

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MIKE MUYA TSHIBENDE

**Extração de Fragmentos de Diagramas de
Classes a partir de Modelos de Processos de
Negócio**

Monografia apresentada como requisito parcial para
a obtenção do grau de Bacharel em Ciência da
Computação

Orientador: Prof^a. Dr^a. Lucinéia Heloisa Thom

Porto Alegre
2015

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling Franco

Diretor do Instituto de Informática: Prof. Luis da Cunha Lamb

Coordenador do Curso de Ciência de Computação: Prof. Carlos Arthur Lang Lisboa

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

RESUMO

A modelagem de processo de negócio tem atraído a atenção dos analistas de sistemas, pois auxilia a modelagem, análise e projeção de sistemas a serem desenvolvidos. No entanto, a notação usada para modelar processos, BPMN (Notação para Modelagem de Processos de Negócio) é diferente da usada para modelar sistemas, diagrama de classes da UML (Linguagem de Modelagem Unificada). Assim, este trabalho apresenta um estudo para facilitar a extração de diagramas de classes a partir de um modelo de processo de negócio em BPMN. Como resultado deste estudo, tem-se um conjunto de regras para auxiliar a extração de modelos de sistema, a partir de modelos em nível de processo de negócio. Com o diagrama de classes obtido, o analista de sistemas poderá ter uma visão estática do vocabulário do sistema, incluindo os diferentes relacionamentos entre as classes e poderá ter uma visão lógica de um esquema de banco de dados.

Palavras-chave: Modelagem de processo de negócio. Diagrama de classes. BPMN. UML.

Extraction of Class Diagram Fragments from Business Process Models

ABSTRACT

The business process modeling has attracted the attention of systems analysts, as it helps to modeling, analyzing and designing systems to be developed. Nonetheless, the notation used to model the processes, BPMN (Business Process Management Notation), differs from the one used to model systems, UML class diagrams. Thereby, this work aim at presenting a study in order to facilitate class diagram extraction from a business process model represented by the BPMN. As result of this study, a methodology has been developed to help the extraction of system models, starting from models in business level. With the class diagram obtained, the systems analyst may have a static view of the system's vocabulary, including the different relationships between classes, as well as a logical view of a database schema.

Keywords: Business Process Modeling. Class Diagram. BPMN. UML.

LISTA DE FIGURAS

Figura 2.1	Ciclo de vida BPM (DUMAS et al., 2013).....	13
Figura 2.2	Modelo de processos de negócio em BPMN para solicitação de compras.....	16
Figura 2.3	Subprocesso “Enviar Fatura” detalhado	16
Figura 2.4	Representação da BPMN para evento de início sem causa definida	17
Figura 2.5	Representação da BPMN para tipos de eventos (SPECIFICATION, 2006).....	17
Figura 2.6	Representação da BPMN para evento intermediário	17
Figura 2.7	Representação da BPMN para evento de fim	18
Figura 2.8	Representação da BPMN para tipos de tarefas.....	18
Figura 2.9	Representação da BPMN para tipos de atividades	19
Figura 2.10	Representação da BPMN para tipos de subprocessos	19
Figura 2.11	Representação da BPMN para tipos de desvios	19
Figura 2.12	Representação da BPMN para o objeto de dados “Ordem de compra”	20
Figura 2.13	Representação da BPMN para objetos de dados, dados de entrada/saída, re- positório de dados	20
Figura 2.14	Representação da BPMN para fluxo de sequencia	21
Figura 2.15	Representação da BPMN para fluxo de mensagem.....	21
Figura 2.16	Representação da BPMN para associação.....	21
Figura 2.17	Representação da BPMN para piscina e raias	21
Figura 2.18	Exemplo de piscina Empresa com suas respectivas raias: Setor Vendas e Setor Compras.....	22
Figura 2.19	Exemplo de anotação e grupo.....	22
Figura 3.1	Exemplo de classe.....	24
Figura 3.2	Exemplo de responsabilidades de classe	24
Figura 3.3	Exemplo de associação entre as classes (BOOCH, 1999).....	25
Figura 3.4	Exemplo de agregação e composição	26
Figura 3.5	Exemplo de generalização	26
Figura 3.6	Dependencia	26
Figura 3.7	Tranformação entre modelos (GROUP et al., 2003)	28
Figura 4.1	Diagrama de classe a partir de um diagrama de workflow (OKAWA et al., 2007)	31
Figura 4.2	Processo de negócio modelado através do diagrama de atividades (SUAREZ, 2008)	32
Figura 4.3	Diagrama de classes correspondente ao modelo da figura 4.2 (SUAREZ, 2008)	33
Figura 5.1	Exemplo de mapeamento de piscina para classe	36
Figura 5.2	Exemplo de mapeamento de raias para classes	37
Figura 5.3	Exemplo de mapeamento de atividades para responsabilidades de classe	37
Figura 5.4	Exemplo de mapeamento da parte “objeto” do rótulo de atividades para classes..	38
Figura 5.5	Exemplo de relacionamentos entre as classes obtidas a partir de piscina e raias ...	39
Figura 5.6	Exemplo de relacionamentos entre as classes obtidas a partir de piscina e de rótulos de atividades	39
Figura 5.7	Exemplo de mapeamento de desvio exclusivo em um atributo status ou em um relacionamento de generalização	40
Figura 5.8	Exemplo de mapeamento de evento temporal para a anotação	40

Figura 5.9 Regras de mapeamento entre processo de negócio em BPMN e diagramas de classes da UML.....	41
Figura 5.10 Modelo de processo de fábrica de cadeiras de escritório	42
Figura 5.11 Aplicação das regras de mapeamento no modelo de processos da figura 5.10	43
Figura 5.12 Diagrama de classes correspondente ao processo de solicitação de cadeiras de escritório.....	43
Figura 6.1 Modelo de processo de locação de veículo.....	44
Figura 6.2 Fragmento de diagrama de classes, adaptado de Guedes (2011), referente ao modelo de processo de negócio apresentado na figura 6.1	45
Figura 6.3 Conhecimento dos alunos em BPMN e em diagrama de classes	45
Figura 6.4 Classes vs número de alunos	46
Figura 6.5 Satisfação dos alunos com as regras de mapeamento.....	46
Figura A.1 Modelo de processo de locação de veículo.....	49
Figura A.2 Modelo de processo de locação de veículo.....	50

LISTA DE TABELAS

Tabela 5.1 Elementos notacionais da BPMN considerados neste trabalho.....	35
---	----

LISTA DE ABREVIATURAS E SIGLAS

BPMN	<i>Notação para Modelagem de Processos de Negócio</i>
BPM	<i>Gerenciamento de Processos de Negócio</i>
UML	<i>Linguagem de Modelagem Unificada</i>
ER	Entidade-Relacionamento
UFRGS	Universidade Federal do Rio Grande do Sul
WfMS	<i>Sistema de Gerenciamento de Workflow</i>
ERP	<i>Enterprise Resource Planning</i>
ADF	<i>Activity-Decision Flow</i>
EPCs	<i>Event-Process Chains</i>
IDEF	<i>Icam DEFinition for Function Modeling</i>
ebXML	<i>e-business XML</i>
BPSS	<i>Business Performance System Solution</i>
LOVeM	<i>Line of Visibility Enterprise Modeling</i>
OMG	<i>Object Management Group</i>
MDA	<i>Model-Driven Architecture</i>
CIM	<i>Computation Independent Model</i>
PIM	<i>Platform Independent Model</i>
PSM	<i>Platform Specific Model</i>
QVT	<i>Query View Transformation</i>

SUMÁRIO

1 INTRODUÇÃO	10
2 FUNDAMENTOS EM MODELAGEM DE PROCESSO DE NEGÓCIO E INTRODUÇÃO À BPMN	12
2.1 Conceitos básicos em modelagem de processos de negócio	12
2.1.1 Ciclo de vida da BPM	13
2.1.2 Sistemas de Gerenciamento de <i>Workflow</i>	14
2.2 Notação para a modelagem de processos de negócio (BPMN)	15
2.2.1 Objetos de fluxo	16
2.2.2 Dados	20
2.2.3 Objetos de conexão	20
2.2.4 Partições	21
2.2.5 Artefatos	22
3 MODELAGEM DE SISTEMAS ATRAVÉS DO DIAGRAMA DE CLASSE DA UML E INTRODUÇÃO A <i>MODEL-DRIVEN ARCHITECTURE</i>	23
3.1 Classes	23
3.2 Relacionamentos	24
3.3 <i>Model-Driven Architecture</i> e transformação entre modelos	27
4 TRABALHOS RELACIONADOS	29
4.1 <i>A method of linking business process modeling with information system desing using UML and its evaluation by prototyping</i> (OKAWA et al., 2007)	30
4.2 <i>Transformation of a process business model to domain model</i> (SUAREZ, 2008)	31
4.3 Considerações sobre os trabalhos relacionados	33
5 MAPEAMENTO DE MODELOS DE PROCESSO DE NEGÓCIO PARA DIAGRAMAS DE CLASSES	35
5.1 Verificação de um modelo de processo de negócio	36
5.1.1 Mapeamento de piscina	36
5.1.2 Mapeamento de raia	37
5.1.3 Mapeamento de atividade	37
5.1.4 Mapeamento de rótulos das atividades	38
5.1.5 Relacionamentos entre classes obtidas a partir de piscinas e raias	38
5.1.6 Relacionamentos entre classes obtidas a partir de piscina (ou raias) e as obtidas a partir de rótulos de atividades	39
5.1.7 Mapeamento de desvios	39
5.1.8 Mapeamento de eventos	40
5.1.9 Exemplo ilustrativo de mapeamento completo das regras propostas	41
6 EXPERIMENTO PARA VALIDAÇÃO DO MÉTODO PROPOSTO	44
6.1 Metodologia adotada	44
6.2 Análise dos resultados obtidos	45
7 CONCLUSÃO	47
REFERÊNCIAS	48
APÊNDICE A — QUESTIONÁRIO APLICADO	49
A.1 Questionário Grupo A	49
A.2 Questionário Grupo B	49
APÊNDICE B — ARTIGO SUBMETIDO	51

1 INTRODUÇÃO

Um processo de negócio consiste em uma série de atividades relacionadas que são realizadas para atingir um determinado objetivo de negócio (WESKE, 2012). Um Processo de negócio permite ao analista compreender melhor o negócio e conseqüentemente auxilia esse analista a melhorá-lo e torná-lo mais flexível para mudanças que surgem no mercado.

A notação BPMN (Notação para Modelagem de Processos de Negócio) é uma das mais populares e mais usadas pelos analistas de negócio (RODRÍGUEZ et al., 2010). BPMN objetiva fornecer uma notação de fácil entendimento por todas as partes envolvidas, incluindo os analistas de negócios, os desenvolvedores e finalmente aqueles responsáveis por gerenciar e monitorar esses processos (SPECIFICATION, 2006).

Considerando especificamente a modelagem de sistemas no desenvolvimento de software, a linguagem UML (Linguagem de Modelagem Unificada) é considerada padrão (FORCE, 1999). Dos diferentes diagramas da UML, o diagrama de classes é aquele encontrado com maior frequência no contexto de orientação a objetos (BOOCH, 1999). Um diagrama de classes serve para modelar a visão estática do projeto de um sistema (BOOCH, 1999) e pode ser considerado uma notação alternativa aos diagramas Entidade-Relacionamento (ER) usados em projetos de bancos de dados (NAVATHE, 2005).

Analistas de negócio e analistas de sistemas usam notações e linguagens diferentes para realizar suas tarefas. Como ambos tipos de profissionais trabalham em níveis de abstração diferentes, o mapeamento de um modelo em nível de processo de negócio para um modelo em nível de sistema pode se tornar uma tarefa difícil de ser realizada. Assim, o analista de sistemas precisa ter conhecimento sobre BPMN para entender as diferentes semânticas existentes nos elementos desses modelos.

O mapeamento de um modelo em nível de processo de negócio para um modelo em nível de sistema vem sendo objeto de diversos estudos na área de Engenharia de Software. Analisando essa questão a partir de uma abordagem de MDA (*Model-Driven Architecture*), verifica-se que os modelos de processos de negócio se encontram em um nível CIM (Computing Independent Model), enquanto que os modelos de sistemas localizam-se nos níveis de PIM (Platform Independent Model) e de PSM (Platform Specific Model) (MILLER et al., 2003). Assim, o mapeamento entre um modelo em um nível CIM para um modelo do nível PIM e, posteriormente, de nível PSM, pode ser realizado usando os princípios de transformação entre modelos propostos pela MDA.

Existem diversos trabalhos que tratam do problema de mapeamento de modelos de pro-

cesso de negócio para modelos de sistemas. Na maioria dos trabalhos encontrados ((SUAREZ, 2008),(OKAWA et al., 2007),(BARROS, 2000)), os modelos de processo são representados através de notações diferentes da BPMN. Em (RODRÍGUEZ et al., 2010) é apresentado uma proposta de mapeamento de processo em BPMN para diagrama de atividades da UML. O diagrama de atividades obtido é transformado em diagramas de classes e diagramas de casos de uso. Porém, não é explicado claramente como foram estabelecidos os relacionamentos entre as classes.

O objetivo deste trabalho é apresentar um conjunto de regras que auxilie no mapeamento de modelos em nível de processo de negócio (ou seja, em nível CIM), especificados em BPMN, para modelos em nível de sistema (em nível PIM), especificados em diagramas de classes da UML. Esse trabalho visa propor um mapeamento manual com assistência de um computador. O mapeamento automático pode ser assunto de um trabalho futuro e pode ser feito com base nas regras de mapeamento propostas neste trabalho.

Como benefício, as regras propostas visam auxiliar na identificação de classes e seus possíveis relacionamentos a partir do modelo de processo de negócio. Além disso, as regras propostas possibilitam ao analista de sistemas obter uma visão lógica de um esquema de banco de dados, e podem ser usadas para criar uma ferramenta que automatize esse mapeamento.

Para validar as regras de mapeamento foi realizado um experimento com alunos de graduação e pós-graduação da disciplina de Engenharia de Software do curso de Ciência da Computação do Instituto de Informática da UFRGS. Melhores resultados foram obtidos com os alunos que usaram as regras propostas.

O restante do trabalho está estruturado da seguinte forma: o capítulo 2 trata dos fundamentos em modelagem de processo de negócio e faz uma introdução à BPMN. O capítulo 3 trata dos fundamentos em diagramas de classes da UML e sobre a MDA. O capítulo 4 trata dos trabalhos relacionados. O capítulo 5 apresenta as regras de mapeamento de um processo de negócio especificado em BPMN para diagramas de classes da UML e apresenta exemplos da aplicação dessas regras em um modelo de processo de negócio. O capítulo 6 apresenta o experimento utilizado para validar as regras de mapeamento. Por fim, o capítulo 7 aborda as conclusões do trabalho.

2 FUNDAMENTOS EM MODELAGEM DE PROCESSO DE NEGÓCIO E INTRODUÇÃO À BPMN

Este capítulo apresenta os fundamentos em modelagem de processos de negócio e apresenta uma introdução à BPMN, a qual é a notação padrão usada para a modelagem de processos.

2.1 Conceitos básicos em modelagem de processos de negócio

Um processo de negócio consiste em uma série de atividades relacionadas, realizadas para atingir um determinado objetivo de negócio (WESKE, 2012). Um Processo de negócio também pode ser visto como uma coleção de eventos, atividades e tomadas de decisões inter-relacionadas, que envolvem um conjunto de atores e objetos, os quais coletivamente levam a resultados que agregam valor a pelo menos um cliente (LINDSAY, 2003).

Um modelo de processo de negócio é a representação gráfica desse processo de negócio. O modelo facilita o entendimento do processo e auxilia no compartilhamento do conhecimento do processo com pessoas envolvidas nele. Além disso, o modelo de processo de negócio auxilia na identificação e prevenção de problemas (DUMAS et al., 2013). Um exemplo de processo de negócio textual de solicitação de compra de mercadorias, cujo modelo de processos é apresentado na seção 2.2, pode ser descrito da seguinte forma: (adaptado do seguinte site¹)

O processo tem início com o recebimento pelo setor de compras de uma organização da solicitação de compras de mercadoria; O setor de compras consulta dados do cliente e envia a ordem de compra para o fornecedor, que é responsável pela análise desta ordem de compra. A análise da ordem de compra é realizada da seguinte forma:

1. Se for aprovada, a ordem de compra é processada e a fatura é enviada para o setor de compras. O envio da fatura pode ser realizado tanto via e-mail como via correios.
2. Caso contrário, a ordem de compra é rejeitada e o processo termina.

O setor de compras recebe então a fatura enviada pelo fornecedor e providencia o pagamento da mesma.

¹<http://portal3.tcu.gov.br/portal/page/portal/TCU/comunidades/gestao_processos_trab/curso_mapeamento_processos_trab>

2.1.1 Ciclo de vida da BPM

O ciclo de vida típico, conforme mostrado na figura 2.1, do gerenciamento de processo de negócio inclui as seguintes fases: identificação, descoberta (ou modelagem), análise, redesenho, implementação e monitoramento (DUMAS et al., 2013).

