrama com os casos de uso e seus detalhamentos. No final é feito um comentário geral sobre o resultado obtido.

O capítulo está organizado da seguinte forma: a Seção 6.1 apresenta os passos genéricos para aplicação do método; a Seção 6.2 utiliza o método no processo relacionado ao hospital e a Seção 6.3 aplica o método no processo relacionado à solicitação de reembolso.

6.1 PASSOS GENÉRICOS PARA APLICAÇÃO DO MÉTODO

Essa seção apresenta uma seqüência de ações que são genéricas para a aplicação do método proposto. Nas próximas seções esses passos são aplicados para cada exemplo de processo utilizado.

A Tabela 4 apresenta os passos genéricos para a aplicação do método proposto. As análises detalhadas de como cada elemento pode ser transformado estão descritas no Capítulo 5.

Passo	Descrição
1	Verificar se o modelo de processo de negócio possui informações relativas ao sistema.
2	Se no passo 1 o modelo está com informações de sistema então (caso contrário o processo é interrompido): <ol style="list-style-type: none"> 2.1. Transformar as Raias em Atores de Sistema; 2.2. Transformar os Grupos em Casos de Uso. 2.3. Transformar as Tarefas de Usuário (que não estejam dentro de um Grupo) em Casos de Uso; 2.4. Transformar as Tarefas de Serviço (que não estejam dentro de um Grupo) em Casos de Uso ou em passos de um caso de uso obtido pelas Tarefas de Usuário. 2.5. Transformar os <i>Gateways</i> em relacionamentos de Casos de Uso analisando as diferentes alternativas descritas no método; 2.6. Transformar os Eventos em elementos dos Casos de Uso obtidos; 2.7. Transformar os Artefatos Objeto de Dados em elementos dos Casos de Uso obtidos.
3	Refinar as descrições de casos de uso obtidas.

Tabela 4 - Passos genéricos para aplicação do método

O refinamento das descrições pode ser realizada a cada transformação. O passo 3 é apenas o refinamento final das descrições obtidas.

6.2 APLICAÇÃO 1 – PROCESSO DE SOLICITAÇÃO DE MATERIAL INTERNO EM UM HOSPITAL

O exemplo utilizado nessa aplicação é de um processo de negócio modelado para atender a um hospital e está relacionado com o pedido de reposição de materiais. O processo foi modelado por uma analista de negócio sem levar em consideração a criação de um sistema para informatizar o mesmo. Posteriormente, esse foi complementado prevendo o uso de um sistema a ser desenvolvido para realizar determinadas tarefas.

6.2.1 REALIZAÇÃO DA TRANSFORMAÇÃO

O processo exibido na Figura 59 contém as tarefas realizadas pelo Enfermeiro Chefe e pelo Gestor para solicitar reposição de estoque de materiais em uma determinada unidade do hospital. Esses materiais são de diversos tipos e possuem um estoque mínimo que deve ser observado em cada unidade.

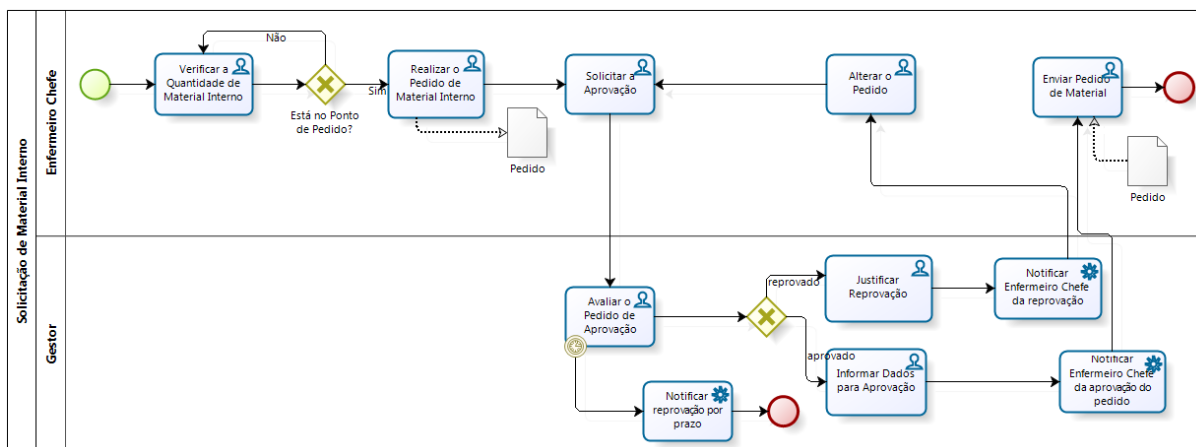


Figura 59 - Processo de Solicitação de Material Interno

Em resumo o processo inicia pelo Enfermeiro Chefe verificando o nível de estoque de um determinado material no sistema. Dependendo da situação é feito um pedido. O pedido deve ser aprovado pelo Gestor para que seja encaminhado ao setor de controle de materiais.

A primeira atividade a ser realizada para transformar o modelo de processo de negócio em casos de uso é verificar se o modelo está em um nível em que possa ser realizada a transformação. Conforme mencionado anteriormente esse modelo foi construído prevendo a utilização de um sistema a ser desenvolvido, tanto que os tipos de tarefas estão diferenciados no diagrama. Esse é um dos fatores necessários para que ele possa ser analisado para identificar casos de uso. O segundo são os elementos da BPMN utilizados. Todos os elementos utilizados podem ser transformados em casos de uso de acordo com o método proposto. Sendo assim, o modelo verificado pode ser transformado, pois está de acordo com as necessidades indicadas no método de transformação.

A próxima atividade é a transformação de cada elemento de acordo com as possibilidades apresentadas no capítulo anterior. Abaixo está um passo a passo da realização desse procedimento:

- 1) As raiais Enfermeiro Chefe e Gestor dão origem a atores do sistema.
- 2) Todas as tarefas da raia do Enfermeiro Chefe dão origem a casos de uso cujo ator é o Enfermeiro Chefe.
- 3) O *gateway* que relaciona as tarefas Verificar Quantidade de Material Interno e Realizar Pedido de Material Interno se torna um relacionamento de extensão entre os casos de uso gerados por essas duas tarefas, sendo que o caso de uso Realizar Pedido de Material Interno estende o caso de uso Verificar Quantidade de Material Interno. O ponto de extensão está no passo de verificação se o pedido é necessário, sendo essa também a condição para que a extensão ocorra.
- 4) A tarefa Realizar Pedido de Material Interno possui um artefato de dados associado a ela. A geração desse artefato se torna uma pós-condição do caso de uso gerado por ela.
- 5) A tarefa Enviar Pedido de Material possui um artefato de dados associado a ela como entrada. A existência desse artefato se torna uma pré-condição do caso de uso originado por essa tarefa.

A Figura 60 mostra o diagrama de casos de uso originado pelas transformações realizadas nas tarefas da raia do Enfermeiro Chefe.