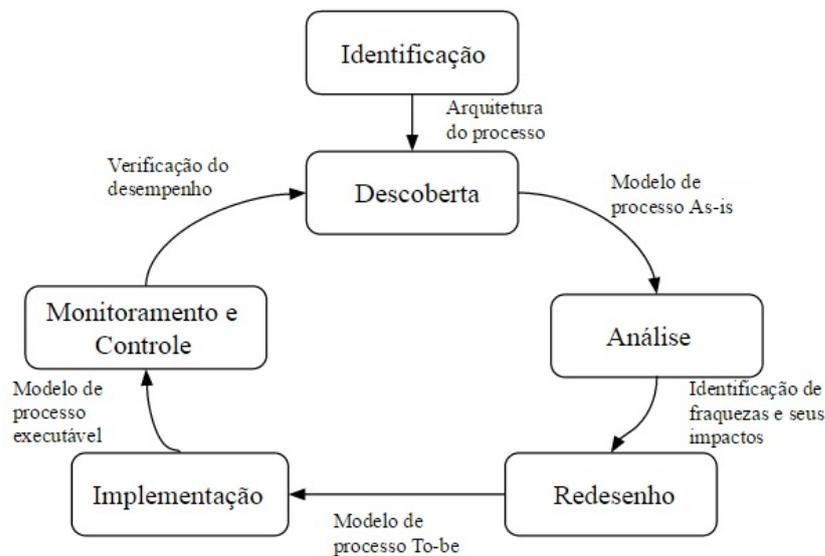


Figura 2.1 – Ciclo de vida BPM (DUMAS et al., 2013)

Identificação: Nesta fase, um problema de negócio é apresentado e procedimentos relacionados a este problema são identificados, delimitados e relacionados. O resultado desta etapa é uma nova arquitetura de processo que provê uma visão geral do processo de negócio em uma organização.

Descoberta (Modelagem): Esta fase visa descobrir e documentar os processos de negócio executados na organização. Nesta etapa, cada processo de negócio identificado é documentado em um formato de um ou mais modelos de processo. O resultado desta etapa é o modelo de processos de negócio que representa a situação atual do processo de negócio organizacional (ou seja, o processo AS-IS).

Análise: Nesta fase, questões relacionadas ao processo AS-IS são identificadas, documentadas e quantificadas através de métricas de medição de desempenho. O resultado desta etapa é uma coleção estruturada de questões (priorizadas em termos de impactos e esforço para resolvê-los).

Redesenho: Esta fase visa identificar alterações no processo que possibilitem a organização atingir seus objetivos de negócio. O resultado final desta fase é a definição e documentação da situação futura do processo de negócio (ou seja, o processo TO-BE).

Implementação: Nesta fase, ocorre o desenvolvimento e a execução de sistemas de TI

(Tecnologia da Informação) que suportam o modelo TO-BE.

Monitoramento: Nesta fase, após o processo redesenhado ser iniciado, dados relevantes são coletados e analisados para verificar o desempenho do processo com relação a suas métricas de medição de desempenho e objetivos.

Considerando o ciclo de vida de BPM, este trabalho apresenta contribuições principalmente para a fase de implementação, pois é nesta etapa que ocorre o desenvolvimento e a execução de sistemas de TI que suportam o modelo To-be. Pode ser necessário extrair diagramas de classes a partir do modelo de processo To-be.

2.1.2 Sistemas de Gerenciamento de *Workflow*

Workflow é a automação de uma parte ou de todo um processo de negócio. Documentos, informações ou atividades são passados de um participante ao outro, afim de que sejam tomadas ações de acordo com um conjunto de regras e procedimentos (COALITION, 1996).

WfMS (Sistema de Gerenciamento de Workflow) é um sistema que possibilita a automatização de modelos de processos de negócio. Um WfMS auxilia na definição, na criação e no gerenciamento da execução do processo de negócio (WESKE, 2012). Exemplos de WfMS são: *Workflow Management (WFM)*, *Enterprise Resource Planning (ERP)*, etc.

Um modelo de processo de negócio deve ser modelado corretamente para evitar a ocorrência de erros durante a sua execução. Problemas como *deadlock*, *livelock* podem causar prejuízos enormes para as empresas.

Workflow e BPM são dois conceitos diferentes. Workflow é apenas a automação de um processo de negócio enquanto o BPM não somente executa, mas também gerência o processo de negócio.

Em (COALITION, 1996) foram apresentados os principais conceitos relacionados ao *workflow* que também são encontrados na BPM. Exemplos são: atividade (automática ou manual), subprocesso, instâncias de processo e instâncias de atividades, controle de fluxo (desvio em paralelo, desvio inclusivo, desvio exclusivo e fluxo de sequencia), participante, item de trabalho e lista de trabalho. A próxima seção apresentará definições de cada um desses conceitos.

2.2 Notação para a modelagem de processos de negócio (BPMN)

Um modelo de processo de negócio é a representação gráfica de um processo de negócio. O modelo facilita o entendimento do processo e auxilia no compartilhamento do conhecimento do processo entre seus participantes. Além disso, o modelo de processo de negócio auxilia no entendimento de problemas atuais da organização e auxilia na identificação de possibilidades de melhoria. O processo de negócio de solicitação de compra de mercadorias descrito anteriormente pode ser representado através do modelo de processos de negócio apresentado na figura 2.2.

Projetistas do OMG (Object Management Group) analisaram diversas notações existentes e buscaram consolidar as melhores ideias destas notações divergentes em uma única notação padrão, BPMN (SPECIFICATION, 2006). Exemplos das notações analisadas são: diagrama de atividades da UML, *UML EDOC Business Processes*, *IDEF*, *ebXML BPSS*, *Activity-Decision Flow (ADF) Diagram*, *RosettaNet*, *LOVeM*, e *Event-Process Chains (EPCs)*.

BPMN tem como principal objetivo fornecer uma notação gráfica de fácil entendimento por parte dos envolvidos no processo incluindo os analistas que capturam as primeiras definições do processo, os desenvolvedores responsáveis por sua automação e, finalmente, aqueles responsáveis por gerenciá-los e monitorá-los (SPECIFICATION, 2006).

BPMN é uma notação com grande poder de expressão e que está sendo utilizada mundialmente por diversas organizações como ferramenta de apoio a documentação, simulação de processos, etc. (RECKER, 2010).

Os elementos gráficos da BPMN usados para modelar processos são divididos nas seguintes categorias (SPECIFICATION, 2006):

- Objetos de fluxo
- Dados
- Objetos de conexão
- Partições
- Artefatos

Este trabalho aborda apenas os objetos de fluxo e partições. Demais categorias de elementos serão abordados em trabalhos futuros.

Obs: Todas as figuras representando os elementos notacionais da BPMN foram obtidas na especificação (SPECIFICATION, 2006).

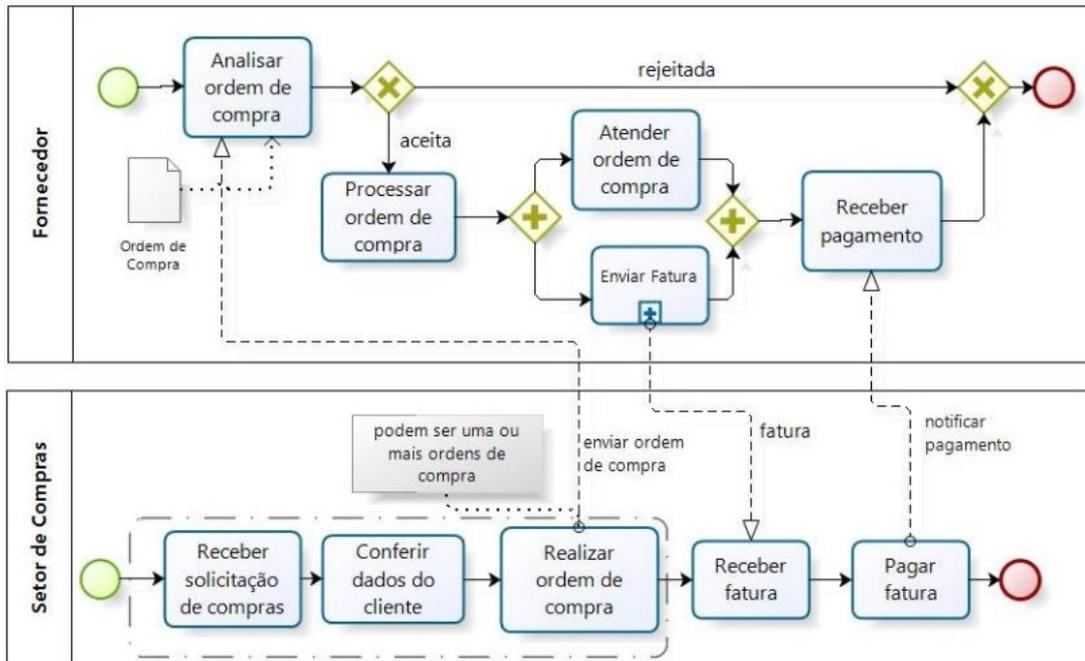


Figura 2.2 – Modelo de processos de negócio em BPMN para solicitação de compras

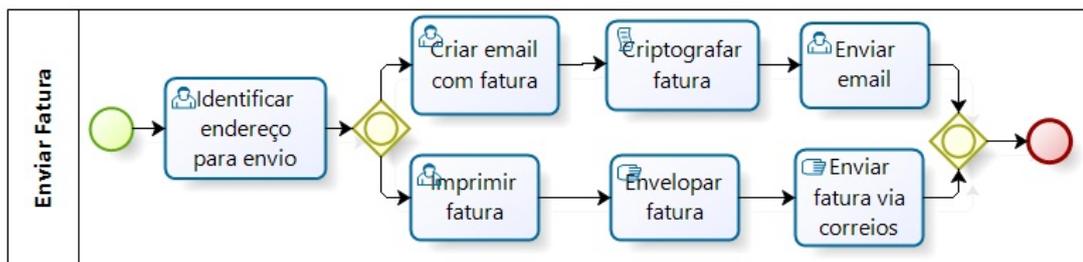


Figura 2.3 – Subprocesso “Enviar Fatura” detalhado

2.2.1 Objetos de fluxo

Objetos de fluxo constituem os principais elementos usados para descrever o comportamento de um processo de negócio. Existem três tipos de objetos de fluxo: eventos, atividades e desvios. As definições aqui apresentadas têm como base a especificação da BPMN (SPECIFICATION, 2006).

Evento: representa algo que ocorre durante a execução de um processo de negócio. Existem três principais tipos de eventos: evento de início, evento intermediário e evento de fim.

Evento de início indica onde um dado processo iniciará. Podem ser de diferentes tipos de acordo com a causa do mesmo. Existem evento de início sem causa específica (Figura 2.4), evento de início do tipo mensagem, evento de início do tipo tempo, evento de início do tipo condicional, evento de início do tipo sinal, evento de início do tipo múltiplo e evento de início do tipo múltiplo paralelo. Neste trabalho serão abordados apenas evento de início temporal e

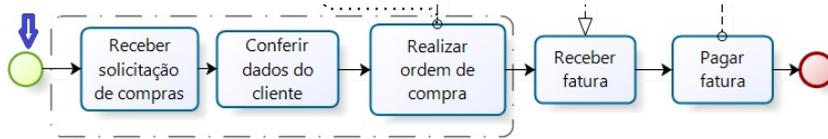


Figura 2.4 – Representação da BPMN para evento de início sem causa definida

evento de início condicional. Evento de início temporal é usado para indicar que o mesmo só inicia quando uma data/hora for atingida. Já o evento de início do tipo condicional possui uma condição de negócio associada ao mesmo. O evento só acontecerá caso a condição de negócio for verdadeira. Demais tipos de eventos de início serão abordados em trabalhos futuros. A figura 2.5 ilustra tipos de eventos na BPMN.

Eventos	Evento de Início			Eventos Intermediários			Evento de Fim
	Eventos de Alto Nível	Evento que ocorre quando interrompido um Sub-Processo	Evento que, quando ocorre, não provoca interrupção de Sub-Processo	Captura	Interrompe a execução da instância em uma divisão ou compartimento	Não interrompe a execução da instância em uma divisão ou compartimento	Lançamento
Simples: Eventos sem tipo indicam pontos de início, de fim e mudanças de estado.							
Mensagem: Recebimento e envio de mensagens.							
Temporal: pontos no tempo, instante no tempo, intervalo de tempo, limite de tempo. Podem ser eventos únicos ou cíclicos.							
Escalável: ativa mudança para um nível mais alto de responsabilidade.							
Condicional: Reação a alterações nas condições de negócio ou a regras de negócio.							
Conector: Conector entre páginas. Dois eventos de conexão equivalem a um fluxo de sequência.							
Erro: Captura ou inserção de erros pre-identificados.							
Cancelamento: reagem ao cancelamento de uma transação ou ativam cancelamento.							
Compensação: Tratamento ou ativação de ação de compensação.							
Sinal: Emitem sinais entre processos. Um mesmo sinal pode ser capturado várias vezes.							
Múltiplo: Ou capturam um dentre um conjunto de eventos, ou lançam um ou mais eventos de qualquer dos tipos definidos.							
Múltiplo Paralelo: capturam, de uma só vez, todos os eventos de um conjunto de eventos que ocorrem em paralelo.							
Final: Ativam a terminação imediata de um processo.							

Figura 2.5 – Representação da BPMN para tipos de eventos (SPECIFICATION, 2006)

Evento intermediário é um evento que ocorre no meio de um processo.

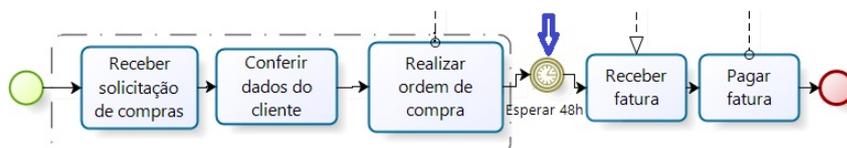


Figura 2.6 – Representação da BPMN para evento intermediário

Evento de fim é usado para indicar quando as instâncias de um processo completam suas atividades.

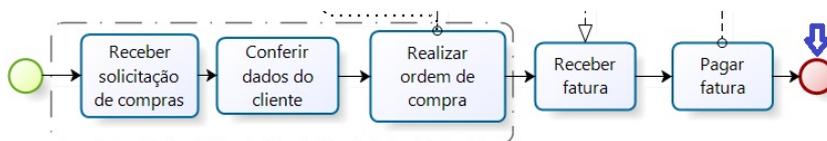


Figura 2.7 – Representação da BPMN para evento de fim

Atividade: é o trabalho realizado dentro de um processo de negócio. Uma atividade pode ser atômica (também chamada tarefa) ou não-atômica (ou seja, subprocesso). Uma atividade do tipo tarefa pode ser manual, automática ou de usuário.

Tarefa manual é uma tarefa que não pode ser automatizada. Por exemplo, a tarefa “Enviar fatura via correios” no processo apresentado na figura 2.3 é um exemplo da representação da BPMN para tarefa manual.

Tarefa automática é uma tarefa que pode ser automatizada por um *WfMS* sem a intervenção de uma entidade externa. Na BPMN existe uma diversidade de tarefas automáticas. Exemplos são: tarefa de serviço, tarefa de envio, tarefa de recebimento etc. A tarefa “Criptografar fatura” no processo apresentado na figura 2.3 é um exemplo da representação da BPMN para tarefa automática do tipo script.

Tarefa de usuário é uma tarefa realizada por um participante de um processo de negócio através de um software. A tarefa “Enviar email” no processo apresentado na figura 2.3 é um exemplo da representação da BPMN para tarefa de usuário.



Figura 2.8 – Representação da BPMN para tipos de tarefas

A figura 2.9 mostra os diferentes tipos de atividades suportados na BPMN. É possível ver que um subprocesso pode ser representado de forma explícita (ou seja, apresentando elementos contidos dentro dele), conforme mostrado na figura 2.9, ou pode ser usada apenas uma atividade única que não mostra os detalhes dos elementos contido dentro dele, conforme apresentado na figura 2.10,

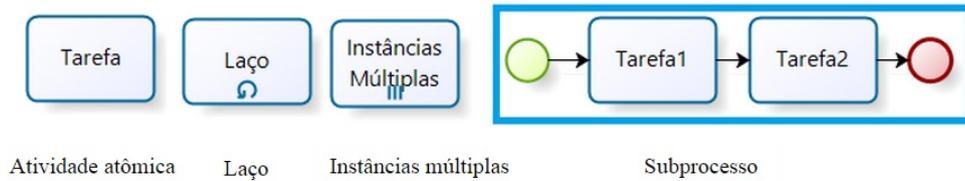


Figura 2.9 – Representação da BPMN para tipos de atividades

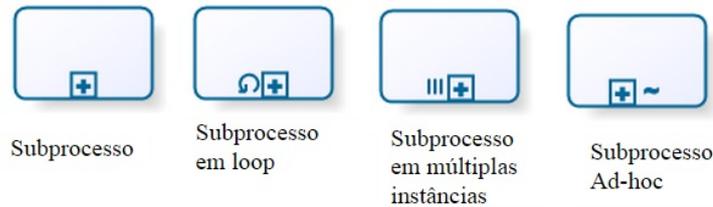


Figura 2.10 – Representação da BPMN para tipos de subprocessos

Desvios: são usados para controlar as convergências e divergências de fluxo de sequência dentro de um processo de negócio. A figura 2.11 apresenta os diferentes tipos de desvios com suas respectivas semânticas.

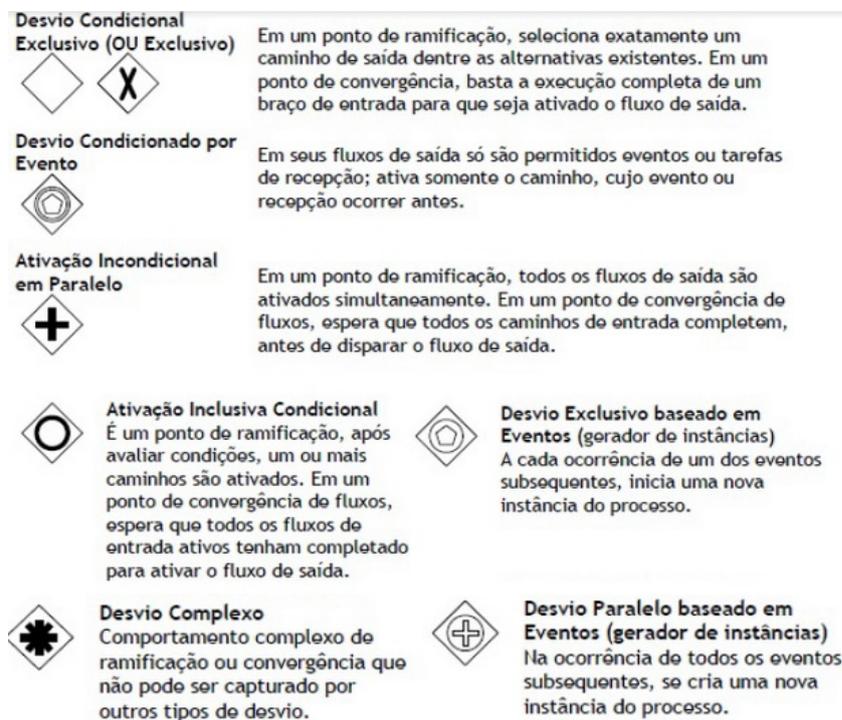


Figura 2.11 – Representação da BPMN para tipos de desvios

A figura 2.2 e figura 2.3 apresentam um exemplo prático da utilização do desvio exclusivo, desvio inclusivo e desvio paralelo nos processos de negócio de solicitação de compra de mercadorias.

2.2.2 Dados

Um dos requisitos tradicionais de uma ferramenta de modelagem de processo de negócio é a capacidade de representar dados em modelos de processos de negócio. Na BPMN, os dados podem ser representados através dos seguintes elementos: objetos de dados, dados de entrada, dados de saída e repositórios de dados. Objeto de dados (Figura 2.12) é usado para mostrar como dados são requeridos ou produzidos por atividades ou processos. Os dados requeridos representam dados de entrada enquanto os dados produzidos representam os dados de saída. O repositório de dados é usado para consultar, recuperar ou atualizar dados dentro de um processo.

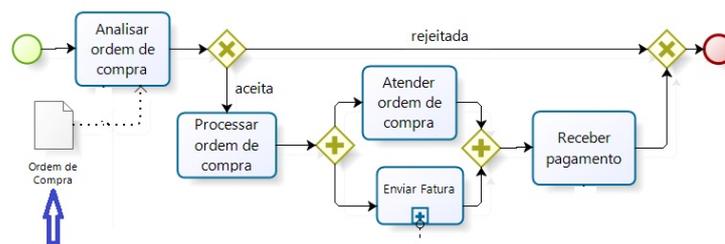


Figura 2.12 – Representação da BPMN para o objeto de dados “Ordem de compra”



Figura 2.13 – Representação da BPMN para objetos de dados, dados de entrada/saída, repositório de dados

2.2.3 Objetos de conexão

Como o nome indica, objetos de conexão são usados para conectar elementos notacionais de um modelo processo de negócio. Os objetos de conexão incluem fluxo de sequência, fluxo de mensagem e associações.

Fluxo de sequência: apresenta a ordem em que as atividades são executadas dentro de um processo de negócio. A figura 2.14 apresenta um exemplo do uso do fluxo de sequência.

Fluxo de mensagem: é usado para a troca de mensagens entre dois participantes de um processo de negócio. Fluxo de mensagem só pode ser usado quando existir pelo menos duas piscinas. O fluxo de mensagem pode ser conectado à piscina, à atividade ou aos eventos de mensagem.

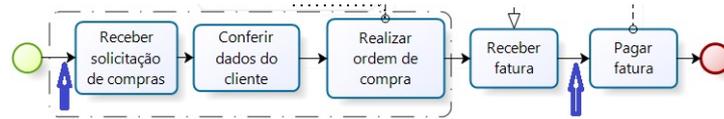


Figura 2.14 – Representação da BPMN para fluxo de sequencia

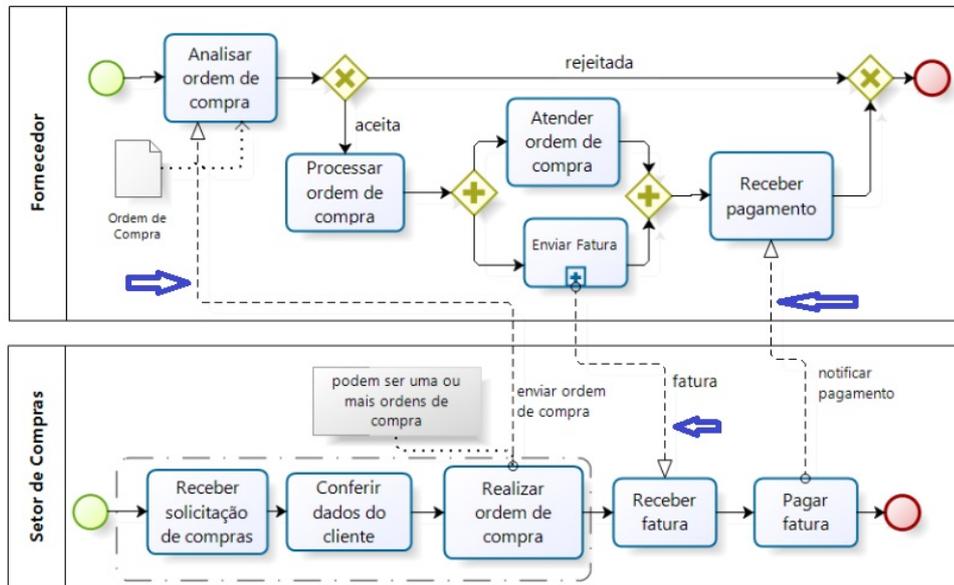


Figura 2.15 – Representação da BPMN para fluxo de mensagem

Associação: relaciona informações e artefatos com elementos notacionais da BPMN.

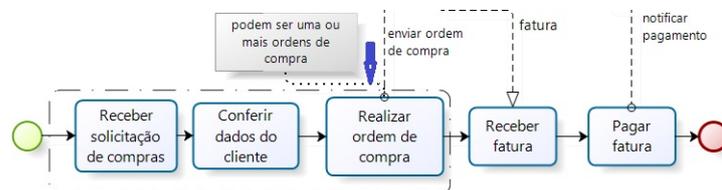


Figura 2.16 – Representação da BPMN para associação

2.2.4 Partições

Partições (ou divisões) representam entidades responsáveis pela realização de atividades. Na BPMN, as partições estão divididas em Piscina e Raia.

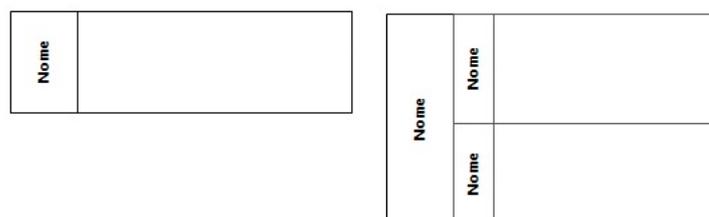


Figura 2.17 – Representação da BPMN para piscina e raias

Piscina: representa entidades (Organizações, Cliente, Fornecedor etc.) responsáveis pela execução de uma parte do processo ou um conjunto de atividades. Uma piscina pode ser vazia (ou seja, caixa preta) ou pode conter um processo dentro dela.

A figura 2.2 apresenta a representação da piscina “Fornecedor” no processo de negócio de solicitação de compras.

Raias: frequentemente representam papéis dentro de uma piscina (representando, por exemplo, departamentos de uma organização). Não pode existir raia fora de piscina, ou seja, caso a piscina deixar de existir as raias também desaparecerão do modelo de processo. A figura 2.18 mostra a representação de piscina Empresa subdividida em duas raias: Setor Vendas e Setor Compras.



Figura 2.18 – Exemplo de piscina Empresa com suas respectivas raias: Setor Vendas e Setor Compras

2.2.5 Artefatos

Os artefatos são usados para documentar o processo de negócio sem interferir no fluxo normal do processo. Os artefatos incluem grupos e anotações.

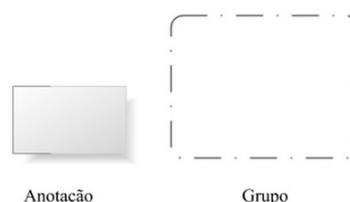


Figura 2.19 – Exemplo de anotação e grupo

O grupo é usado para agrupar elementos de um processo sem afetar a execução do processo. BPMN não define uma semântica associada a esse elemento, sendo usado apenas para fins de visualização no diagrama. Por exemplo, na figura 2.2 o grupo é usado para agrupar as atividades prévias ao estudo de crédito do cliente (receber solicitação de compras, conferir dados do cliente, realizar ordem de compras). Anotações são usadas para comentar ou dar informações adicionais que podem ser importantes para o leitor do modelo de processo.

3 MODELAGEM DE SISTEMAS ATRAVÉS DO DIAGRAMA DE CLASSE DA UML E INTRODUÇÃO A *MODEL-DRIVEN ARCHITECTURE*

Este capítulo apresenta os conceitos básicos sobre diagrama de classes da UML e sobre MDA (*Model-Driven Architecture*).

Diagramas de classes da UML são encontrados com maior frequência na modelagem de sistemas orientados a objetos. Eles são usados para modelar a visão estática do projeto de um sistema. Na maioria dos casos, isso envolve a modelagem do vocabulário do sistema, a modelagem de colaborações e/ou a modelagem de esquemas (BOOCH, 1999).

Além disso, diagramas de classes oferecem suporte para os requisitos funcionais (os serviços/funcionalidades que o sistema deverá fornecer aos usuários finais) de um sistema. Através do diagrama de classes é possível obter uma visão lógica de um esquema de banco de dados (BOOCH, 1999).

Diagramas de classes podem ser considerados uma notação alternativa aos diagramas Entidade-Relacionamento (ER) (NAVATHE, 2005). Onde as classes representariam as tabelas e os relacionamentos entre as tabelas são equivalentes aos relacionamentos entre as classes. O diagrama de classes inclui: classes, interfaces e relacionamento de dependência, de generalização e de associação (FORCE, 1999).

As definições aqui apresentadas sobre diagramas de classes têm como fonte a especificação da UML (FORCE, 1999).

3.1 Classes

Uma classe é definida como um descritor de um conjunto de objetos com propriedades em comum em termos de estrutura, comportamento e relacionamento. Objetos de uma classe compartilham atributos, relacionamentos e semânticas.

Atributo representa a propriedade de um item que está sendo modelado e compartilhado por todos os objetos dessa classe. Por exemplo, todo funcionário de uma empresa possui um nome, um cpf, um endereço. Assim, a classe Funcionário terá como atributos: nome, cpf, endereço etc.

Operação é uma abstração de algo que pode ser realizado com um objeto e que é compartilhado por todos os objetos dessa classe. Por exemplo, um funcionário pode calcular salário, cadastrar clientes etc. A figura 3.1 apresenta um exemplo da representação de uma classe com

seus atributos e suas operações.

nome	Funcionário
atributos	Nome: String Cpf: Integer Endereço: String
operações	CalcularSalário(); CadastrarCliente();

Figura 3.1 – Exemplo de classe

Responsabilidade é um contrato ou obrigação de uma determinada classe. Ao criar uma classe, se cria uma declaração de que todos os objetos dessa classe têm mesmo estado e mesmo tipo de comportamento. Por exemplo, um objeto da classe Vendedor pode ser responsável por cadastrar pedido do cliente e determinar riscos de um pedido do cliente. Ao modelar classes, um bom ponto de partida é especificar as responsabilidades dos itens existentes em seu vocabulário. A medida que os modelos são aprimorados, as responsabilidades podem ser traduzidas em atributos e operações, capazes de atender as responsabilidades da classe (BOOCH, 1999).

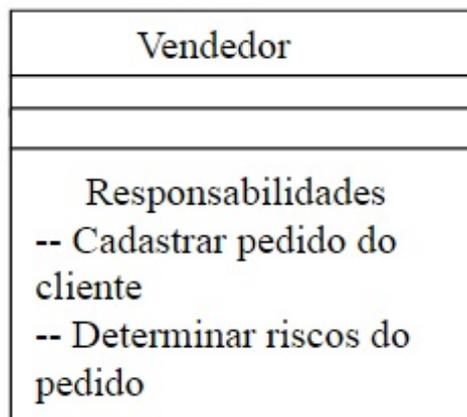


Figura 3.2 – Exemplo de responsabilidades de classe

3.2 Relacionamentos

Ao fazer uma modelagem de um sistema, é necessário mostrar como as classes se relacionam entre si. Os relacionamentos entre classes incluem:

- Associação
- Generalização
- Dependência

Associação: descreve as conexões específicas entre objetos ou outras instâncias em um sistema. Através da associação, é possível navegar do objeto de uma classe até o objeto de outra classe e vice-versa. O tipo mais comum das associações é a associação binária entre duas classes. Existem também associações enésimas que são usadas para conectar mais de duas classes. Uma associação pode possuir um nome, um papel, uma multiplicidade ou uma agregação. O nome pode ser usado para descrever a natureza de uma associação. Para não ter ambiguidade pode-se atribuir uma direção para o nome da associação para orientar como o nome deverá ser lido, conforme mostrado na figura 3.3.

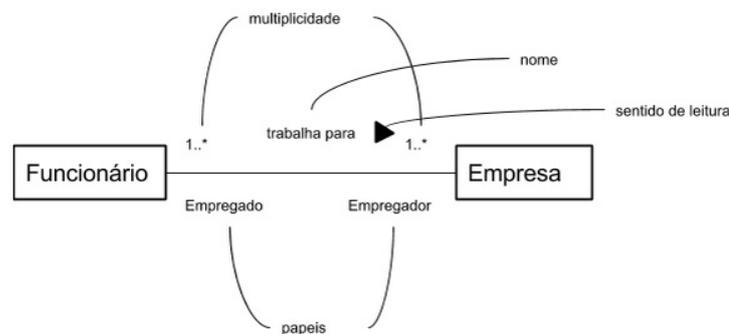


Figura 3.3 – Exemplo de associação entre as classes (BOOCH, 1999)

Papel é a face que a classe próxima a uma das extremidades apresenta à classe encontrada na outra extremidade da associação. Por exemplo, um funcionário desempenha o papel de empregado enquanto a empresa desempenha o papel de empregador.

Multiplicidade representa um intervalo de inteiros que especifica o tamanho possível do conjunto de objetos relacionados. Em outras palavras, a multiplicidade é a quantidade de objetos que podem ser conectados pela instância de uma associação. Pode ser representada uma multiplicidade de exatamente um (1); zero ou um (0..1); zero ou mais (0..*); um ou mais (1..*).

Agregação é usada para fazer a modelagem de um relacionamento “todo/parte”, no qual uma classe representa um item maior (ou seja, a classe “todo”), formado por itens menores (ou seja, as classes “partes”). Em outras palavras, a agregação é usada para mostrar que um objeto todo contém os objetos das partes. A figura 3.4 mostra um exemplo de um relacionamento de agregação entre um departamento e um funcionário dentro da empresa. Onde, a classe Departamento representa a classe “todo” enquanto a classe Funcionário representa a classe “parte”.

Composição é uma variante semanticamente mais forte do que a agregação. Neste caso, os objetos “partes” só podem pertencer a um único objeto “todo” e têm o seu tempo de vida coincidente com o dele. Ou seja, se o objeto “todo” deixar de existir, os objetos “partes” relaci-

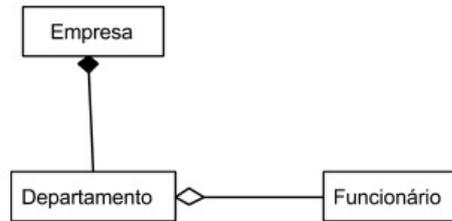


Figura 3.4 – Exemplo de agregação e composição

onados a ele também desaparecerão. A figura 3.4 mostra um exemplo de um relacionamento de composição entre uma empresa e o departamento dentro da empresa. Onde, a classe Empresa representa a classe “todo” enquanto a classe Departamento representa a classe "parte".

Generalização: é um relacionamento entre itens gerais (ou seja, superclasses ou classes mãe) e tipos mais específicos desses itens (ou seja, subclasses ou classes filha). A generalização significa que os objetos da classe filha podem ser utilizados em qualquer local em que a classe mãe ocorra, mas não vice-versa. A classe filha herda as propriedades de uma classe mãe, principalmente seus atributos e operações. Uma generalização é representada graficamente como uma linha sólida com uma seta triangular apontando para a classe mãe, conforme ilustrado na figura 3.5.

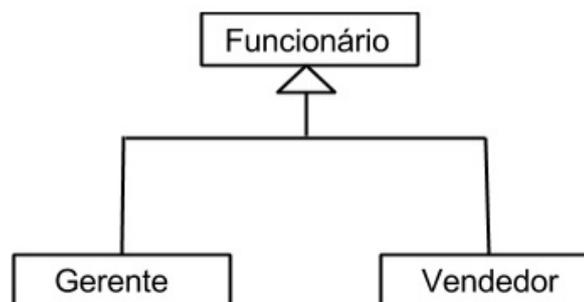


Figura 3.5 – Exemplo de generalização

Dependência: é um relacionamento usado para mostrar a ideia de que um item (por exemplo, objeto da classe Cliente) usa as informações e serviços de outro item (por exemplo, objeto da classe Fornecedor). A dependência é representada graficamente como linhas tracejadas, conforme mostrado na figura 3.6, apontando o item do qual o outro depende.