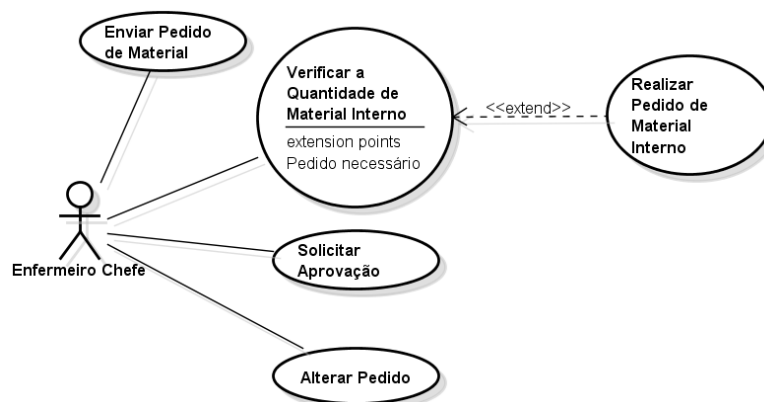


Figura 60 - Diagrama de Casos de Uso obtido a partir das tarefas da raia do Enfermeiro Chefe

- 6) As tarefas Avaliar Pedido de Aprovação, Informar Dados para Aprovação, Justificar Reprovação e Notificar Reprovação por Prazo se tornam casos de uso do sistema associados ao ator Gestor.
- 7) O *gateway* exclusivo que sucede a atividade Avaliar Pedido de Aprovação dá origem a relacionamentos de especialização entre os casos de uso Avaliar Pedido de Aprovação, Informar Dados para Aprovação e Justificar Reprovação. Os casos de uso dentro do *gateway* se tornam filhos do caso de uso que precede o *gateway*.

- 8) As tarefas Notificar Enfermeiro Chefe da reprovação e Notificar Enfermeiro Chefe da aprovação tornam-se passos dos casos de uso Justificar Reprovação e Informar Dados para Aprovação, respectivamente.
- 9) O evento intermediário temporal anexado a tarefa Avaliar Pedido de Aprovação é transformado em um relacionamento de extensão entre o caso de uso Notificar Reprovação por Prazo e o caso de uso Avaliar Pedido de Reprovação. O ponto de extensão ocorre na verificação do prazo para a aprovação.

A Figura 61 apresenta o diagrama de casos de uso obtido a partir das transformações realizadas na raia do Gestor.

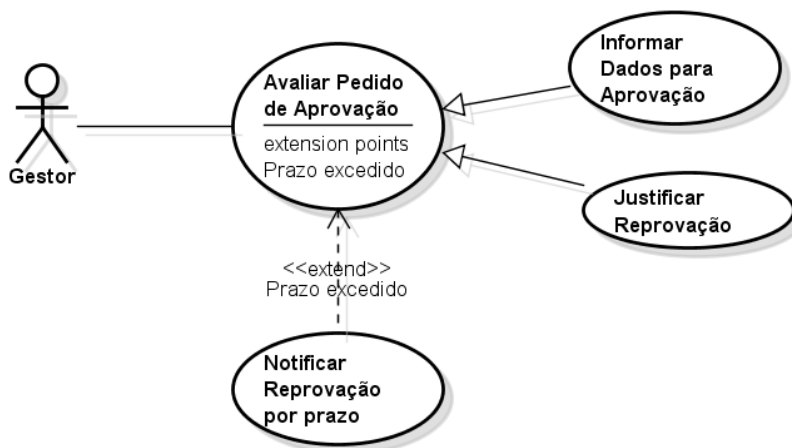


Figura 61 - Diagrama de Casos de Uso obtido a partir da raia do Gestor

Além dos diagramas também são geradas as versões iniciais das descrições dos casos de uso obtidos. A Figura 62, Figura 63 e Figura 64 apresentam as versões iniciais dos casos de uso obtidos relacionados ao ator Enfermeiro Chefe, exceto do Solicitar Aprovação e Alterar Pedido. Esses dois casos de uso possuem na descrição inicial apenas o nome do ator. As outras descrições possuem mais de um elemento que foi obtido através do uso do método de transformação.

Nome do Caso de Uso:	Verificar Quantidade de Material Interno
Descrição:	
Lista de Atores:	Enfermeiro Chefe
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	...
Fluxos alternativos:	Passo <i>n</i> - Sistema verifica quantidade em estoque
Pontos de Extensão:	Ponto Extensão Pedido Necessário No Passo <i>n</i> – Sistema verifica quantidade em estoque, se a quantidade em estoque está no nível que é necessário realizar um pedido então utiliza o caso de uso Realizar Pedido de Material Interno.
Pós-condições:	

Figura 62 - Descrição do caso de uso Verificar Quantidade de Material Interno

Nome do Caso de Uso:	Realizar Pedido de Material Interno
Descrição:	
Lista de Atores:	Enfermeiro Chefe
Pré-condições:	
Acionador:	Caso de uso Verificar Quantidade de Material Interno
Fluxo básico de eventos:	
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	Pedido gerado.

Figura 63 - Descrição do caso de uso Realizar Pedido de Material Interno

Nome do Caso de Uso:	Enviar Pedido de Material Interno
Descrição:	
Lista de Atores:	Enfermeiro Chefe
Pré-condições:	Existência do Pedido
Acionador:	
Fluxo básico de eventos:	
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 64 - Descrição do caso de uso Enviar Pedido de Material Interno

Observando os detalhamentos pode-se verificar que a descrição que obteve mais elementos a partir da transformação foi a do caso de uso Verificar Quantidade de Material de Estoque. Essa descrição apresenta um ponto de extensão e uma condição associada ao mesmo. A forma de descrever um ponto de extensão varia dependendo do formato utilizado para o caso de uso. Independentemente da forma de descrição o importante é expressar a existência do relacionamento de extensão e da condição para que o mesmo ocorra, podendo essa informação aparecer textualmente de diferentes maneiras.

A Figura 65, Figura 66, Figura 67, Figura 68 apresentam as descrições obtidas para os casos de uso relacionados com o ator Gestor.

Nome do Caso de Uso:	Avaliar Pedido de Aprovação
Descrição:	
Lista de Atores:	Gestor
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	...
	Passo <i>n</i> - Sistema verifica data do pedido.
	Passo <i>m</i> – Usuário seleciona situação do pedido.
Fluxos alternativos:	Fluxo Alternativo 1 – Pedido Aprovado No Passo <i>m</i> o ator aprovou o pedido. Utilizar o caso de uso Informar Dados para Aprovação.
	Fluxo Alternativo 2 – Pedido Reprovado No Passo <i>m</i> o ator reprovou o pedido. Utilizar o caso de uso Justificar Reprovação.
Pontos de Extensão:	Ponto de Extensão Prazo Excedido No Passo <i>n</i> – Sistema verifica data do pedido, se a quantidade de dias desde a data tiver excedido o prazo então é utilizado o caso de uso Notificar Reprovação por Prazo.
Pós-condições:	

Figura 65 - Descrição do caso de uso Avaliar Pedido de Aprovação

Nome do Caso de Uso:	Notificar Reprovação por Prazo
Descrição:	
Lista de Atores:	Gestor
Pré-condições:	
Acionador:	Caso de uso Avaliar Pedido de Aprovação
Fluxo básico de eventos:	
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 66 - Descrição do caso de uso Notificar Reprovação por Prazo

O caso de uso Notificar Reprovação por Prazo poderia também ser um cenário alternativo do caso de uso Avaliar Pedido de Aprovação. Nessa modelagem se optou por fazer um caso de uso separado, pois o mesmo pode ser reutilizado em outra parte do sistema. A finalidade desse caso de uso é simplesmente avisar da reprovação do pedido, sendo que o mesmo está sendo acionado pelo caso de uso de avaliação. Entretanto, esse caso de uso poderia ser disparado pelo próprio tempo do sistema, conforme mencionado na explicação sobre eventos do tipo tempo. A forma como esse caso de uso seria acionado (*job*, serviço ou outros) não é definido na modelagem por se tratar de restrições de implementação. Entretanto, a separação dele em outro caso de uso pode auxiliar a equipe de projeto do sistema a definir a melhor forma de implementação.