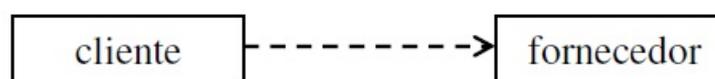


Figura 3.6 – Dependência

3.3 Model-Driven Architecture e transformação entre modelos

MDA (Model-Driven Architecture) é um *framework* de desenvolvimento de software criado pelo OMG, baseado na UML e outros padrões da indústria. MDA determina como um software pode ser desenvolvido colocando a modelagem no 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 (MILLER et al., 2003).

A abordagem MDA divide os modelos em três níveis de abstração (MILLER et al., 2003):

- *Computation Independent Model* (CIM)
- *Platform Independent Model* (PIM)
- *Platform Specific Model* (PSM).

No nível CIM não são mostradas informações estruturais dos sistemas. Nesse nível estão contidos os modelos de processo de negócio. Esses modelos têm como objetivo a representação genérica do processo, facilitando o seu entendimento e auxiliando no compartilhamento do conhecimento do mesmo com pessoas envolvidas nele, sem levar em consideração restrições de tecnologia. Ou seja, nível CIM independe de plataforma.

Modelos de sistema representados através da UML com o uso de diagramas de classes 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. Para transformar um modelo CIM para um modelo PIM é necessário identificar quais partes do CIM são suportadas por um sistema.

PSM é a visão de um sistema em relação a uma plataforma específica. Diagramas de seqüência da UML constituem exemplos de modelos no nível PSM, pois contém detalhes específicos da plataforma de implementação utilizada.

A transformação entre modelos é o processo de conversão de um modelo em outro modelo do mesmo sistema (GROUP et al., 2003). MDA apresenta a transformação dos modelos PIM para modelos PSM. A transformação é feita através da definição de regras de mapeamento entre modelos. As regras de mapeamento são definidas para elementos dos metamodelos de cada um dos modelos criados.

Como MDA apresenta o mapeamento a partir do nível PIM (ou seja, o nível que contém o modelo de sistema), a motivação deste trabalho é completar a MDA propondo um conjunto de regras de mapeamento de um modelo em nível CIM para um modelo em nível PIM.

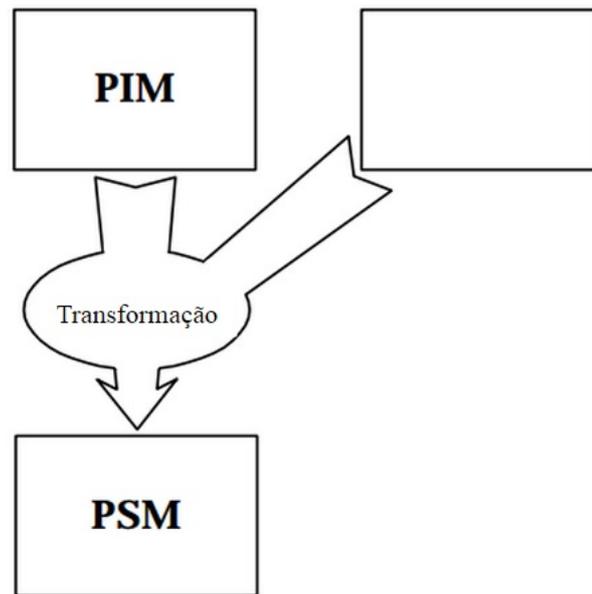


Figura 3.7 – Tranformação entre modelos (GROUP et al., 2003)

A transformação de um modelo em um nível CIM para um modelo em nível PIM é objeto de diversos estudos na área de Engenharia de Software. Diversos trabalhos encontrados na literatura abordam essa questão com base na abordagem proposta pela MDA.

4 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados alguns trabalhos relacionados com a transformação de modelos em nível de processo de negócio para modelos em nível de sistema. Foi realizado um estudo de junho de 2014 até o início de 2015. Bibliotecas eletrônicas como IEEEExplore, ACM e as bibliotecas das editoras de revistas técnicas, como a Elsevier, Springer foram as principais fontes de pesquisa.

Analisando os trabalhos obtidos na literatura, observa-se que existem diversas propostas sugeridas para resolver o problema de mapeamento de modelos em nível de processo de negócio para modelos em nível de sistema. Atualmente, não existe um método de mapeamento que seja considerado como padrão. Porém diversos trabalhos encontrados na literatura se baseiam na abordagem MDA para realizar suas transformações.

Na maioria dos trabalhos relacionados estudados os processos de negócios foram modelados através de notações diferentes da BPMN. Em (SUAREZ, 2008; BARROS, 2000; MOLINA et al., 2000) foram usados diagramas de atividade da UML para modelar os processos de negócio. Em (MOLINA et al., 2000) foi feito inicialmente a transformação de um diagrama de atividades para um diagrama de casos de uso. O diagrama de casos de uso obtido foi transformado para um diagrama classes. Em (OKAWA et al., 2007) foi usada uma notação genérica de modelagem de processo de negócio.

Em (RUNGWORAWUT, 2005) foi apresentado o mapeamento de modelos de processos de negócio em BPMN para diagrama de classes. O autor apresenta apenas um método para obter nomes de classes a partir de rótulos de atividades. Não foi explicado claramente como foram obtidos relacionamentos entre as classes. Além disso, não foram apresentados possíveis mapeamentos de elementos notacionais da BPMN tais como divisores e objetos de fluxo como atividades, eventos, desvios etc.

Em (RODRÍGUEZ et al., 2010) é apresentado uma transformação de modelos de processos de negócio em BPMN para diagramas de classes e diagramas casos de uso usando um conjunto de regras bem definidas. A transformação é feita em duas etapas. O modelo BPMN é transformado inicialmente em um diagrama de atividades. Piscinas e raias foram transformadas em partições de atividades no diagrama de atividades; Tarefas foram mapeadas para ações e os fluxos de dados foram mapeados para repositórios de dados. Em seguida, o diagrama de atividades obtido é transformado em um diagrama de classes e diagrama de casos de uso. Partição e repositório de dados do diagrama de atividades foram mapeadas para classes no diagrama de classes. Caso existir subpartições no diagrama de atividades, essas também foram mapeadas

para classes que se relacionam com a classe obtida a partir da partição através de um relacionamento de agregação. As ações dentro de uma partição foram mapeadas para operações dentro das classes.

As próximas seções apresentam com mais detalhes alguns dos trabalhos citados anteriormente.

4.1 A method of linking business process modeling with information system desing using UML and its evaluation by prototyping (OKAWA et al., 2007)

Neste trabalho, o autor compara os métodos usados para obter diagramas de classes a partir de um diagrama de casos de uso e da descrição textual de um processo de negócio. O autor divide o processo de mapeamento em quatro estágios:

1. Análise de processo de negócio atual e a criação de modelos de processos AS-IS que representam esses processos.
2. Simulação de cada processo de negócio.
3. Melhoria do modelo de processo AS-IS para o modelo To-Be.
4. Transformação de cada modelo de processo para um diagrama de classes da UML correspondente.

A transformação é feita seguindo as seguintes políticas:

1. Associar cada modelo de processo de negócio a um único diagrama de classes. Dessa forma, mudanças feitas em um modelo de processo de negócio afetarão apenas um 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 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.

A figura 4.1 mostra um exemplo do mapeamento usando a política descrita acima.

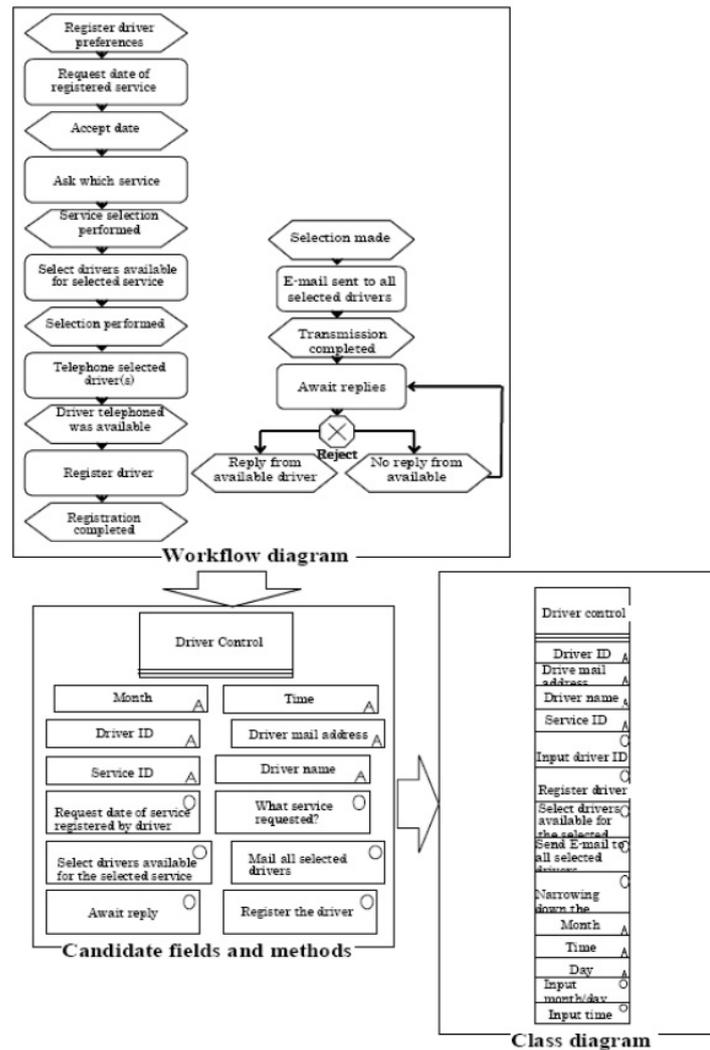


Figura 4.1 – Diagrama de classe a partir de um diagrama de workflow (OKAWA et al., 2007)

4.2 Transformation of a process business model to domain model (SUAREZ, 2008)

Este trabalho apresenta uma metodologia para realizar transformação de modelos de processos de negócio modelados através do diagrama de atividades, conforme mostrado na figura 4.2, para diagrama de classes da UML. A transformação consiste nas seguintes etapas:

1. Mapear cada elemento do diagrama de atividades para elementos do diagrama de classes de acordo com a regra de mapeamento proposta pelo autor.
2. Refinar o diagrama de classes obtido na etapa 1.

Na etapa 1 cada raia do diagrama de atividades é transformada em uma única classe. Os objetos de negócios instanciados várias vezes, mas que diferem em seus estados também são considerados como classes. Por exemplo, se um pedido possui diversos estados, tais como: em análise, aprovado, reprovado etc. Então, o pedido é candidato a classe.

A etapa 2 consiste nos seguintes passos:

- Comparar os atributos em comum entre classes que têm uma relação de associação em comum.
- Se existir apenas um atributo em comum entre duas ou mais classes relacionadas, representar esse atributo como uma única classe e relacioná-lo por meio de uma associação de generalização com as classes que a possuem. Se o número de atributos em comum for maior que um, criar uma nova classe que contém esses atributos em comum.
- Para as classes que resultam da transformação de raias, usar o mesmo critério usado para classes que têm atributos em comum. Pois as raias podem ter atributos em comum em um mesmo diagrama de classes.

A figura 4.3 mostra o diagrama de classes obtido, após aplicação das regras sugeridas acima, a partir do modelo de processo apresentado na figura 4.2.

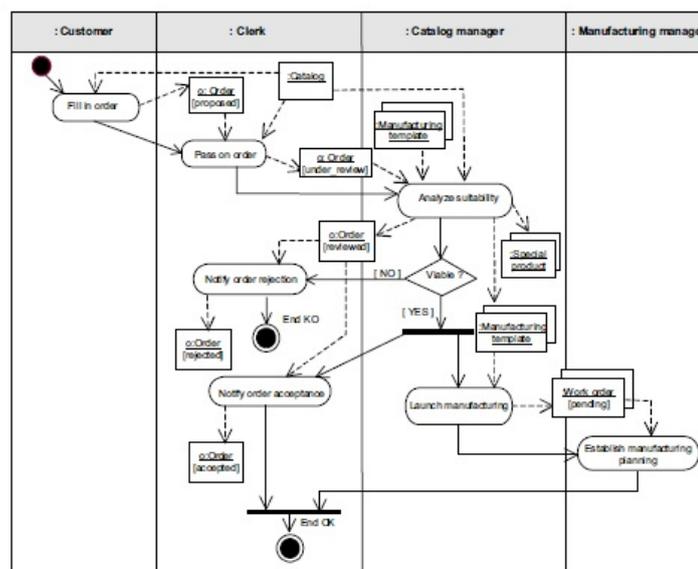


Figura 4.2 – Processo de negócio modelado através do diagrama de atividades (SUAREZ, 2008)

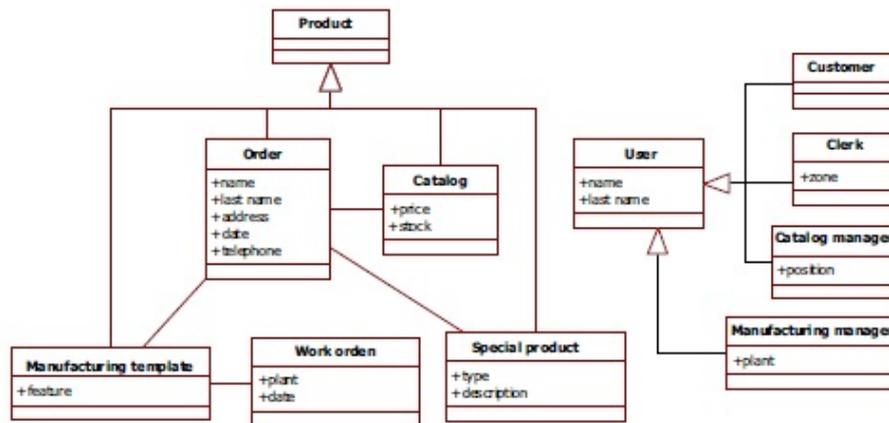


Figura 4.3 – Diagrama de classes correspondente ao modelo da figura 4.2 (SUAREZ, 2008)

4.3 Considerações sobre os trabalhos relacionados

Em (OKAWA et al., 2007) um dos problemas é a notação utilizada para modelar processos de negócio. A transformação de um modelo em nível de processos de negócio para um modelo em nível de sistema foi feita a partir de um modelo de processos modelado com uma notação genérica de modelagem de processos de negócio. Além disso, o autor apenas apresentou um diagrama de classes para fazer a modelagem do vocabulário de sistema. Ou seja, apenas foram apresentadas classes com suas propriedades (atributos, métodos) sem se preocupar com os relacionamentos entre as classes.

Em (SUAREZ, 2008) foi realizada uma transformação de um processo de negócio modelado através de um diagrama de atividades para um diagrama de classes. Nesse trabalho, o autor apenas apresentou um diagrama de classes sem explicar no texto como foram obtidos os relacionamentos de associação entre as classes. Além disso, o mapeamento de eventos e desvios não foi considerado pelo autor.

Em (RODRÍGUEZ et al., 2010) o autor apresenta a transformação de modelos de processos de negócio em BPMN para um diagramas de classes e um diagramas casos de uso. A transformação é feita em duas etapas. O modelo de processo de negócio em BPMN é transformado inicialmente em um diagrama de atividades. Em seguida o diagrama de atividades é transformado em um diagrama de classes e diagrama de casos de uso. O autor não deixa claro por que foi realizado inicialmente o mapeamento para um diagrama de atividades. A BPMN é uma notação que tem um poder de expressão maior do que um simples diagrama de atividades. Assim, acredita-se que é possível realizar um mapeamento direto de um modelo de processo de negócio em BPMN para diagrama de classes sem precisar passar por um diagrama de atividades. Os objetos de fluxo como desvio exclusivo, eventos temporais não têm mapeamento

específico neste trabalho. O autor também não apresenta um método para obter candidatos a classes a partir de rótulos de atividades. Uma contribuição desse trabalho está na formalização das transformações onde o autor usa definições formais das transformações realizadas usando uma linguagem entre modelos, *Query View Transformation* (QVT¹) proposto pela MDA. A formalização permite a automatização das transformações.

Em (RUNGWORAWUT, 2005) o autor apresenta a transformação de um modelo de processo de negócio em BPMN para um diagrama de classes. Esse trabalho não apresenta um conjunto de regras de mapeamento de elementos notacionais da BPMN para elementos notacionais do diagrama de classes. Foi apresentado apenas um método para obtenção de candidatos a classes a partir de rótulos das atividades “*Noun Phrase Identification*”. Em nenhum momento foram citados elementos notacionais como piscina, raias que também poderiam ser candidatos a classes. Os desvios e eventos também não foram tratados. O autor aplicou uma estratégia chamada “*Concept Category Strategy*” para obter outros conceitos importantes que poderiam ser candidatos a classes. Esta estratégia envolve a participação ativa de um analista de sistema para decidir qual dos candidatos a classes obtidos serão representados dentro de um diagrama de classes.

O fato de não haver uma proposta consolidada para realizar o mapeamento de modelos de processos de negócio em BPMN para diagramas de classes indica que ainda existem evoluções que podem ser feitas nesse assunto, dentre as quais se destacam:

- Diferentes possibilidades de transformação do desvio exclusivo (XOR);
- Diferentes possibilidades de relacionamentos entre classes obtidas a partir de piscinas e raias;
- Possibilidade de relacionamento entre classes obtidas a partir de piscinas (ou raia) e as obtidas a partir de rótulos de atividades;
- Possibilidade de mapeamento de eventos do tipo temporal;

A proposta apresentada no próximo capítulo apresenta um método que atenda as evoluções mencionadas acima.

¹<<http://www.omg.org/spec/QVT/1.2/>>

5 MAPEAMENTO DE MODELOS DE PROCESSO DE NEGÓCIO PARA DIAGRAMAS DE CLASSES

O objetivo deste capítulo é apresentar um conjunto de regras de mapeamento que possibilitam a obtenção de informações para um diagrama de classes da UML a partir de um modelo de processo de negócio.

O mapeamento entre os modelos foi realizado com base na especificação da BPMN (SPECIFICATION, 2006) e de UML (FORCE, 1999). Cada elemento notacional da BPMN, conforme apresentado na tabela 5.1, foi analisado para identificar um possível mapeamento com o respectivo elemento do diagrama de classes.

Objetos de fluxo (eventos temporais, atividades do tipo tarefa e desvios exclusivos), as partições (piscinas e raias) foram considerados neste trabalho. Já os objetos de conexão (fluxo de sequência, fluxo de mensagem e associação), demais tipos de atividades, eventos e desvios não foram considerados neste trabalho, mas podem ser assunto para trabalhos futuros.

Tabela 5.1 – Elementos notacionais da BPMN considerados neste trabalho

Elementos BPMN	Descrição
Piscina	Usada para representar diferentes participantes (Ex: Organizações, Clientes, Fornecedores etc.) de processos de negócio.
Raia	Usada para particionar uma piscina, permitindo representar diferentes departamentos dentro de uma organização.
Atividade	Usada para representar o trabalho realizado dentro de um processo (neste trabalho, apenas atividade do tipo tarefa é considerada).
Desvio	Usado para controlar as divergências e as convergências no fluxo de sequência de um processo (nesse trabalho, apenas o desvio exclusivo, ou seja, XOR, é considerado).
Evento	Usado para representar algo que ocorre no início, durante ou no fim de um processo (neste trabalho, apenas o evento temporal e condicional são considerados).

Para alguns casos, foram identificados mais de uma possibilidade de mapeamento. Para obter melhores resultados, o processo deve ter sido modelado corretamente e sem ambiguidades. Um processo mal modelado pode levar a construção de um diagrama de classes que não reflete a visão correta do projeto de sistema. Contudo, para o conjunto de regras de mapeamento proposto neste trabalho, o modelo de processo foi assumido como correto.

Apenas a especificação gráfica dos modelos de processos é usada pelo método, ou seja, a respectiva descrição textual não é usada. Conseqüentemente, não objetiva-se gerar diagrama de classes completo, mas apenas um fragmento do mesmo, incluindo: nomes de classes; responsabilidades de classes; e sugestões de relacionamentos entre as classes.

Para cada modelo de processo, um único diagrama de classes é criado; Conseqüente-

mente, atividades compostas (ou seja, subprocessos) são mapeadas separadamente, gerando outros diagramas de classes.

5.1 Verificação de um modelo de processo de negócio

Um modelo de processo de negócio deve ser modelado corretamente para obtenção de melhores resultados em seu mapeamento para o diagrama de classes. Essa correteude deve ser garantida pelo projetista do modelo de processo.

Porém, a especificação da BPMN (SPECIFICATION, 2006) não define, por exemplo, regras de como definir rótulos de atividades em um processo. Atualmente, existem estudos com objetivo de contribuir com boas práticas de nomenclatura das atividades de processos.

Mendling et al.(2010), por exemplo, realizaram testes com mais de 600 modelos de processos e evidenciam que o rótulo na forma “verbo (na forma imperativa) + objeto” é mais adequado na nomenclatura das atividades. Por exemplo, em atividades como “Aprovar documento”, “Emitir ordem de serviço”, os objetos são o “documento” e a “ordem de serviço”, respectivamente. Esse formato diminui possíveis ambiguidades no entendimento das atividades.

Visando obter melhores resultados, as regras de mapeamento apresentadas neste trabalho assumem que os modelos de processo são construídos seguindo esse formato verbo (imperativo) + objeto.

5.1.1 Mapeamento de piscina

Piscinas representam entidades (cliente, fornecedores, organizações, vendedor etc.) responsáveis pela execução de um conjunto de atividades (SPECIFICATION, 2006).

Esse conceito é análogo ao de uma classe no contexto de diagrama de classes. Ambas, piscina e classe, servem para representar entidades (ou “objetos”) que compartilham características e responsabilidades. Assim, neste trabalho uma piscina é considerada como candidata a classe, como representado no exemplo da figura 5.1.

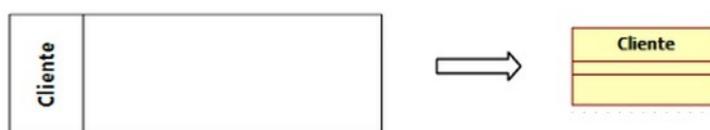


Figura 5.1 – Exemplo de mapeamento de piscina para classe

5.1.2 Mapeamento de raia

Raias frequentemente representam papéis dentro de uma piscina (representando, por exemplo, departamentos de uma organização) (SPECIFICATION, 2006).

Esse conceito também é análogo ao de uma classe. Ambas, raia e classe, assim como no caso anterior referente a piscina, servem para representar entidades (ou “objetos”) que compartilham características e responsabilidades. Assim, neste trabalho, uma raia também é considerada como candidata a classe, como representado no exemplo da figura 5.2

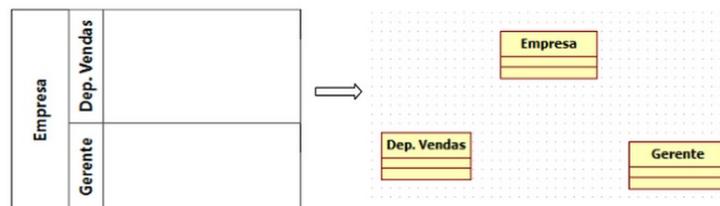


Figura 5.2 – Exemplo de mapeamento de raias para classes

5.1.3 Mapeamento de atividade

Atividades representam trabalhos realizados por uma entidade que participa de um processo (SPECIFICATION, 2006). Cada entidade precisa realizar determinadas atividades para obter resultados que agregam valor ao negócio.

Esse conceito é análogo ao de responsabilidades de classes, em um diagrama de classes, visto que uma responsabilidade é um contrato ou obrigações de uma determinada classe (FORCE, 1999). Dessa forma, atividades são consideradas neste trabalho como candidatas a responsabilidades de classes.

Conforme os diagramas de classes vão sendo aprimorados pelo analista, este pode traduzir responsabilidades sugeridas em um conjunto de atributos e operações, capazes de atender as responsabilidades da classe, como representado no exemplo da figura 5.3.

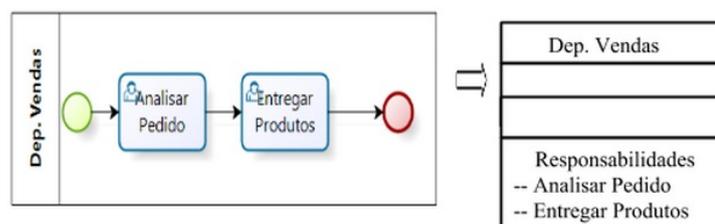


Figura 5.3 – Exemplo de mapeamento de atividades para responsabilidades de classe

5.1.4 Mapeamento de rótulos das atividades

Um objeto de negócio representa uma entidade ativa no domínio do negócio, incluindo o seu nome de negócio, atributos, comportamentos, relações e restrições (SPECIFICATION, 2006).

Objeto de negócio pode representar, por exemplo, uma pessoa ou um conceito. Em atividades como “Aprovar documento”, “Emitir ordem de serviço”, os objetos de negócio são o “documento” e a “ordem de serviço”, respectivamente.

Esse conceito também é análogo ao de uma classe. Ambos, objeto de negócio e classe, servem para representar entidades (ou “objetos”) que compartilham características. A parte “objeto” contida nos rótulos das atividades, considerando o formato “verbo (imperativo) + objeto”, é considerada neste trabalho como candidata a classe, como representado no exemplo da figura 5.4.

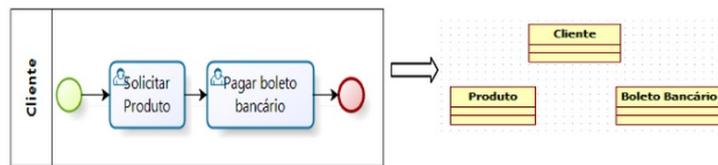


Figura 5.4 – Exemplo de mapeamento da parte “objeto” do rótulo de atividades para classes

5.1.5 Relacionamentos entre classes obtidas a partir de piscinas e raias

Há duas formas previstas para relacionar as classes obtidas a partir de piscinas e raias.

A primeira forma é usar um relacionamento de generalização entre classes obtidas a partir de piscinas e raias caso existir uma relação do tipo “é um” ou “é um tipo de” entre elas. Neste caso a classe obtida a partir da piscina deve ser considerada como classe mãe e as classes obtidas a partir das raias devem ser consideradas como classes filhas.

A segunda forma é usar um relacionamento de composição entre classes obtidas a partir de piscinas e raias quando se quer diferenciar o “todo” da “parte”. Neste caso a classe obtida a partir da piscina deve ser considerada como classe “todo” e as classes obtidas a partir das raias devem ser consideradas como classes “parte”. Esse relacionamento se justifica, pois segundo a especificação da BPMN (SPECIFICATION, 2006), raias ficam sempre dentro de piscinas. Se a piscina deixar de existir as suas raias também desaparecerão. A figura 5.5 apresenta um exemplo desse mapeamento.

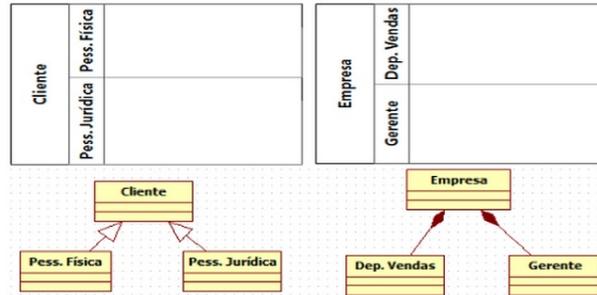


Figura 5.5 – Exemplo de relacionamentos entre as classes obtidas a partir de piscina e raias

5.1.6 Relacionamentos entre classes obtidas a partir de piscina (ou raias) e as obtidas a partir de rótulos de atividades

Para classes obtidas a partir da parte “objeto” dos rótulos de atividade, a parte “verbo” do rótulo é usada para nomear o relacionamento de associação entre a classe obtida a partir da piscina (ou raia) do processo e a classe obtida a partir do rótulo da atividade contida dentro dessa piscina (ou raia). A figura 5.6 apresenta um exemplo desse mapeamento.

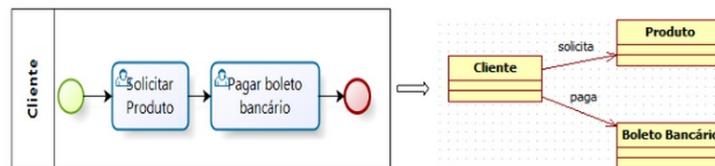


Figura 5.6 – Exemplo de relacionamentos entre as classes obtidas a partir de piscina e de rótulos de atividades

5.1.7 Mapeamento de desvios

Este trabalho considera apenas o mapeamento de desvio exclusivo (XOR) da BPMN para elementos do diagrama de classes. Demais tipos de desvios como desvio inclusivo, desvio paralelo podem ser temas de trabalhos futuros.

No caso do desvio exclusivo, em um ponto de divergência de fluxo de sequência, apenas um fluxo de saída é selecionado entre as alternativas existentes.

Essa situação pode representar dois tipos de informação em um diagrama de classes:

- Um atributo (ex: *status*) para diferenciar o estado das instâncias da classe obtida a partir da parte “objeto” do rótulo da atividade que precede o desvio.
- Um relacionamento de generalização entre a classe obtida a partir da parte “objeto” do rótulo da atividade que precede o desvio (nesse caso, é considerada como classe mãe)

e as classes obtidas a partir dos estados das instâncias dessa classe caso as instâncias possuam mais características do que a classe mãe.

A figura 5.7 apresenta um exemplo desse mapeamento.

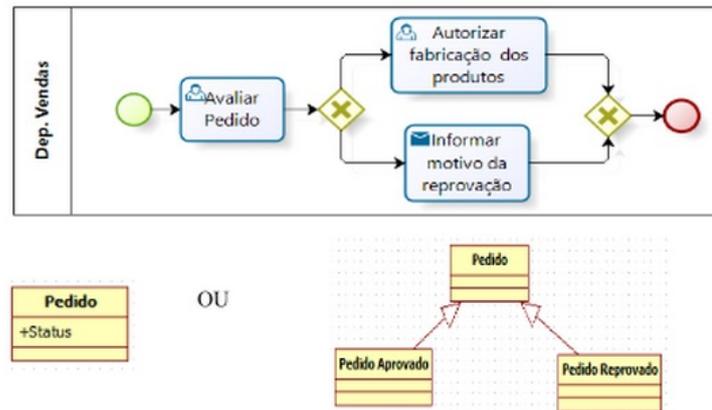


Figura 5.7 – Exemplo de mapeamento de desvio exclusivo em um atributo status ou em um relacionamento de generalização

5.1.8 Mapeamento de eventos

Evento é algo que ocorre durante o processo. Existem três tipos de eventos: evento inicial, evento intermediário e evento final. Este trabalho considera apenas o mapeamento de eventos com causa definida principalmente os eventos do tipo temporal e condicional. Demais tipos de eventos como mensagem, cancelamento, entre outros podem ser temas de trabalhos futuros. Um evento sem causa definida não tem mapeamento para elemento de diagrama de classes pois não acrescenta nenhuma informação ao diagrama de classes.

Um evento temporal (ou condicional) é mapeado para anotação associada à classe obtida a partir da parte “objeto” do rótulo da atividade que sucede a esse evento. A anotação apresenta informações sobre o evento. A figura 5.8 apresenta um exemplo do mapeamento de um evento de início temporal.

Como não existem eventos de fim temporal (ou condicional), a regra de mapeamento se aplica apenas para eventos de início e evento intermediários.

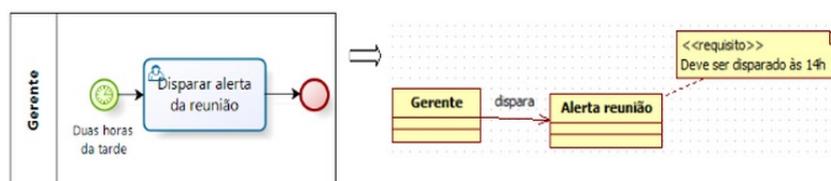


Figura 5.8 – Exemplo de mapeamento de evento temporal para a anotação

Um resumo com as regras de mapeamento de processo para diagrama de classes é apresentado na figura 5.9.

Regra	Descrição
1	se houver piscinas, mapear cada uma delas para classes
2	se houver raias, mapear cada uma delas para classes
3	criar relacionamentos de: 3.1. generalização entre classes obtidas a partir de uma piscina e suas respectivas raias quando existir uma relação do tipo "é um" ou "é um tipo de" entre a piscina e as raias, sendo a piscina a classe-mãe e as raias as classes-filhas 3.2. composição entre classes obtidas a partir de uma piscina e suas respectivas raias quando existir uma relação do tipo "todo-parte" entre a piscina e as raias, sendo a piscina a classe-todo e as raias as classes parte
4	mapear cada tarefa de uma piscina (ou raia) para responsabilidades da classe obtida por meio da respectiva piscina (ou raia)
5	se os rótulos das atividades estiverem no formato "verbo (imperativo) + objeto" (como, por exemplo, "Avaliar pedido" ou "Enviar ordem de serviço"), transformar a parte referente ao "objeto" do rótulo em classe. Para cada classe criada dessa forma: 5.1. criar um relacionamento de associação entre a classe obtida por meio do rótulo de atividade e a classe obtida por meio da piscina (ou raia) que contém a respectiva atividade 5.2. se houver piscina (ou raia), nomear esse relacionamento de associação com a parte "verbo (imperativo)" do rótulo 5.3 se houver eventos temporais (ou condicionais), mapear cada um deles para anotação associada à classe obtida a partir da parte "objeto" do rótulo da atividade que sucede a esses eventos. Colocar na anotação informações sobre o evento
6	mapear cada desvio exclusivo (ou seja, XOR) para um dos dois tipos de elementos: 6.1. um atributo (ex.: status) usado para diferenciar o estado de instâncias da classe obtida a partir da parte "objeto" do rótulo da atividade que precede o desvio 6.2. um relacionamento de generalização entre a classe obtida a partir da parte "objeto" do rótulo da atividade que precede o desvio (nesse caso, é considerada como classe mãe) e as classes obtidas a partir dos estados das instâncias dessa classe caso as instâncias possuírem mais características do que a classe mãe

Figura 5.9 – Regras de mapeamento entre processo de negócio em BPMN e diagramas de classes da UML

5.1.9 Exemplo ilustrativo de mapeamento completo das regras propostas

Para ilustrar o uso das regras de mapeamento, são apresentados o modelo BPMN (Figura 5.10) e a descrição textual de um processo adaptado de Suarez et al. (2008).