Nome do Caso de Uso:	Informar Dados para Aprovação
Descrição:	
Lista de Atores:	Gestor
Pré-condições:	
Acionador:	Caso de uso Avaliar Pedido de Aprovação
Fluxo básico de eventos:	...
Fluxos alternativos:	Passo <i>n</i> - Sistema envia notificação da aprovação do pedido.
Pontos de Extensão:	
Pós-condições:	

Figura 67 - Descrição do caso de uso de Informar Dados para Aprovação

Nome do Caso de Uso:	Justificar Reprovação
Descrição:	
Lista de Atores:	Gestor
Pré-condições:	
Acionador:	Caso de uso Avaliar Pedido de Aprovação
Fluxo básico de eventos:	...
Fluxos alternativos:	Passo <i>n</i> - Sistema envia notificação da reprovação do pedido.
Pontos de Extensão:	
Pós-condições:	

Figura 68 - Descrição do caso de uso de Justificar Reprovação

De acordo com o exposto no capítulo anterior essas descrições não são completas porque nem todos os elementos da descrição estão contidos nos diagramas de processo de negócio. É necessário um refinamento de cada descrição gerada. Esse refinamento é feito pelo analista de

sistema com base em outras informações relativas ao sistema que está sendo modelado. A forma de fazer o refinamento não é escopo desse trabalho. Entretanto, na Figura 69 é apresentado um exemplo de como seria uma versão final de um dos casos de uso gerados na transformação. O caso de uso usado como exemplo é o de Avaliar Pedido de Aprovação. As informações destacadas são as originadas pelo método. As outras são as inseridas como refinamento.

Nome do Caso de Uso:	Avaliar Pedido de Aprovação
Descrição:	Esse caso de uso descreve o comportamento do sistema para a realização do processo de Avaliar Pedido de Aprovação
Lista de Atores:	Gestor
Pré-condições:	Usuário autenticado no sistema.
Acionador:	Caso de uso inicia quando o ator seleciona a opção de Avaliar Pedido de Aprovação.
Fluxo básico de eventos:	<ol style="list-style-type: none"> 1. Ator seleciona Pedido 2. Sistema apresenta informações do Pedido. 3. Sistema verifica data do Pedido. 4. Ator seleciona Situação do Pedido.
Fluxos alternativos:	<p>Fluxo Alternativo 1 – Pedido Aprovado No Passo <i>m</i> o ator aprovou o pedido. Utilizar o caso de uso Informar Dados para Aprovação.</p> <p>Fluxo Alternativo 2 – Pedido Reprovado No Passo <i>m</i> o ator reprovou o pedido. Utilizar o caso de uso Justificar Reprovação.</p>
Pontos de Extensão:	<p>Ponto de Extensão Prazo Excedido No Passo 3 – Sistema verifica data do pedido, se a quantidade de dias desde a data tiver excedido o prazo então é utilizado o caso de uso Notificar Reprovação por Prazo.</p>
Pós-condições:	Nenhuma

Figura 69 - Descrição (após refinamento) do caso de uso Avaliar Pedido de Aprovação

Observando o caso de uso com a descrição final apresentada na Figura 69 pode-se verificar que a maior parte da sua descrição foi obtida a partir do diagrama de processo de negócio utilizando o método de transformação proposto. A descrição complementar indicou informações sobre a interação entre o ator e o sistema, bem como pré-condições, descrição resumida e acionador.

6.2.2 CONSIDERAÇÕES SOBRE O RESULTADO OBTIDO

A aplicação do método no modelo de processo de Solicitação de Material Interno resultou em um diagrama de casos de uso contendo nove casos de uso e dois atores. Além do diagrama de casos de uso também foi possível obter alguns itens do detalhamento de cada um dos casos de uso originados.

Todos os casos de uso identificados configuram casos de uso do sistema a ser desenvolvido. Nesse ponto o uso do método auxiliou o analista a identificar os mesmos. O resultado obtido poderia ser diferente caso fossem escolhidas outras possibilidades no processo de transformação. Por exemplo, a notificação da aprovação ou reprovação do pedido após o *gateway* poderia dar

origem a casos de uso separados e não simplesmente passos de outros casos de uso. Os resultados são variados e dependem das escolhas feitas ao longo da transformação.

Em relação às descrições dos casos de uso essas não foram obtidas totalmente. Alguns casos de uso tiveram uma descrição inicial mais completa que outros. Foi o que ocorreu com o Avaliar Pedido de Aprovação. Esse possui vários itens do seu detalhamento obtidos pelo método. Já os casos de uso de Alterar Pedido e Solicitar Aprovação não tiveram nenhum item do detalhamento além do próprio ator. Esse fato mostra que o uso do *gateway* e do evento anexado a tarefa contribui para gerar uma descrição mais completa. Quanto mais elementos no diagrama BPMN ligados as tarefas maiores são as possibilidades de encontrar elementos de casos de uso através das transformações propostas.

A transformação do evento anexado a tarefa de Avaliar Pedido de Aprovação deu origem a um relacionamento de extensão com outro caso de uso, o de Notificar reprovação por prazo. Da forma que foi feita a transformação esse caso de uso da notificação só é disparado através do Avaliar Pedido de Transformação. Entretanto, pode ser necessário que essa notificação seja disparada automaticamente pelo sistema e não apenas quando fosse feita a avaliação do pedido. Uma das formas de resolver isso seria colocar que o acionador do caso de uso poderia ser o próprio tempo, ou então modelar outro processo contendo um evento de tempo que disparasse uma tarefa para verificar a situação do prazo e caso o mesmo tivesse excedido essa tarefa acionaria a tarefa de notificar reprovação por prazo. De qualquer forma esse caso de uso necessita de um refinamento após a transformação.

6.3 APLICAÇÃO 2 – PROCESSOS PARA PEDIDO DE REEMBOLSO DE DESPESAS DE VIAGEM

Esse exemplo de aplicação do método realiza a transformação para modelo de casos de uso de um modelo de processo de negócio para pedido de reembolso de despesas de viagem. Esse processo foi modelado considerando que determinadas tarefas serão suportadas por um sistema a ser desenvolvido.

O modelo descreve o processo realizado para solicitar o reembolso de despesas realizadas durante uma viagem de trabalho. Em um primeiro momento o funcionário solicita o reembolso. Esse deve ser analisado pelo seu superior imediato que encaminha, ou não, ao setor financeiro. Esse último é responsável por providenciar o pagamento caso todas as informações estejam corretas. A Figura 70 a seguir apresenta o diagrama BPMN que descreve esse processo.

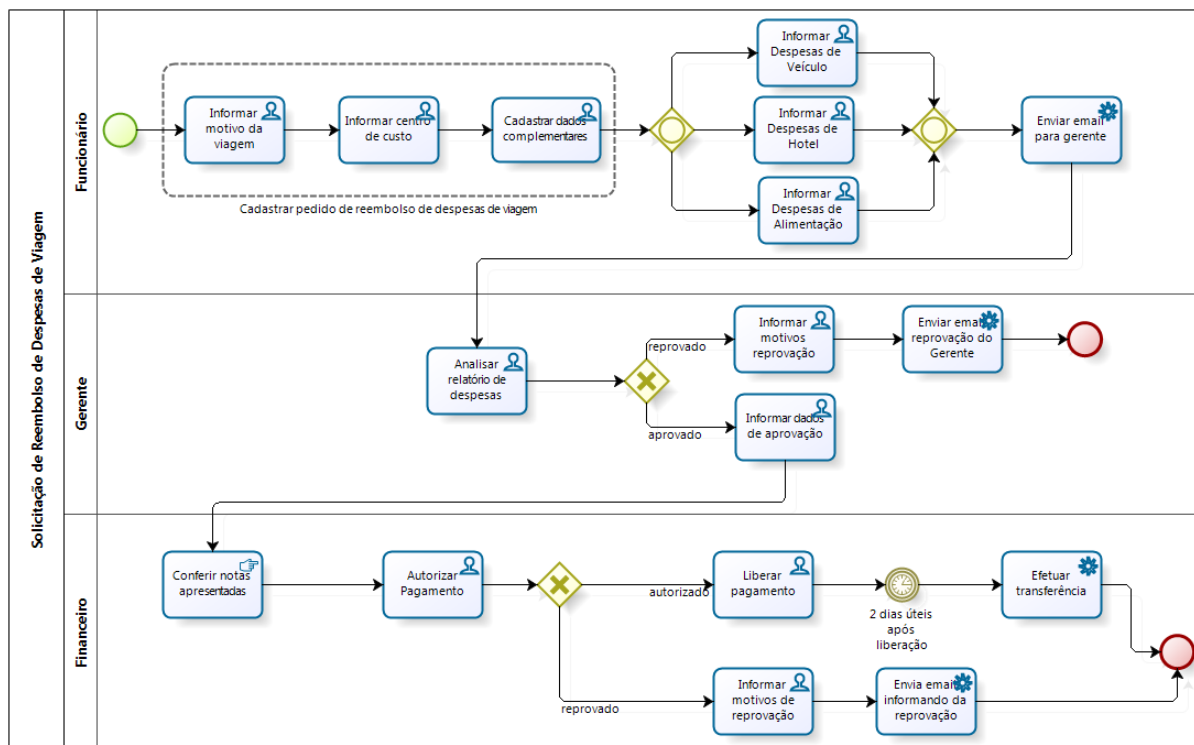


Figura 70 - Processo de Negócio para Solicitação de Reembolso de Despesas de Viagem

6.3.1 REALIZAÇÃO DA TRANSFORMAÇÃO

A primeira atividade do método de transformação consiste em verificar se o modelo de processo de negócio origem está em um nível que permita a identificação de elementos de sistema no mesmo. Conforme mencionado na seção anterior esse modelo apresenta o processo considerando que determinadas tarefas serão suportadas por um sistema a ser desenvolvido. Nessa situação o modelo pode ser transformado para modelo de casos de uso.

A próxima atividade é identificar os atores, casos de uso, relacionamentos e itens da descrição dos casos de uso utilizando como base as possibilidades de transformação descritas no método de transformação. Em seguida, cada caso de uso identificado deve ser refinado pelo analista de sistemas, principalmente na sua descrição. Conforme já mencionado a transformação não gera os casos de uso com as descrições completas. Tanto as descrições como os próprios casos de usos devem ser revisados pelo analista de sistema.

A seguir está um passo a passo descrevendo as transformações realizadas para cada elemento do processo de negócio:

- 1) As Raias Funcionário, Gerente e Financeiro dão origem a atores do sistema com os mesmos nomes.

- 2) O grupo Cadastrar Pedido de Reembolso de Despesas de Viagem dá origem a um caso de uso com esse nome associado ao ator Funcionário.
- 3) Todas as tarefas de usuário e de serviço da raia Funcionário dão origem a casos de uso associado com o ator Funcionário, exceto as tarefas dentro do grupo citado no passo anterior.
- 4) O *gateway* inclusivo que ocorre após o grupo origina relacionamentos de extensão entre o caso de uso Cadastrar Pedido de Reembolso de Despesas de Viagem e os casos de uso Informar Despesas de Veículo, Informar Despesas de Hotel e Informar Despesas de Alimentação. Cada relacionamento dá origem a um ponto de extensão que indica o local do fluxo do caso de uso base onde o comportamento é estendido. Cada ponto de extensão está associado com uma condição para que ocorra. Essa condição está definida no *gateway*. Por exemplo, para informar despesas de veículo é necessário que tenha ocorrido esse tipo de despesa na viagem.

A Figura 71 apresenta o diagrama de casos de uso contendo os associados com o ator Funcionário.

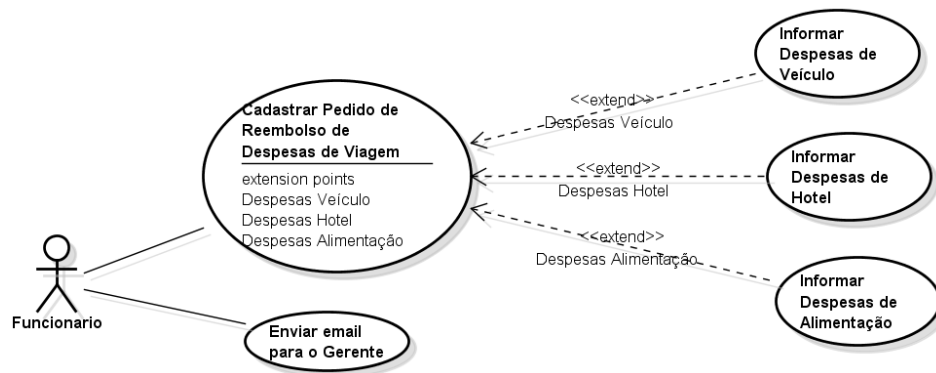


Figura 71 - Casos de uso do ator Funcionário

- 5) As tarefas de usuário da raia Gerente se tornam casos de uso associados com o ator Gerente.
- 6) A tarefa de serviço Enviar email Reprovação do Gerente não se torna outro caso de uso, mas sim um ou mais passos do fluxo básico do caso de uso Informar Motivos Reprovação. A tarefa poderia ter originado um caso de uso separado, mas nessa transformação optou-se por deixar essa ação dentro do caso de uso Informar Motivos de Reprovação, por ser uma ação simples e automática do sistema, não justificando um caso de uso próprio. Esse é um exemplo de decisão tomada ao longo da transformação. O método proposto prevê a ocorrência disso uma vez que definir mapeamentos fixos entre os elementos poderia originar um modelo distorcido e fora da realidade.

- 7) O *gateway* exclusivo após a tarefa Analisar Relatório de Despesas dá origem a relacionamento de generalização entre os casos de uso Analisar Relatório de Despesas, Informar Motivos Reprovação e Informar Dados de Aprovação. O caso de uso Analisar Relatório de Despesas se torna um caso de uso pai. Os outros dois casos de uso mencionados se tornam filhos desse pai.

A Figura 72 a seguir apresenta o diagrama dos casos de uso associados com o ator Gerente.

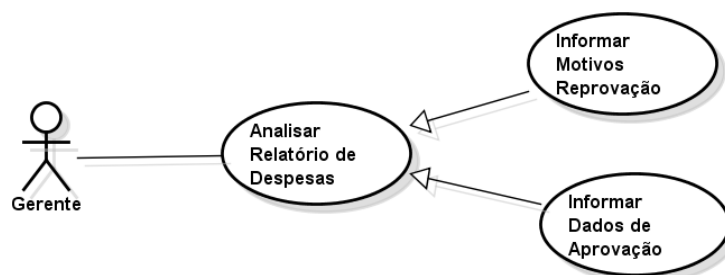


Figura 72 - Casos de uso do ator Gerente

- 8) As tarefas de usuário da raia Financeiro se tornam casos de uso associados ao ator Financeiro.
- 9) O *gateway* exclusivo que ocorre após a tarefa Autorizar Pagamento é transformado em relacionamentos de extensão entre os casos de uso Liberar Pagamentos e Informar Motivos de Reprovação com o caso de uso Autorizar Pagamento. Os pontos de extensão são locais do fluxo básico onde o comportamento pode ser estendido e a ocorrência dos mesmos está associada com a condição definida no *gateway*.
- 10) A tarefa de serviço Enviar email informando da reprovação não é transformada em um caso de uso separado, mas sim em um ou mais passos do fluxo do caso de uso Informar Motivos de Reprovação.
- 11) A tarefa Efetuar Transferência é transformada em um caso de uso com o mesmo nome associado ao ator Financeiro. Essa tarefa é precedida por um evento intermediário temporal. Esse evento se torna o acionador desse caso de uso. Ou seja, ele é disparado quando esse evento ocorrer. Nesse caso dois dias úteis após a liberação é disparado o caso de uso Efetuar Transferência. A informação do acionador não aparece no diagrama de casos de uso, apenas na descrição, conforme pode ser observado na descrição apresentada mais adiante, na Figura 83.