O processo de vendas de uma fábrica de cadeiras de escritório, cujo modelo é apresentado na figura 5.10, é descrito da seguinte forma: um cliente acessa um site de uma empresa X especializada na fabricação de cadeiras de escritório; o cliente faz um pedido que deve incluir seu nome, CPF, endereço, produtos desejados e data de solicitação; ao receber o pedido do cliente, o departamento de vendas confere o pedido e inicia o processo enviando ao gerente de catálogo, que é o responsável pela análise do pedido; o departamento de vendas é então responsável por informar ao cliente o resultado final da análise de seu pedido; o gerente de catálogo analisa a viabilidade do pedido:

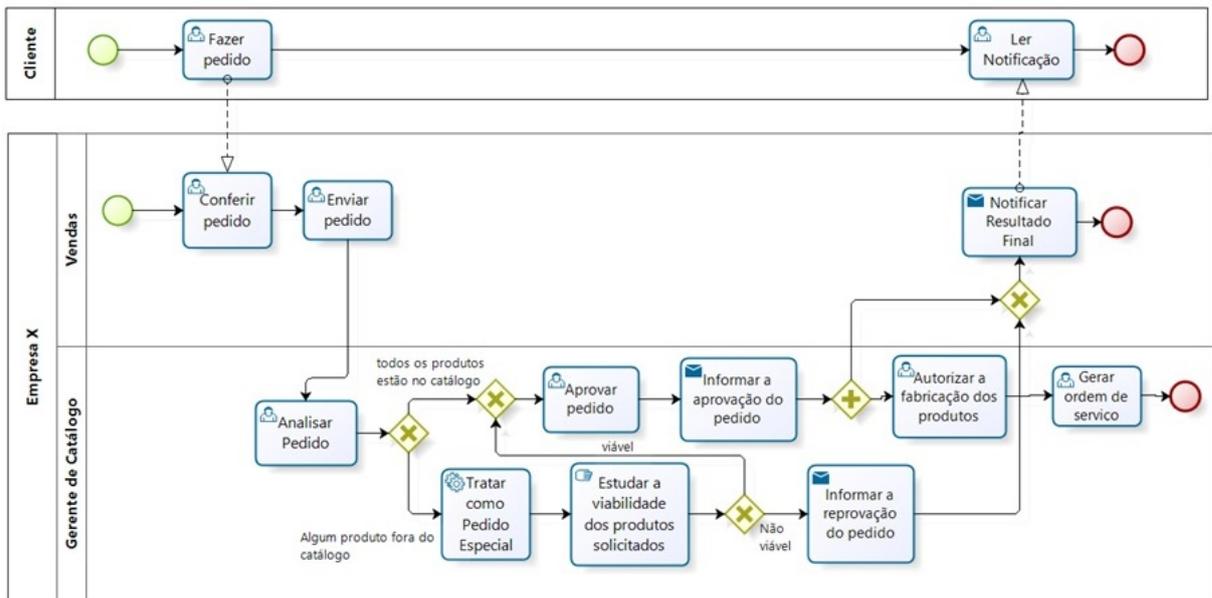


Figura 5.10 – Modelo de processo de fábrica de cadeiras de escritório

1. Se todos os produtos solicitados estiverem no catálogo, o pedido é aprovado, a fabricação dos produtos é autorizada, e uma ordem de serviço para a fabricação dos produtos é gerada.
2. Caso contrário, é considerado como um pedido especial e o gerente de catálogo deve estudar a viabilidade dos produtos solicitados.
 1. Se for viável o pedido é aprovado, a fabricação é autorizada, e é gerada uma ordem de serviço.
 2. Caso contrário, o pedido é reprovado.

Aplicando as regras de mapeamento como mostra a figura 5.11 no modelo de processo da figura 5.10, um fragmento de diagrama de classes contendo oito classes (Figura 5.12) foi gerado.

As classes obtidas a partir das piscinas (Cliente e Empresa X) e raias (Vendas, Ger. Catálogo) são apresentadas com suas respectivas responsabilidades. As classes “Pedido Esp”, “Pedido”, “Ord. Serviço” e “Produto” foram obtidas a partir dos rótulos das atividades.

Através da regra 6.1 (ou seja, mapeamento do desvio XOR) apresentada na figura 5.9, foi obtido o atributo “status” encontrado na classe “Pedido” a partir do desvio exclusivo que sucede a tarefa “Analisar Pedido”.

O relacionamento de composição entre a classe Empresa X, Vendas e Ger. Catálogo foi usado para apresentar que essas duas classes têm sua existência dependente da classe Empresa X. Se esta deixar de existir, outras duas também deixarão de existir. O relacionamento de as-

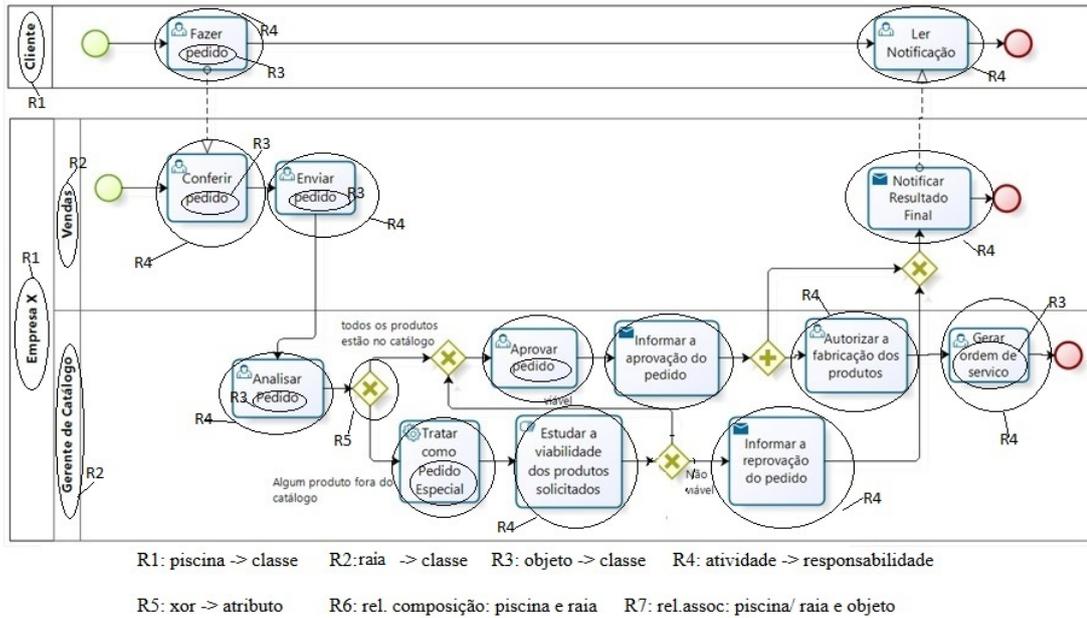


Figura 5.11 – Aplicação das regras de mapeamento no modelo de processos da figura 5.10

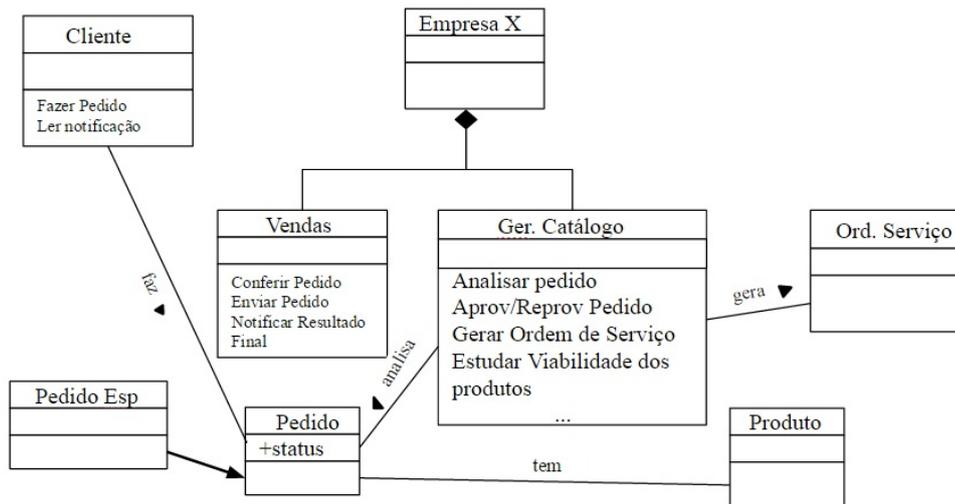


Figura 5.12 – Diagrama de classes correspondente ao processo de solicitação de cadeiras de escritório

sociação entre as classes Cliente - Pedido; Ger.Catálogo - Pedido; Ger. Catálogo - Ord.Serviço foram obtidos a partir das regras 5.1 e 5.2 apresentadas na figura 5.9.

Como se trata de fragmentos de diagrama classes, vale a pena lembrar que o relacionamento de associação estabelecido entre as classes “Pedido” e “Produto” e o relacionamento de generalização entre a classe “Pedido” e “Pedido Esp” não foram obtidos a partir das regras de mapeamento propostas neste trabalho. Assim, em alguns casos, os relacionamentos entre classes devem ser estabelecidos por um analista. O diagrama de classes obtido é apenas a versão inicial a ser posteriormente melhorado.

6 EXPERIMENTO PARA VALIDAÇÃO DO MÉTODO PROPOSTO

Este capítulo apresenta os experimentos executados com alunos de graduação e pós-graduação da disciplina de Engenharia de Software do curso de Ciência da Computação do Instituto de Informática com o intuito de avaliar a criação de diagramas de classes com e sem o uso das regras de mapeamento apresentadas neste trabalho.

A seção 6.1 descreve a metodologia adotada, enquanto a seção 6.2 analisa os resultados obtidas através do experimento.

6.1 Metodologia adotada

Um experimento foi realizado com alunos de graduação e pós-graduação da disciplina de Engenharia de Software do curso de Ciência da Computação do instituto de informática da UFRGS com o intuito de avaliar a criação dos diagramas de classes com e sem o uso das regras de mapeamento propostas neste trabalho. Para isso, foi disponibilizado um tutorial¹ que apresenta definições sobre elementos notacionais da BPMN e sobre o diagrama de classes.

Ao todo, 53 alunos foram divididos em dois grupos conforme a disponibilidade de computadores no laboratório onde foi realizado o experimento.

- Grupo A, com 25 alunos, realizou a atividade sem o uso das regras de mapeamento propostas.
- Grupo B, com 28 alunos, realizou a atividade usando as regras de mapeamento propostas.

Para ambos os grupos, foi usado um processo de locação de veículo adaptado de Guedes (2009) como apresentado na figura 6.1. Um questionário foi aplicado aos alunos via Google forms. A figura 6.2 apresenta um fragmento de diagrama de classes, adaptado de Guedes (2011), referente ao modelo de processo de negócio apresentado na figura 6.1.

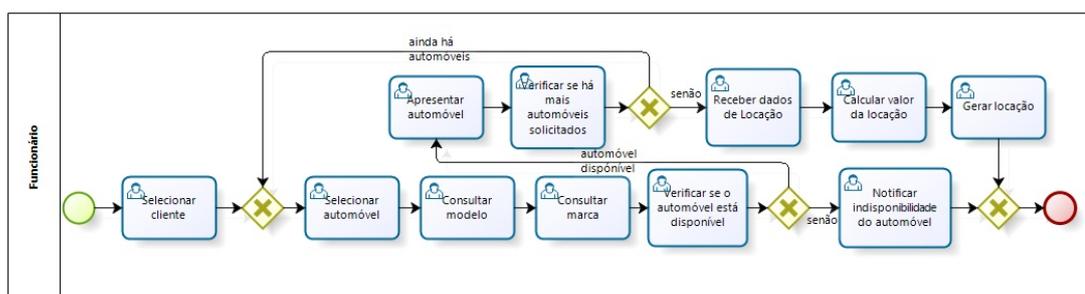


Figura 6.1 – Modelo de processo de locação de veículo

¹<https://docs.google.com/document/d/1S2OLyuTQXezaoSUE_1L3thgQXSi8AVvhcMSwHYpIBFQ/edit>

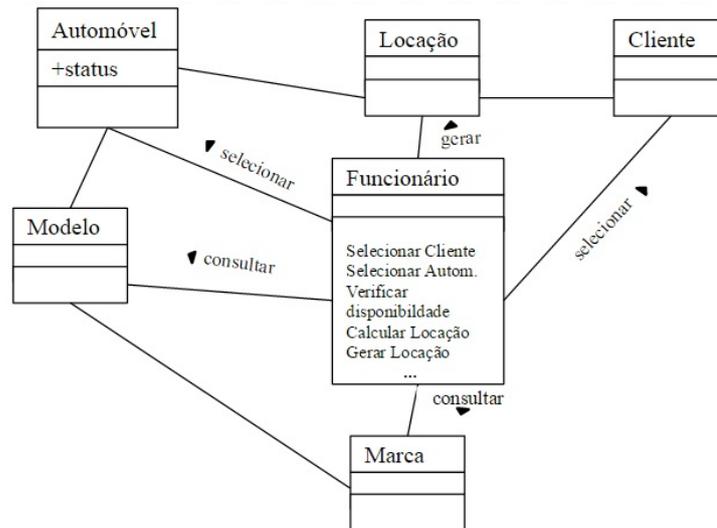


Figura 6.2 – Fragmento de diagrama de classes, adaptado de Guedes (2011), referente ao modelo de processo de negócio apresentado na figura 6.1

6.2 Análise dos resultados obtidos

O roteiro de experimento aplicado aos alunos foi projetado para avaliar a criação dos diagramas de classes com e sem o uso das regras de mapeamento apresentadas neste trabalho.

Foi realizada uma pesquisa para identificar o nível de conhecimento dos participantes em BPMN e diagrama de classes da UML. A Figura 6.3 apresenta os resultados sobre o conhecimento em BPMN e em diagrama de classes para cada grupo.

Os resultados apontam que a maioria dos alunos tinham conhecimento nenhum ou básico em BPMN e básico ou intermediário em diagrama de classes. A figura 6.4 mostra os resultados

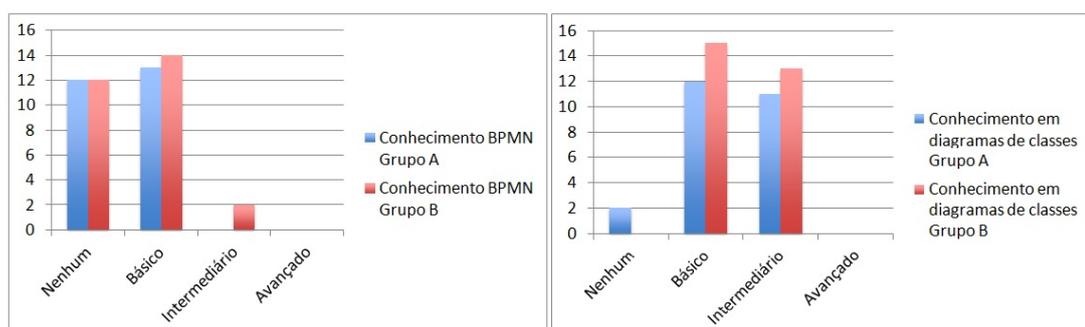


Figura 6.3 – Conhecimento dos alunos em BPMN e em diagrama de classes

obtidos após análise dos diagramas de classes extraídos pelos alunos.

Dezoito alunos do grupo B, que representa 64 % do grupo, (grupo com as regras de mapeamento) obtiveram um diagrama de classes com um número de seis classes iguais às apresentadas (Figura 6.2).

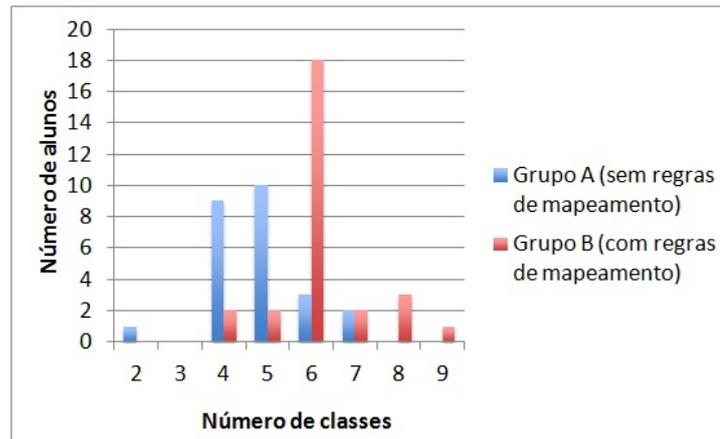


Figura 6.4 – Classes vs número de alunos

Além disso, seis alunos do grupo B identificaram mais do que seis classes, o que representam outros 21 % do grupo. Ao total, 85 % dos alunos do grupo B identificaram seis ou mais classes. Por outro lado, para o grupo A, apenas três alunos (ou seja, 12 %) identificaram seis classes, e 20 alunos (ou seja, 80 % do grupo) identificaram menos do que seis classes.

A Figura 6.5 apresenta o resultado de satisfação dos alunos do grupo B em relação às regras de mapeamento que foram fornecidas. A maioria concordou parcialmente que as regras de mapeamento auxiliam na extração de diagramas de classes a partir de um modelo de processo de negócio. Somente dois alunos não concordaram com o uso das regras.



Figura 6.5 – Satisfação dos alunos com as regras de mapeamento

Um fator importante foi o fato do experimento ter sido realizado presencialmente. Isto permitiu a análise em tempo real das principais dificuldades apresentadas pelos participantes, através do acompanhamento de suas ações. O grupo B (grupo que utilizou as regras de mapeamento) levou mais tempo para submeter o diagrama de classes. Isso pode ter acontecido pelo fato de que os alunos do grupo B tiveram que ler as regras de mapeamento antes de responder ao questionário. Desta forma, o tempo não foi considerado como elemento de comparação.