A Figura 73 abaixo apresenta o diagrama com os casos de uso associados ao ator Financeiro.

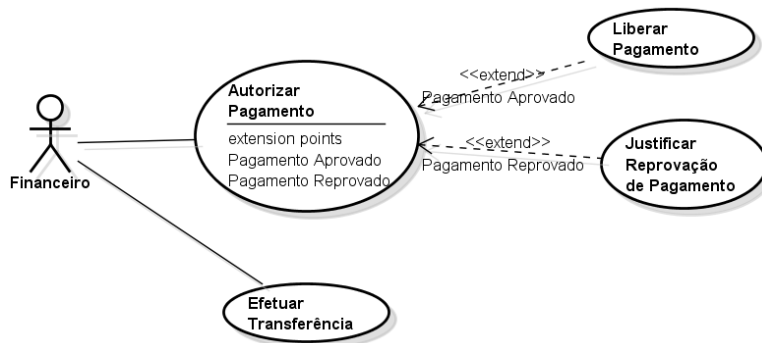


Figura 73 - Casos de uso do ator Financeiro

As descrições iniciais de cada caso de uso também são originadas a partir da aplicação do método. As figuras a seguir apresentam cada uma das descrições geradas com um comentário sobre a mesma.

As figuras de 74 a 76 apresentam as descrições obtidas para os casos de uso Cadastrar Pedido de Reembolso de Despesas de Viagem, Informar Despesas de Veículo e Enviar Email para o Gerente. As descrições dos casos de uso Informar Despesas de Hotel e Informar Despesas de Alimentação são mostradas por serem praticamente iguais a do caso de uso Informar Despesas de Veículo, apenas mudando o tipo da despesa. A transformação realizada é igual, o que mudaria seria algum item de refinamento da própria descrição, mas que não é obtido a partir da transformação. As linhas destacadas são as obtidas pelo método, já as sem destaque são refinamentos inseridos.

Nome do Caso de Uso:	Cadastrar Pedido de Reembolso de Despesas de Viagem
Descrição:	
Lista de Atores:	Funcionário
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	<p>Passo a – Funcionário informa motivo de viagem.</p> <p>Passo b – Funcionário informa centro de custo.</p> <p>Passo c – Funcionário informa dados complementares.</p> <p>... (demais passos de sistema e passos não obtidos pela transformação)</p> <p>Passo d – Funcionário informa se ocorreram despesas de veículo.</p> <p>Passo e – Funcionário informa se ocorreram despesas de hotel.</p> <p>Passo f – Funcionário informa se ocorreram despesas de alimentação.</p>
Fluxos alternativos:	
Pontos de Extensão:	<p>Ponto de Extensão Despesas Veículo <i>Após o Passo d – Funcionário informa se ocorreram despesas de veículo, se sim então o caso de uso Informar Despesas de Veículo é utilizado.</i></p> <p>Ponto de Extensão Despesas Hotel <i>Após o Passo e – Funcionário informa se ocorreram despesas de hotel, se sim então o caso de uso Informar Despesas de Hotel é utilizado.</i></p> <p>Ponto de Extensão Despesas Hotel <i>Após o Passo f – Funcionário informa se ocorreram despesas de alimentação, se sim então o caso de uso Informar Despesas de Alimentação é utilizado.</i></p>
Pós-condições:	

Figura 74 - Descrição do caso de uso Cadastrar Pedido de Reembolso de Viagem

A descrição de caso de uso apresentada na Figura 74 obteve uma parte do seu fluxo básico e todos os pontos de extensão a partir dos elementos do processo de negócio. A obtenção dos passos foi possível devido ao uso do elemento Grupo no diagrama BPMN. Já a descrição obtida para os casos de uso Informar Despesas de Veículo e Enviar Email para o Gerente são mais simples porque não há tantos elementos disponíveis no diagrama para gerar a descrição. É importante ressaltar a possibilidade de se obter uma pré-condição no caso de uso Enviar Email para o Gerente. Essa pré-condição pode existir se o caso de uso Cadastrar Pedido de Reembolso de Despesas de Viagem gerar algum resultado necessário para ser usado pelo Enviar Email para o Gerente. A existência ou não dessa pré-condição dependeria de um maior refinamento dessas descrições.

Nome do Caso de Uso:	Informar Despesas de Veículo
Descrição:	
Lista de Atores:	Funcionário
Pré-condições:	
Acionador:	Caso de uso Cadastrar Pedido de Reembolso de Despesas de Viagem
Fluxo básico de eventos:	... Passo <i>m</i> – Funcionário informa tipo e valor das despesas. Passo <i>n</i> – Sistema salva as informações.
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 75 - Descrição do caso de uso Informar Despesas de Veículo

Nome do Caso de Uso:	Enviar Email para o Gerente
Descrição:	
Lista de Atores:	Funcionário
Pré-condições:	(resultado do caso de uso Cadastrar Pedido de Reembolso de Despesas de Viagem)
Acionador:	
Fluxo básico de eventos:	... Passo <i>m</i> – Sistema envia email informando do cadastro do pedido.
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 76 - Descrição do caso de uso Enviar Email para o Gerente

As figuras de 77 a 79 apresentam as descrições dos casos de uso associados com o ator gerente. O caso de uso pai Analisar Relatório de Despesas apresenta o fluxo principal padrão para os outros dois casos de uso filhos. Esses por sua vez possuem o seu fluxo de eventos específico. O caso de uso pai possui fluxos alternativos que estão relacionados com cada um dos casos de uso filhos.

Nome do Caso de Uso:	Analisar Relatório de Despesas
Descrição:	
Lista de Atores:	Gerente
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	... Passo <i>n</i> – Gerente seleciona situação do Relatório de Despesas do pedido. <i>(até esse ponto a descrição é do caso de uso pai, a partir desse passo é utilizado um dos casos de uso filho, sendo que o fluxo encerra no filho)</i>
Fluxos alternativos:	Fluxo Alternativo 1 – Relatório Aprovado No Passo <i>m</i> o Gerente aprovou o relatório. Utilizar o caso de uso Informar Dados de Aprovação. Fluxo Alternativo 2 – Relatório Reprovado No Passo <i>m</i> o Gerente reprovou o pedido. Utilizar o caso de uso Informar Motivos Reprovação.
Pontos de Extensão:	
Pós-condições:	

Figura 77 - Descrição do caso de uso Analisar Relatório de Despesas

Nome do Caso de Uso:	Informar Dados de Aprovação
Descrição:	
Lista de Atores:	Gerente
Pré-condições:	
Acionador:	Caso de uso Analisar Relatório de Despesas
Fluxo básico de eventos:	... Passo <i>n</i> – Gerente informa dados de aprovação. Passo <i>m</i> – Sistema salva informações de aprovação. <i>(os passos acima são ilustrativos, não foram obtidos a partir do processo de negócio)</i>
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 78 - Descrição do caso de uso Informar Dados de Aprovação

Os passos mostrados na descrição do caso de uso Informar Dados de Aprovação (Figura 78) são resultados de um refinamento. Não foram obtidos pelo método. Já o passo de sistema relacionado ao envio de email do caso de uso Informar Motivos Reprovação foi obtido pela tarefa de serviço contida no diagrama.

Nome do Caso de Uso:	Informar Motivos Reprovação do Relatório
Descrição:	
Lista de Atores:	Gerente
Pré-condições:	
Acionador:	Caso de uso Analisar Relatório de Despesas
Fluxo básico de eventos:	... Passo <i>n</i> – Gerente informa motivos da reprovação. Passo <i>m</i> – Sistema envia email de reprovação do Gerente.
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 79 - Descrição do caso de uso Informar Motivos Reprovação

Os últimos casos de uso gerados pela transformação são os associados com o ator Financeiro. A Figura 80 apresenta a descrição obtida para o caso de uso Autorizar Pagamento.

Nome do Caso de Uso:	Autorizar Pagamento
Descrição:	
Lista de Atores:	Financeiro
Pré-condições:	
Acionador:	
Fluxo básico de eventos:	... Passo <i>m</i> – Financeiro informa situação do pagamento.
Fluxos alternativos:	
Pontos de Extensão:	Ponto de Extensão Pagamento Aprovado No passo <i>m</i> o Financeiro aprovou o pagamento. O caso de uso Liberar Pagamento é disparado. Ponto de Extensão Pagamento Reprovado No passo <i>m</i> o Financeiro reprovou o pagamento. O caso de uso Justificar Reprovação do Pagamento é disparado.
Pós-condições:	

Figura 80 - Descrição do caso de uso de Autorizar Pagamento

O caso de uso Autorizar Pagamento possui duas extensões. Essas estão descritas nas figuras 81 e 82. Em relação ao caso de uso Liberar Pagamento o único item obtido a partir do modelo, além do ator é o acionador.

Nome do Caso de Uso:	Liberar Pagamento
Descrição:	
Lista de Atores:	Financeiro
Pré-condições:	
Acionador:	Caso de uso Autorizar Pagamento
Fluxo básico de eventos:	Passo <i>m</i> – Financeiro informa dados para liberação do pagamento ...(demais passos)
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 81 - Descrição do caso de uso Liberar Pagamento

A descrição obtida para o caso de uso Informar Motivos de Reprovação do Pagamento possui além do acionador um passo do fluxo básico executado pelo sistema relacionado a ação de enviar email, obtido através da tarefa de serviço.

Nome do Caso de Uso:	Informar Motivos de Reprovação do Pagamento
Descrição:	
Lista de Atores:	Financeiro
Pré-condições:	
Acionador:	Caso de uso Autorizar Pagamento
Fluxo básico de eventos:	Passo <i>m</i> – Financeiro informa motivos da reprovação. Passo <i>n</i> – Sistema envia email informado da reprovação.
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 82 - Descrição do caso de uso Informar Motivos de Reprovação do Pagamento

O último caso de uso gerado nesse exemplo é o de Efetuar Transferência. Esse caso de uso foi originado a partir de uma tarefa precedida por um evento intermediário temporal. Na descrição desse caso de uso esse evento é colocado como seu acionador, conforme apresentado na Figura 83.

Nome do Caso de Uso:	Efetuar Transferência
Descrição:	
Lista de Atores:	Financeiro
Pré-condições:	
Acionador:	Dois dias úteis após liberação do pagamento
Fluxo básico de eventos:	Passo <i>n</i> – Sistema realiza transferência do pagamento.
Fluxos alternativos:	
Pontos de Extensão:	
Pós-condições:	

Figura 83 - Descrição do caso de uso Efetuar Transferência

6.3.2 CONSIDERAÇÕES SOBRE O RESULTADO OBTIDO

A transformação do processo de negócio de Solicitação de Reembolso de Despesas de Viagem deu origem a um modelo com doze casos de uso. Analisando os casos de uso obtidos verifica-se que os mesmos são efetivamente casos de uso de sistema e que podem ser utilizados como base para a descrição dos requisitos do sistema a ser desenvolvido para atender o negócio. O número de casos de uso obtido poderia ser diferente se durante o processo de transformação fossem adotadas diferentes alternativas que o método apresenta. É o exemplo das tarefas de serviço relacionadas aos envios de email das raiais do Gerente e Financeiro. Essas não deram origem a casos de uso separado, mas sim a passos de casos de uso gerados por tarefas precedentes as de serviço.

Outro exemplo de decisão de quem está executando o método é em relação a utilizar relacionamento de generalização ou extensão para transformar *gateways* exclusivos. No exemplo do *gateway* associado à tarefa de Analisar Relatório de Despesas, esse deu origem a um relacionamento de generalização entre casos de uso. Já na raia do Financeiro optou-se por gerar relacionamento de extensão entre os casos de uso. A decisão entre generalização e extensão depende do tipo de relação que se quer expressar. Na generalização o caso de uso pai é utilizado apenas para conter os passos que são comuns aos filhos para que esses passos não precisem ser repetidos. Geralmente esse caso de uso é abstrato. Já no relacionamento de extensão o caso de uso base tem uma função específica e o seu comportamento pode ser estendido por outros casos de uso em determinadas condições.

Na transformação realizada o uso da generalização para o caso de uso Analisar Relatório de Despesas significa que o mesmo representa apenas um conjunto de procedimentos que são iguais para os casos de uso Informar Motivos de Reprovação e Informar Dados de Aprovação. Dessa forma o usuário vai usar esse caso de uso apenas para uma dessas duas finalidades, aprovar ou não o relatório. Já no caso do Autorizar Pagamento esse caso de uso pode ter outras ações que não necessariamente liberar ou reprovar o pagamento. A liberação ou reprovação do mesmo seriam funções adicionais desse caso de uso. O usuário poderia utilizar o mesmo para informar alguns dados pertinentes à autorização e só depois autorizar ou não o pagamento.

Em relação ao diagrama de casos de uso originado pode se considerar o mesmo como satisfatório, uma vez que os casos de uso obtidos são pertinentes a um sistema que atenda o modelo de processo de negócio utilizado como origem, apresentando uma visão geral do sistema a ser desenvolvido.

No que diz respeito às descrições dos casos de uso pode-se destacar a obtida no caso de uso Cadastrar Pedido de Reembolso de Despesas de Viagem. Essa descrição contém passos do fluxo básico e pontos de extensão que foram originados a partir da transformação do elemento Grupo e *gateway*. Isso mostra que a utilização do elemento Grupo auxilia na geração de descrições mais completas de casos de uso. O uso do Grupo permite ao analista uma maior liberdade para definir quais tarefas e/ou outros elementos que em conjunto podem dar origem a um caso de uso.

O último destaque em relação a essa transformação está no tratamento do evento intermediário temporal. Esse evento foi transformado em um acionador do caso de uso Efetuar Transferência. Isso implica dizer que o próprio sistema dispara esse caso de uso quando essa condição temporal for verdadeira. O ator do caso de uso permanece o Financeiro, que é o principal interessado, mas o acionador é o tempo. Os detalhes de como o caso de uso é disparado não são abordados no nível dos casos de uso sendo decisões no nível de projeto (*design*) do sistema.

7 CONSIDERAÇÕES FINAIS

Essa pesquisa apresentou um método para ser utilizado na transformação de modelos de processos de negócio expressos utilizando a BPMN para modelos de casos de uso da UML. O método descrito auxilia o analista de sistemas a extrair os casos de uso de sistema a partir de um modelo de processo de negócio construído em BPMN.

O entendimento entre os profissionais de negócio e profissionais de sistema é fundamental para que se construam sistemas eficientes e que atendam as necessidades de negócio da organização. De nada adianta um sistema bem implementado em termos técnicos que não atenda as expectativas do negócio.