7 CONCLUSÃO

Neste trabalho, foi apresentado um método para facilitar a extração de fragmentos de diagramas de classes a partir de modelos de processos de negócio em BPMN.

Baseado na especificação da BPMN e da UML, cada elemento notacional da BPMN é analisado para identificar um possível mapeamento com o respectivo elemento do diagrama de classes da UML.

As regras de mapeamento de modelos em nível de processo de negócio para modelos em nível de sistemas apresentadas neste trabalho demonstram que é possível obter fragmentos de diagramas de classes a partir de um modelo de processo de negócio em BPMN sem a necessidade de diagramas intermediários.

Além disso, pelos resultados obtidos, com o experimento de validação realizado, pode-se concluir que as regras auxiliam na extração de tais diagramas de classes. O diagrama de classes obtido poderá ser uma versão inicial a ser posteriormente melhorado. Os especialistas podem na sequência validar e completar o diagrama de classes obtido.

O estudo apresentado atende ao objetivo proposto que é mapear um modelo de processo de negócio em BPMN para diagrama de classes. Entretanto, o estudo realizado apresentou limitações quanto à validação dos modelos de processo. Não foi realizada uma verificação explícita do modelo de processo para avaliar sua correteza. Além disso, não foi proposto um método de relacionamento entre classes obtidas a partir de rótulos de atividades.

A resolução das limitações apresentadas são sugestões de possíveis trabalhos futuros. Além disso, sugere-se a criação de uma ferramenta que auxilie na extração de diagramas de classes com base no algoritmo apresentado na figura 5.9. Dessa forma, o mapeamento poderia ser realizado de forma semi-automática facilitando cada vez mais a tarefa de análise do projetista.

REFERÊNCIAS

- BARROS, J. P. From activity diagrams to class diagrams. In: CITESEER. **Workshop Dynamic Behaviour in UML Models: Semantic Questions In conjunction with Third International Conference on UML, York, UK**. [S.l.], 2000.
- BOOCH, G. **The unified modeling language user guide**. [S.l.]: Pearson Education India, 1999.
- COALITION, W. M. **Terminology & Glossary**. [S.l.], 1996.
- DUMAS, M. et al. **Fundamentals of business process management**. [S.l.]: Springer, 2013.
- FORCE, U. R. T. **OMG UML Specification**. 1999.
- GROUP, O. M. et al. Mda guide version 1.0. 1. **OMG specifica**, 2003.
- LINDSAY. Business processes—attempts to find a definition. **Information and software technology**, Elsevier, v. 45, n. 15, p. 1015–1019, 2003.
- MILLER et al. Mda guide version 1.0. 1. **Object Management Group**, v. 234, p. 51, 2003.
- MOLINA, J. G. et al. Towards use case and conceptual models through business modeling. In: **Conceptual Modeling—ER 2000**. [S.l.]: Springer, 2000. p. 281–294.
- NAVATHE. Sistemas de banco de dados. **Sham, Addison**, 2005.
- OKAWA, T. et al. A method of linking business process modeling with information system design using uml and its evaluation by prototyping. In: IEEE. **Asia-Pacific Service Computing Conference, The 2nd IEEE**. [S.l.], 2007. p. 458–465.
- RECKER, J. Opportunities and constraints: the current struggle with bpmn. **Business Process Management Journal**, Emerald Group Publishing Limited, v. 16, n. 1, p. 181–201, 2010.
- RODRÍGUEZ, A. et al. Semi-formal transformation of secure business processes into analysis class and use case models: An mda approach. **Information and Software Technology**, Elsevier, v. 52, n. 9, p. 945–971, 2010.
- RUNGWORAWUT, W. From business world to software world: Deriving class diagrams from business process models. In: **Proc. of the 5th WSEAS Int. Conf. on Applied Informatics and Communications**. [S.l.: s.n.], 2005. p. 233–238.
- SPECIFICATION, B. P. M. N. Omg final adopted specification. **Object Management Group (February 2006)**. Online available: http://www.omg.org/bpmn/Documents/OMG_Final_Adopted_BPMN_1-0_Spec_06-02-01.pdf (accessed: 23.08.14), 2006.
- SUAREZ, E. Transformation of a process business model to domain model. In: **Proceedings of the World Congress on Engineering**. [S.l.: s.n.], 2008. v. 1.
- WESKE, M. **Business process management: concepts, languages, architectures**. [S.l.]: Springer Science & Business Media, 2012.

APÊNDICE A — QUESTIONÁRIO APLICADO

A.1 Questionário Grupo A

Com base no tutorial lido, tente extrair um diagrama de classe a partir do modelo de processo BPMN abaixo que ilustra um processo de locação de automóveis.

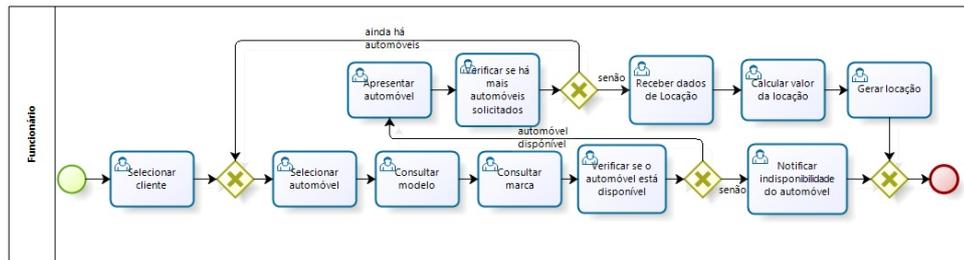


Figura A.1 – Modelo de processo de locação de veículo

1. Conhecimento em BPMN: (campo múltipla escolha. Opções: “nenhum”, “básico”, “intermediário” e “avançado”)
2. Conhecimento em Diagrama de classe: (campo múltipla escolha. Opções: “nenhum”, “básico”, “intermediário” e “avançado”)
3. Quantas classes foram obtidas? (campo texto)
4. Quais são as possíveis classes? Liste cada classe com as possíveis operações entre parênteses caso houver. (campo texto)
5. Quais relacionamentos foram estabelecidos entre as classes? Exemplo: Classe A,B (Generalização, tendo A classe mãe e B classe filha); X,Y, Z (Composição, tendo X classe "Todo", Y e Z classes "Parte"). (campo texto)

A.2 Questionário Grupo B

Com base no tutorial lido e nas regras de mapeamento, tente extrair um diagrama de classe a partir do modelo de processo BPMN abaixo que ilustra um processo de locação de

APÊNDICE B — ARTIGO SUBMETIDO

O presente artigo foi submetido para o Simpósio Brasileiro de Banco de Dados (SBBD) 2015, a qual é promovido pela Sociedade Brasileira de Computação. O evento será realizado em Pretrópolis do dia 13 a 16 de outubro de 2015.

Extração de Fragmentos de Diagramas de Classes a partir de Modelos de Processo de Negócio

Mike M. Tshibende¹, Carlos Habekost¹, Lucinéia Heloisa Thom¹, Marcelo Fantinato²

¹Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{mmtshibende, carlos.habekost, lucineia}@inf.ufrgs.br

²Escola de Artes, Ciências e Humanidades (EACH) – Universidade de São Paulo (USP)
Caixa Postal 02.125 – 03-828-000 – São Paulo – SP – Brazil

m.fantinato@usp.br

Abstract. *The business process modeling has attracted the attention of systems analysts, as it helps to modeling, analyzing and designing systems to be developed. Nonetheless, the notation used to model the processes, BPMN (Business Process Management Notation), differs from the one used to model systems, UML class diagrams. Thereby, this work aim at presenting a study in order to facilitate class diagram extraction from a business process model represented by the BPMN. As result of this study, a methodology has been developed to facilitate the extraction of system models, starting from models in business level. With the class diagram obtained, the systems analyst may have a static view of the system's vocabulary, including the different relationships between classes, as well as a logical view of a database schema.*

Resumo. *A modelagem de processo de negócio tem atraído a atenção dos analistas de sistemas, pois auxilia a modelagem, análise e projeção de sistemas a serem desenvolvidos. No entanto, a notação usada para modelar processos, BPMN (Business Process Management Notation) é diferente da usada para modelar sistemas, diagrama de classes da UML. Assim, este trabalho visa apresentar um estudo para facilitar a extração de diagramas de classes a partir de um modelo de processo de negócio em BPMN. Como resultado deste estudo, tem-se um conjunto de regras para facilitar a extração de modelos de sistema, a partir de modelos em nível de processo de negócio. Com o diagrama de classes obtido, o analista de sistemas poderá ter uma visão estática do vocabulário do sistema, incluindo os diferentes relacionamentos entre as classes e poderá ter uma visão lógica de um esquema de banco de dados.*

1. Introdução

Um processo de negócio consiste em uma série de atividades relacionadas que são realizadas para atingir um determinado objetivo de negócio [Weske 2012]. Um Processo de negócio permite ao analista compreender melhor o negócio e conseqüentemente ajuda esse analista a melhorá-lo e torná-lo mais flexível para mudanças que surgem no mercado.

A notação BPMN (Business Process Modeling and Notation) é uma das mais populares e mais usadas pelos analistas de negócio [Rodríguez et al. 2010]. BPMN objetiva

fornecer uma notação de fácil entendimento por todas as partes envolvidas, incluindo os analistas de negócios, os desenvolvedores e finalmente aqueles responsáveis por gerenciar e monitorar esses processos [Specification 2006].

Considerando especificamente a modelagem de sistemas no desenvolvimento de software, a linguagem UML (Unified Modeling Language) é considerada padrão [Force 1999]. Dos diferentes diagramas da UML, o diagrama de classes é aquele encontrado com maior frequência no contexto de orientação a objetos [Booch et al. 1999]. Um diagrama de classes serve para modelar a visão estática do projeto de um sistema [Booch et al. 1999] e pode ser considerado uma notação alternativa aos diagramas Entidade-Relacionamento (ER) usados em projetos de bancos de dados [Elmasri et al. 2005].

Analistas de negócio e analistas de sistemas usam notações e linguagens diferentes para realizar suas tarefas. Como ambos tipos de profissionais trabalham em níveis de abstração diferentes, o mapeamento de um modelo em nível de processo de negócio para um modelo em nível de sistema pode se tornar uma tarefa difícil de ser realizada. Assim, o analista de sistemas precisa ter conhecimento sobre BPMN para entender as diferentes semânticas existentes nos elementos desses modelos.

O mapeamento de um modelo em nível de processo de negócio para um modelo em nível de sistema vem sendo objeto de diversos estudos na área de Engenharia de Software. Analisando essa questão a partir de uma abordagem de MDA (Model-driven Architecture), verifica-se que os modelos de processos de negócio se encontram em um nível CIM (Computing Independent Model), enquanto que os modelos de sistemas localizam-se nos níveis de PIM (Platform Independent Model) e de PSM (Platform Specific Model) [Belaunde et al. 2003]. Assim, o mapeamento entre um modelo em um nível CIM para um modelo do nível PIM e, posteriormente, de nível PSM, pode ser realizado usando os princípios de transformação entre modelos propostos pela MDA. Existem diversos trabalhos que tratam do problema de mapeamento de modelos de processo de negócio para modelos de sistemas. Na maioria dos trabalhos encontrados ([Suarez et al. 2008],[Okawa et al. 2007],[Barros and Gomes 2000]), os modelos de processo são representados através de notações diferentes da BPMN. Em [Rodríguez et al. 2010] é apresentada uma proposta de mapeamento de processo em BPMN para diagrama de atividades da UML. O diagrama de atividades obtido é transformado em diagramas de classes e diagramas de casos de uso. Porém, não é explicado claramente como foram estabelecidos os relacionamentos entre as classes.

O objetivo deste trabalho é apresentar um conjunto de regras que auxilie no mapeamento de modelos em nível de processo de negócio (ou seja, em nível CIM), especificados em BPMN, para modelos em nível de sistema (em nível PIM), especificados em diagramas de classes da UML. Esse trabalho visa propor um mapeamento manual com assistência de um computador. O mapeamento automático pode ser assunto de um trabalho futuro.

Como benefício, as regras propostas visam auxiliar na identificação de classes e seus possíveis relacionamentos a partir do modelo de processo de negócio. Além disso, as regras propostas possibilitam ao analista de sistemas obter uma visão lógica de um esquema de banco de dados, e podem ser usadas para criar uma ferramenta que automatize

esse mapeamento.

Para validar as regras de mapeamento foi realizado um experimento com alunos de graduação e pós-graduação da disciplina de Engenharia de Software do curso de Ciência da Computação de uma universidade pública federal. Melhores resultados foram obtidos com os alunos que usaram as regras propostas.

O restante do artigo está estruturado da seguinte forma: a seção 2 trata dos fundamentos em BPMN e diagramas de classes de UML. A seção 3 trata das regras de mapeamento de processos de negócio especificados em BPMN para diagramas de classes da UML. Na seção 4, é abordado o experimento usado para validar as regras de mapeamento. Por fim, a seção 5 aborda as conclusões do trabalho.

2. Fundamentos

Um modelo de processo de negócio é a representação genérica desse processo. O modelo facilita o entendimento do processo e auxilia no compartilhamento do conhecimento do processo com pessoas envolvidas nele. Além disso, o modelo de processo de negócio auxilia na identificação e prevenção de problemas [Dumas et al. 2013]. Projetistas de OMG (Object Management Group) analisaram diversas notações existentes e buscaram consolidar as melhores ideias destas notações divergentes em uma única notação padrão, BPMN [Specification 2006]. Exemplos das notações revisadas são: diagrama de atividades da UML, UML EDOC Business Processes, IDEF, ebXML BPSS, Activity-Decision Flow (ADF) Diagram, RosettaNet, LOVeM, e Event-Process Chains (EPCs). BPMN tem como principal objetivo fornecer uma notação gráfica de fácil entendimento por parte dos envolvidos com o processo incluindo os analistas que capturam as primeiras definições do processo, os desenvolvedores responsáveis por sua automação e, finalmente, aqueles responsáveis por gerenciá-los e monitorá-los [Specification 2006]. BPMN é uma notação com grande poder de expressão e que está sendo usada mundialmente por diversas organizações como ferramenta de apoio a documentação, simulação de processos, etc. [Recker 2010]. Os elementos gráficos da BPMN usados para modelar processos são divididos nas seguintes categorias: objetos de fluxo; objetos de dados; objeto de conexão; partições e artefatos [Specification 2006].

Diagramas de classes da UML são encontrados com maior frequência na modelagem de sistemas orientados a objetos. Eles são usados para modelar a visão estática do projeto de um sistema. Na maioria dos casos, isso envolve a modelagem do vocabulário do sistema, a modelagem de colaborações ou a modelagem de esquemas [Booch et al. 1999]. Além disso, diagramas de classes oferecem suporte para os requisitos funcionais (os serviços que o sistema deverá fornecer aos usuários finais) de um sistema. Através do diagrama de classes é possível obter uma visão lógica de um esquema de banco de dados [Booch et al. 1999]. Os diagramas de classes podem ser considerados uma notação alternativa aos diagramas Entidade-Relacionamento (ER) [Elmasri et al. 2005]. Onde as classes representariam as tabelas e os relacionamentos entre as tabelas são equivalentes aos relacionamentos entre as classes. O diagrama de classes inclui: classes, interfaces e relacionamento de dependência, de generalização e de associação [Force 1999].

3. Mapeamento de modelos de processo de negócio para diagramas de classes

O objetivo do mapeamento proposto é possibilitar a obtenção de informações para um diagrama de classes da UML a partir de um modelo de processo de negócio. O mapeamento entre os modelos foi realizado com base na especificação da BPMN [Specification 2006] e de UML [Force 1999]. Cada elemento notacional da BPMN, conforme apresentado na tabela 1, foi analisado para identificar um possível mapeamento com o respectivo elemento do diagrama de classes. Objetos de fluxo (eventos temporais, atividades do tipo tarefa e desvios exclusivos), as partições (piscinas e raias) foram considerados neste trabalho. Já os objetos de conexão (fluxo de sequência, fluxo de mensagem e associação), demais tipos de atividades, eventos e desvios não foram considerados neste trabalho, mas podem ser assunto para trabalhos futuros.

Tabela 1. Elementos notacionais da BPMN considerados

Elementos BPMN	Descrição
Piscina	Usada para representar diferentes organizações que participam de um processo.
Raia	Usada para particionar uma piscina, permitindo representar diferentes departamentos dentro de uma organização.
Atividade	Usada para representar o trabalho realizado dentro de um processo (neste trabalho, apenas atividade do tipo tarefa é considerada).
Desvio	Usado para controlar as divergências e as convergências no fluxo de sequência de um processo (nesse trabalho, apenas desvio exclusivo, ou seja, XOR, é considerado).
Evento	Usado para representar algo que ocorre no início, durante ou no fim de um processo (neste trabalho, apenas o evento temporal é considerado).

Para alguns casos, foram identificados mais de uma possibilidade de mapeamento. Para obter melhores resultados, o processo deve ter sido modelado corretamente e sem ambiguidades. Um processo mal modelado pode levar a construção de um diagrama de classes que não reflete a visão correta do projeto de sistema. Contudo, para o conjunto de regras de mapeamento proposto neste trabalho, o modelo de processo foi assumido como correto.