O método apresentado procura minimizar a distância que existe entre o modelo feito pelo analista de negócio na BPMN e o modelo de casos de uso da UML, este último mais utilizado por profissionais da área de desenvolvimento de sistemas. Para isso foi estudada a notação BPMN e quais os conceitos de cada elemento dessa notação, procurando encontrar relações entre esses conceitos com elementos de casos de uso. Esse estudo abordou os metamodelos da BPMN e UML procurando estabelecer essas conexões em nível dos metamodelos, atendendo assim as indicações da MDA para transformações entre modelos.

A questão de pesquisa definida foi “Como transformar modelos de processo de negócio, feitos em BPMN, para modelos de sistema expressos através de casos de uso da UML?”. Com base na análise dos trabalhos relacionados percebeu-se limitações no tratamento utilizado para o processo de transformação entre modelos. As propostas existentes identificam determinados elementos dos casos de uso assim como analisam um conjunto limitado de elementos gráficos da BPMN. Também ocorre de cada estudo propor a mesma transformação de formas diferentes. A solução proposta no presente trabalho apresenta alternativas para realizar a transformação de cada elemento. A decisão de qual alternativa utilizar fica a critério de quem estiver realizando a transformação. As descrições dessas alternativas apresentadas no Capítulo 5 auxiliam nessa escolha.

A resposta para a questão de pesquisa é que a transformação pode ser feita analisando cada elemento do modelo descrito em BPMN e verificando como esse pode ser transformado em um elemento de caso de uso. A transformação não é automática, sendo necessárias decisões ao longo da mesma. Também é necessário um refinamento no modelo de casos de uso gerado, uma vez que nem todas as informações que compõem esse modelo estão no processo de negócio. Resumindo, a transformação gera uma versão inicial do modelo de casos de uso.

Para que a transformação seja mais completa é necessário que cada vez mais os processos de negócio definidos identifiquem informações relevantes para a construção dos modelos de sistema. Mais especificamente no assunto dessa pesquisa seria interessante se os modelos em BPMN

contivessem informações que permitissem uma transformação mais precisa para elementos da UML. O ideal seria uma espécie de unificação entre BPMN e UML permitindo criar modelos que representem tanto o negócio como o sistema e que esses modelos fossem consistentes entre si. Essa consistência teria que ser garantida por essa linguagem/notação.

A principal contribuição desse trabalho para o assunto relacionado à transformação de modelos de negócio para sistema é o método de transformação definido. Esse método apresenta várias possibilidades de transformação para cada elemento da BPMN, verificando como esse pode gerar um elemento de caso de uso de sistema. Utilizando o método é possível obter uma versão inicial dos diagramas de casos de uso juntamente com as descrições iniciais dos mesmos, conforme pode ser observado nas aplicações de exemplo apresentadas no Capítulo 6.

Além dessa contribuição principal pode-se destacar também como contribuição a análise da transformação de elementos da BPMN como artefatos, eventos e diferentes tipos de tarefas. A geração de elementos da descrição dos casos de uso, não somente do modelo de casos de uso também constitui uma contribuição adicional.

7.1 LIMITAÇÕES DO ESTUDO

O método apresentado atende ao objetivo proposto que é transformar um modelo de processos de negócio em BPMN para modelo de casos de uso. Entretanto, algumas limitações devem ser consideradas:

- Processos com mais de um participante (*Pool*) e conseqüentemente troca de mensagens entre participantes não são contempladas;
- Apenas eventos com causa temporal e condicional são abordados nas transformações;
- A abordagem aos elementos dos metamodelos se limita a identificação das classes relacionadas com a transformação, não sendo definidas regras formais para aplicação das transformações.

7.2 TRABALHOS FUTUROS

A resolução das limitações apresentadas na seção anterior são sugestões de possíveis trabalhos futuros. Além dessas também pode-se citar:

- A definição de algoritmos para as transformações dos elementos no nível dos metamodelos utilizando uma linguagem de transformação de modelos, como a QVT. O uso de uma linguagem de transformação entre modelos pode automatizar algumas transformações definidas no método.

- Identificação de transformações possíveis para geração das classes de domínio a partir de um processo de negócio. Dessa forma é possível a geração de modelo de classes de domínio além do modelo de casos de uso.
- Verificação de padrões de ocorrência de cada alternativa de transformação procurando diminuir a intervenção humana no processo. Para isso podem ser utilizados padrões de *workflow* [THO09];
- Realização de um estudo de caso ou um experimento controlado aplicando o método proposto em um ambiente corporativo.

A realização dos estudos sugeridos acima é importante para a continuidade do trabalho apresentado nesta pesquisa, uma vez que o desafio de transformar o entendimento em nível de negócio para um sistema de informação adequado continua sendo um desafio diário dentro das organizações.

REFERÊNCIAS BIBLIOGRÁFICAS

- [AAL98] AALST, W. "The application of Petri Nets to workflow management". *The Journal of Circuits, Systems and Computers*, vol. 8-1, 1998, pp. 21-66.
- [AMB05] AMBLER, S. "The Elements of UML(TM) 2.0 Style". Cambridge University Press, 2005, 200p.
- [AZE08] AZEVEDO JUNIOR, D; CAMPOS, R. "Systematization of Requirements Definition for Software Development Processes with a Business Modeling Architecture". *IFIP International Federation for Information Processing, Research and Practical Issues of Enterprise Information Systems II*, vol. 1, 2008, pp. 1571-5736 (Print).
- [BAE07] BAE, J; KANG, S. "A Method to Generate a Feature Model from a Business Process Model for Business Applications". In: 7th IEEE International Conference, 2007, pp. 879-884.
- [BAR02] BARI, M., LÉVESQUE, G. and SEFFAH, A. "Gathering Additional Requirements via Temporal Actors". In: Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002), 2002, 4p.
- [BAS02] BASTOS, R; DUBUGRAS, D. "Extending UML Activity Diagram for Workflow Modeling in Production Systems". In: HICSS '02 Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), 2002, 10p.
- [BER09] BERROCAL, J. ALONSO, J. RODRÍGUEZ, J. "Patrones para la Extracción de Casos de Uso a partir de Procesos de Negocio". *Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos*, vol. 3-3, 2009, pp. 1-11.
- [BOO05] BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. "The Unified Modeling Language User Guide". Addison-Wesley Professional; 2ª. ed, 2005, 496p.
- [BPM09] OBJECT MANAGEMENT GROUP. "Business Process Model and Notation (BPMN)". Versão 1.2. Capturado em <http://www.omg.org>, 2010.
- [COC05] COCKBURN, A. "Escrevendo Casos de Uso Eficazes". Porto Alegre: Bookman, 2005. 254p.
- [CON01] CONSTANTINE, L; LOCKWOOD, L. "Structure and style in use cases for user interface design, Object modeling and user interface design: designing interactive systems". Boston, MA: Addison-Wesley Longman Publishing Co., 2001, 452p.
- [DAM08] DAMIJ, N; DAMIJ, T; GRAD, J; JELENC, F. "A methodology for business process improvement and IS development". *Information and Software Technology*, vol. 50-11, Out 2008, pp. 1127-1141.
- [DIJ02] DIJKMAN, R; JOOSTEN, S. "Deriving Use Case Diagrams from Business Process Models". CTIT Technical Report Series, Relatório Técnico, Holanda, 2002, 6p.

- [ERI00] ERIKSSON, H; PENKER, M. “Business Modeling with UML”. John Wiley Professional, 1a ed, 2000, 480p.
- [EST03] ESTRADA, H; MARTÍNEZ, A; PASTOR, O. “Goal-Based Business Modeling Oriented towards Late Requirements Generation”. *Conceptual Modeling - ER 2003 In Conceptual Modeling*, vol 2813, 2003, pp. 277-290.
- [HAR04] HARMON, P. “The OMG’s model driven architecture and BPM”. Business Process Trends. Capturado em <http://www.bptrends.com>, 2011.
- [HAR10] HARMON, P. “BPMN for Business – The Role of The Customer”. *Business Process Trends*, 2010. Capturado em <http://www.bptrends.com>, 2010.
- [HER10] HEREDIA, L.; BASTOS, R. “Abordagens de transformações de modelos de processos de negócio para modelos de sistema”. Artigo de Introdução à Pesquisa II, PPGCC, PUCRS, 2010, 10p.
- [JAC92] JACOBSON, I. CHRISTERSON, M. JONSSON, P. VERGAARD, O. “Object Oriented Software Engineering: A Use Case Driven Approach”. Workingham: Addison-Wesley, 1992, 552p.
- [JAC98] JACOBSON, I.; GRADY, B.; RUMBAUGH, J. “The Unified Software Development Process”. New York: Addison-Wesley, 1998. 463p.
- [KRU00] KRUCHTEN, P. “The Rational Unified Process: An Introduction”. Upper Sadle River, New Jersey: Addison-Wesley, 2ª. ed, 2000, 320p.
- [LAR04] LARMAN, C. “Utilizando UML e Padrões”. Porto Alegre: Bookman, 2ª. ed, 2004, 607p.
- [LAU03] LAUESEN, S. “Task descriptions as functional requirements”. *IEEE Software*, vol. 20-2, Mar-Abr 2003, pp. 58-65.
- [LEF00] LEFFINGWELL, D; WIDRIG, D. “Managing Software Requirements: A Unified Approach”. England: Addison-Wesley, 1a ed, 2000, 544p.
- [LIE04] LIEW, P; KONTOGIANNIS, K; TONG, T. “A framework for business model driven development”. In: Software Technology and Engineering Practice The 12th International Workshop, 2004, 8p.
- [LUN03] LUNN, K; SIXSMITH, A; LINDSAY, A; VAARAMA, M. “Traceability in requirements through process modelling, applied to social care applications”. *Information and Software Technology*, Vol. 45-15, Dez 2003, pp. 1045-1052.
- [MAC01] MACIASZEK, L. “Requirements Analysis and System Design: Developing Information Systems with UML”. Addison-Wesley, 2001, 378p.
- [MEN06] MENS, T. VAN GORP, P. “A taxonomy of model transformation”. *Electronic Notes in Theoretical Computer Science*, vol. 152, Mar 2006, pp. 125-142.

- [MUT08] MUTSCHLER, B.; REICHERT, M.; BUMILLER, J. “Unleashing the Effectiveness of Process-oriented Information Systems: Problem Analysis, Critical Success Factors and Implications”. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 38-3, pp. 280-291.
- [OAT06] OATES, B. J. “Researching Information Systems and Computing”. Sage Publications Ltd., 2006, 360p.
- [ODE03] ODEH, M.; KAMM, R. “Bridging the gap between business models and system models”. *Information and Software Technology*, vol. 45-15, 2003, pp. 1053-1060.
- [OKA07] OKAWA, T.; HIRABAYASHI, S.; KAMINISHI, T.; KOIZUMI, H.; SAWAMOTO, J. “A Method of Linking Business Process Modeling with Information System Design Using UML and its Evaluation by Prototyping”. In: Asia-Pacific Service Computing Conference The 2nd IEEE, 2007, pp. 458-465.
- [OMG03] OBJECT MANAGEMENT GROUP. “MDA Guide Version 1.01”. Capturado em: http://www.omg.org/mda/mda_files/MDA_Guide_Version1-0.pdf, 2011.
- [OMG05] OBJECT MANAGEMENT GROUP. “Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, OMG Adopted Specification ptc/05-11-01”, Capturado em <http://www.omg.org>, 2011.
- [OMG09b] OBJECT MANAGEMENT GROUP. “BPMN Elements and Attributes”. Capturado em: http://www.bpmn.org/Documents/BPMN_Elements_and_Attributes.pdf, 2011.
- [PHA98] PHALP, K. “The CAP framework for business process modeling”. *Information and Software Technology*, vol. 40-13, Nov 1998, pp. 731-744.
- [PRE06] PRESSMAN, R. “Engenharia de Software”. São Paulo: McGraw-Hill, 6ª. ed, 2006, 752p.
- [RAT02] RATIONAL SOFTWARE. “Is the Clock an Actor? By Antony Crain”. The Rational Edge. Capturado em <http://www.ibm.com/developerworks/rational/library/content/RationalEdge/jun02/DrUseCaseJun02.pdf>, 2011.
- [ROD10] RODRÍGUEZ, A; GUZMÁN, I; FERNÁNDEZ-MEDINA, E; PIATTINI, M. “Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach”. *Information and Software Technology*, vol. 52-9, Set 2010, pp. 945-971.
- [SAN02] SANTANDER, V.; CASTRO, J. “Deriving Use Cases from Organizational Modeling”. In: Proceedings of the 10th Anniversary IEEE Joint International Conference on Requirements Engineering (Washington, DC, USA, 2002), 2002, pp. 32-42.
- [SCH05] SCHEER, A; THOMAS, O; ADAM, O. “Process Modeling Using Event-Driven Process Chains”. In: Proceedings of the Workshop on Semantics for Business Process Management SBPM 06 held at the 3rd European Semantic Web Conference ESWC, 2005, pp. 119-145.

- [SIN09] SINGH, S; WOO, C. “Investigating business-IT alignment through multi-disciplinary goal concepts”. *Journal Requirements Engineering*, vol. 14-3, 2009, pp. 177-207.
- [SOM07] SOMMERVILLE, I. “Engenharia de Software”. São Paulo: Addison Wesley, 8ª. ed, 2007. 552p.
- [SOM09] SOMÉ, S. “A Meta-Model for Textual Use Case Description”. In *Journal of Object Technology*, vol. 8-7, Nov-Dez 2009, pp. 87-106.
- [THO09] THOM, L.; REICHERT, M.; IOCHPE, C. “Activity Patterns in Process-aware Information systems. Basic Concepts and Empirical Evidence”. *IJBPM - International Journal of Business Process and Information Management*, vol.4-2, 2009, pp. 93-110.
- [UML11] OBJECT MANAGEMENT GROUP. “Unified Modeling Language (UML): Superstructure. Versão 2.4.1”, Capturado em <http://www.omg.org>, 2011.
- [VAR08] VARA, J; SÁNCHEZ, J. “Improving Requirements Analysis through Business Process Modelling: A Participative Approach”. In: 11th International Conference, BIS, Innsbruck, Austria, 2008, 13p.
- [VER96] VERNADAT, F. “Enterprise Modeling and Integration”. London: Chapman & Hall, 1996, 496p.
- [WAT08] WATSON, A. “Visual Modelling: past, present and future”. OMG White Paper. Capturado em http://www.uml.org/Visual_Modeling.pdf, 2010.
- [WAZ09] WAZLAWICK, R. “Metodologia de Pesquisa para Ciência da Computação”. Rio de Janeiro: Elsevier, 2009, 184p.
- [WEI07] WEILKIENS, T.; OESTEREICH, B. “UML 2 Certification Guide”. San Francisco: *OMG Press*, 2007, 320p.
- [WES07] WESKE, M. “Business Process Management”. Berlin: Springer, 2007, 368p.
- [WHI09] WHITE, S.; MIERS, D. “BPMN Modeling and Reference Guide”. Lighthouse Pt: Future Strategies Inc, 2008, 229p.
- [WMC99] WORKFLOW MANAGEMENT COALITION (WMC). “Terminology and Glossary. Bruxelas, 65p”. Capturado em <http://www.wfmc.org>, 2010.