Apenas a especificação gráfica dos modelos de processos é usada pelo método, ou seja, a respectiva descrição textual não é usada. Conseqüentemente, não objetiva-se gerar diagrama de classes completo, mas apenas um fragmento do mesmo, incluindo: nomes de classes; responsabilidades de classes; e sugestões de relacionamentos entre as classes. Para cada modelo de processo, um único diagrama de classes é criado; Conseqüentemente, atividades compostas (ou seja, subprocessos) são mapeadas separadamente, gerando outros diagramas de classes.

3.1. Verificação de um modelo de processo de negócio

Um modelo de processo de negócio deve ser modelado corretamente para obtenção de melhores resultados em seu mapeamento para o diagrama de classes. Essa correteza deve ser garantida pelo projetista do modelo de processo.

Porém, a especificação da BPMN [Specification 2006] não define, por exemplo, regras de como definir rótulos de atividades em um processo. Atualmente, existem estudos com objetivo de contribuir com boas práticas de nomenclatura das atividades de processos. Mendling et al.(2010), por exemplo, realizaram testes com mais de 600 modelos de processo e conseguiram demonstrar que o rótulo na forma “verbo (na forma

imperativa) + objeto” é mais adequado na nomenclatura das atividades. Por exemplo, em atividades como “Aprovar documento”, “Emitir ordem de serviço”, os objetos são o “documento” e a “ordem de serviço”, respectivamente. Esse formato diminui possíveis ambiguidades no entendimento das atividades. Visando obter melhores resultados, as regras de mapeamento apresentadas neste trabalho assumem que os modelos de processo são construídos seguindo esse formato verbo (imperativo) + objeto.

3.2. Mapeamento de piscina

Piscinas representam entidades (pessoas, organizações, etc.) responsáveis pela execução de uma parte do processo ou um conjunto de atividades [Specification 2006]. Esse conceito é análogo ao de uma classe no contexto de diagrama de classes. Ambas, piscina e classe, servem para representar entidades (ou “objetos”) que compartilham características e responsabilidades. Assim, neste trabalho uma piscina é considerada como candidata a classe, como representado no exemplo da figura 1.



Figura 1. Exemplo de mapeamento de piscina para classe

3.3. Mapeamento de raia

Raias frequentemente representam papéis dentro de uma piscina (representando, por exemplo, departamentos de uma organização) [Specification 2006]. Esse conceito também é análogo ao de uma classe. Ambas, raia e classe, assim como no caso anterior referente a piscina, servem para representar entidades (ou “objetos”) que compartilham características e responsabilidades. Assim, neste trabalho, uma raia também é considerada como candidata a classe, como representado no exemplo da figura 2

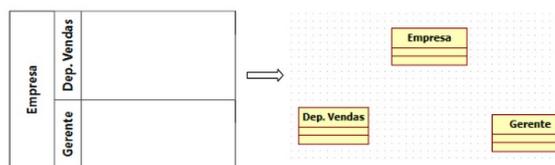


Figura 2. Exemplo de mapeamento de raias para classes

3.4. Mapeamento de atividade

Atividades representam trabalhos realizados por uma entidade que participa de um processo [Specification 2006]. Cada entidade precisa realizar determinadas atividades para obter resultados que agregam valor ao negócio. Esse conceito é análogo ao de responsabilidades de classes, em um diagrama de classes, visto que uma responsabilidade é um contrato ou obrigações de uma determinada classe [Force 1999]. Dessa forma, atividades são consideradas neste trabalho como candidatas a responsabilidades de classes. Conforme os diagramas de classes vão sendo aprimorados pelo analista, ele pode traduzir responsabilidades sugeridas em um conjunto de atributos e operações, capazes de atender as responsabilidades da classe, como representado no exemplo da figura 3.

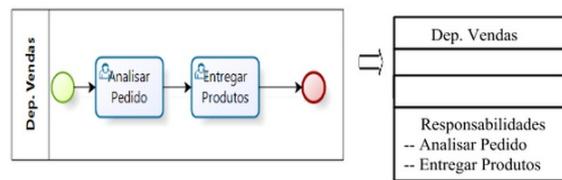


Figura 3. Exemplo de mapeamento de atividades para responsabilidades de classe

3.5. Mapeamento de rótulos das atividades

Um objeto de negócio representa uma entidade ativa no domínio do negócio, incluindo o seu nome de negócio, atributos, comportamentos, relações e restrições [Specification 2006]. Objeto de negócio pode representar, por exemplo, uma pessoa ou um conceito. Em atividades como “Aprovar documento”, “Emitir ordem de serviço”, os objetos de negócio são o “documento” e a “ordem de serviço”, respectivamente. Esse conceito também é análogo ao de uma classe. Ambos, objeto de negócio e classe, servem para representar entidades (ou “objetos”) que compartilham características. A parte “objeto” contida nos rótulos das atividades, considerando o formato “verbo (imperativo) + objeto”, é considerada neste trabalho como candidata a classe, como representado no exemplo da figura 4.

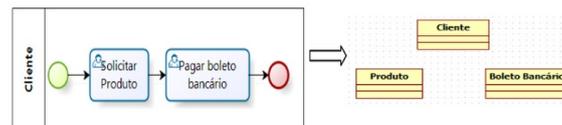


Figura 4. Exemplo de mapeamento da parte “objeto” do rótulo de atividades para classes

3.6. Relacionamentos entre classes obtidas a partir de piscinas e raias

Há duas formas previstas para relacionar as classes obtidas a partir de piscinas e raias. A primeira forma é usar um relacionamento de generalização entre classes obtidas a partir de piscinas e raias caso existir uma relação do tipo “é um” ou “é um tipo de” entre elas. Neste caso a classe obtida a partir da piscina deve ser considerada como classe mãe e as classes obtidas a partir das raias devem ser consideradas como classes filhas. A segunda forma é usar um relacionamento de composição entre classes obtidas a partir de piscinas e raias quando se quer diferenciar o “todo” da “parte”. Neste caso a classe obtida a partir da piscina deve ser considerada como classe “todo” e as classes obtidas a partir das raias devem ser consideradas como classes “parte”. Esse relacionamento se justifica, pois segundo a especificação da BPMN [Specification 2006], raias ficam sempre dentro de piscinas. Se a piscina deixar de existir as suas raias também desaparecerão. A figura 5 apresenta um exemplo desse mapeamento.

3.7. Relacionamentos entre classes obtidas a partir de rótulos de atividade

Para classes obtidas a partir da parte “objeto” dos rótulos de atividade, a parte “verbo” do rótulo é usada para nomear o relacionamento de associação entre a classe obtida a partir da piscina (ou raia) do processo e a classe obtida a partir do rótulo da atividade contida dentro dessa piscina (ou raia). A figura 6 apresenta um exemplo desse mapeamento.

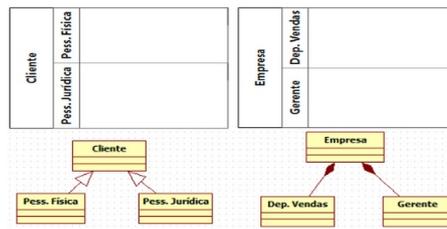


Figura 5. Exemplo de relacionamentos entre as classes obtidas a partir de piscina e raias

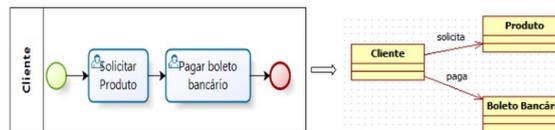


Figura 6. Exemplo de relacionamentos entre as classes obtidas a partir de piscina e de rótulos de atividade

3.8. Mapeamento de desvios

Este trabalho considera apenas o mapeamento de desvio exclusivo (XOR) da BPMN para elementos do diagrama de classes. No caso do desvio exclusivo, em um ponto de divergência de fluxo de sequência, apenas um fluxo de saída é selecionado entre as alternativas existentes. Essa situação pode representar dois tipos de informação em um diagrama de classe: (1) um atributo (ex: *status*) para diferenciar o estado das instâncias da classe obtida a partir da parte “objeto” do rótulo da atividade que precede o desvio e (2) um relacionamento de generalização entre a classe obtida a partir da parte “objeto” do rótulo da atividade que precede o desvio (nesse caso, é considerada como classe mãe) e as classes obtidas a partir dos estados das instâncias dessa classe caso as instâncias possuam mais características do que a classe mãe. A figura 7 apresenta um exemplo desse mapeamento.

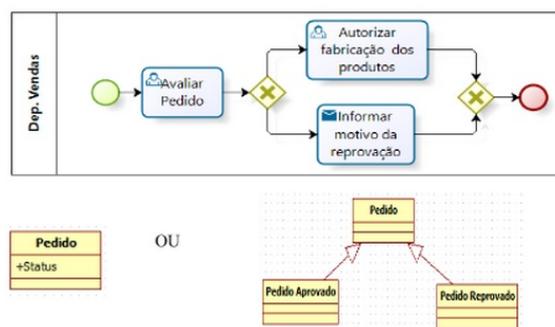


Figura 7. Exemplo de mapeamento de desvio exclusivo em um atributo status ou em um relacionamento de generalização

3.9. Mapeamento de eventos

Este trabalho considera apenas o mapeamento de eventos temporais. Um evento sem tipo definido não tem mapeamento para elemento de diagrama de classes pois não acrescenta nenhuma informação ao diagrama de classes. Demais tipos de eventos serão analisados nos trabalhos futuros. Evento temporal é mapeado para anotação associada à classe obtida

a partir da parte “objeto” do rótulo da atividade que sucede a esse evento temporal. A anotação apresenta informações sobre o evento. A figura 8 apresenta um exemplo desse mapeamento. Um resumo com os passos de mapeamento de processo para diagrama de classes é apresentado na figura 9.

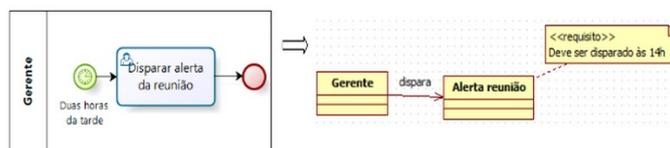


Figura 8. Exemplo de mapeamento de evento temporal para a anotação

Passo	Descrição
1	se houver piscinas, mapear cada uma delas para classes
2	se houver raias, mapear cada uma delas para classes
3	criar relacionamentos de: 3.1. generalização entre classes obtidas a partir de uma piscina e suas respectivas raias quando existir uma relação do tipo "é um" ou "é um tipo de" entre a piscina e as raias, sendo a piscina a classe-mãe e as raias as classes-filhas 3.2. composição entre classes obtidas a partir de uma piscina e suas respectivas raias quando existir uma relação do tipo "todo-parte" entre a piscina e as raias, sendo a piscina a classe-todo e as raias as classes parte
4	mapear cada tarefa de uma piscina (ou raia) para responsabilidades da classe obtida por meio da respectiva piscina (ou raia)
5	se os rótulos das atividades estiverem no formato "verbo (imperativo) + objeto" (como, por exemplo, "Avaliar pedido" ou "Enviar ordem de serviço"), transformar a parte referente ao "objeto" do rótulo em classe. Para cada classe criada dessa forma: 5.1. criar um relacionamento de associação entre a classe obtida por meio do rótulo de atividade e a classe obtida por meio da piscina (ou raia) que contém a respectiva atividade 5.2. nomear esse relacionamento de associação com a parte "verbo (imperativo)" do rótulo 5.3 se houver eventos temporais, mapear cada um deles para anotação associada à classe obtida a partir da parte "objeto" do rótulo da atividade que sucede a esses eventos. Colocar na anotação informações sobre o evento
6	mapear cada desvio exclusivo (ou seja, XOR) para uma dos dois tipos de elemento: 6.1. um atributo (ex.: status) usado para diferenciar o estado de instâncias da classe obtida a partir da parte "objeto" do rótulo da atividade que precede o desvio 6.2. um relacionamento de generalização entre a classe obtida a partir da parte "objeto" do rótulo da atividade que precede o desvio (nesse caso, é considerada como classe mãe) e as classes obtidas a partir dos estados das instâncias dessa classe caso as instâncias possuam mais características do que a classe mãe

Figura 9. Passos para mapeamento entre processo de negócio em BPMN e diagramas de classes da UML

3.10. Exemplo ilustrativo de mapeamento completo das regras propostas

Para ilustrar o uso das regras de mapeamento, são apresentados o modelo BPMN (Figura 10) e a descrição textual de um processo adaptado de Suarez et al. (2008). O processo de vendas de uma fábrica de cadeiras de escritório, cujo modelo é apresentado na figura 10, é descrito da seguinte forma: um cliente acessa um site de uma empresa X especializada na fabricação de cadeiras de escritório; o cliente faz um pedido que deve incluir seu nome, CPF, endereço, produtos desejados e data de solicitação; ao receber o pedido do cliente, o departamento de vendas confere o pedido e inicia o processo enviando ao gerente de catálogo, que é o responsável pela análise do pedido; o departamento de vendas é então responsável por informar ao cliente o resultado final da análise de seu pedido; o gerente de catálogo analisa a viabilidade do pedido: (1) se todos os produtos solicitados estiverem no catálogo, o pedido é aprovado, a fabricação dos produtos é autorizada, e uma ordem

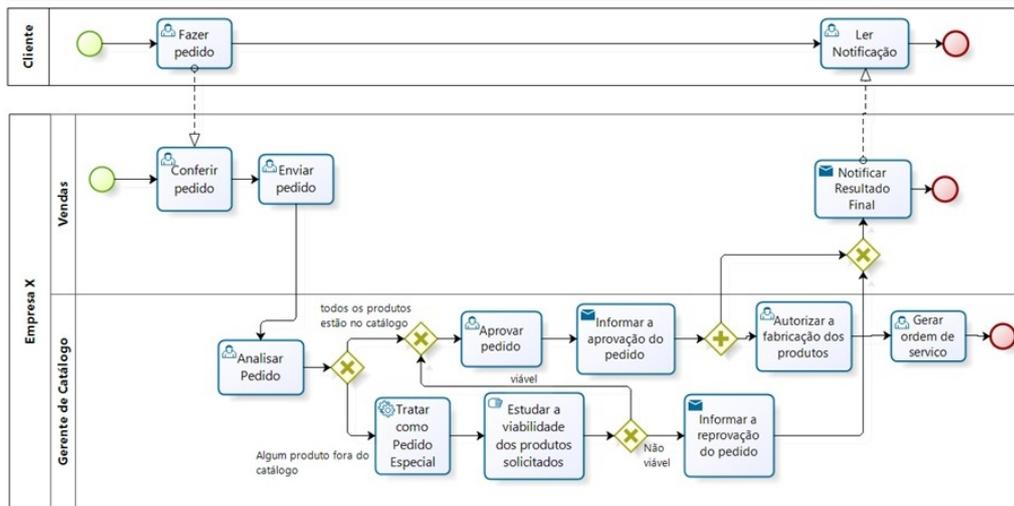


Figura 10. Modelo de processo de fábrica de escritório

de serviço para a fabricação dos produtos é gerada. (2) caso contrário, é considerado como um pedido especial e o gerente de catálogo deve estudar a viabilidade dos produtos solicitados; (2.1) se for viável o pedido é aprovado, a fabricação é autorizada, e é gerada uma ordem de serviço; (2.2) caso contrário, o pedido é reprovado.

Aplicando os passos de mapeamento (Figura 9) no modelo de processo da figura 10, um diagrama de classes contendo oito classes (Figura 11) foi gerado. As classes obti-

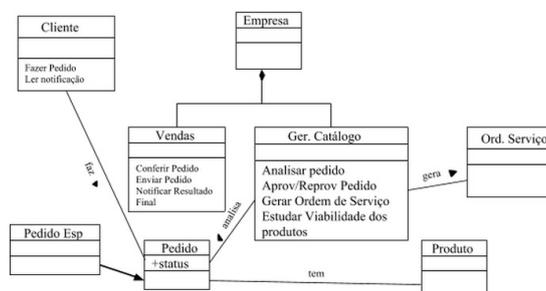


Figura 11. Diagrama de classe correspondente ao processo de solicitação de cadeiras de escritório

das a partir das piscinas e raias são apresentadas com suas respectivas responsabilidades. No diagrama de classes obtido (Figura 11) é possível ver os relacionamentos estabelecidos entre as classes. O relacionamento de composição entre a classe Empresa X, Vendas e Ger. Catálogo foi usado para apresentar que essas duas classes têm sua existência dependente da classe Empresa X. Se esta deixar de existir, outras duas também deixarão de existir. O relacionamento de generalização entre as classes Pedido e Pedido Especial apresenta uma noção de herança entre elas.

4. Experimento para validação do método proposto

Um experimento foi realizado com alunos de graduação e pós-graduação da disciplina de Engenharia de Software do curso de Ciência da Computação de uma universidade pública

federal brasileira com o intuito de avaliar a criação dos diagramas de classes com e sem o uso das regras de mapeamento propostas neste trabalho. Para isso, foi disponibilizado um tutorial que apresenta definições sobre elementos notacionais da BPMN e sobre o diagrama de classes.

Ao todo, 53 alunos foram divididos em dois grupos conforme a disponibilidade de computadores no laboratório onde foi realizado o experimento. O grupo A, com 25 alunos, realizou a atividade sem o uso das regras de mapeamento propostas. Já o grupo B, com 28 alunos, realizou a atividade usando as regras de mapeamento propostas. Para ambos os grupos, foi usado um processo de locação de veículo adaptado de Guedes (2009) como apresentado na figura 12. Um questionário foi aplicado aos alunos via Google forms. A figura 13 apresenta um fragmento de diagrama de classes referente ao modelo de processo de negócio apresentado na figura 12, conforme apresentado por Guedes (2009).

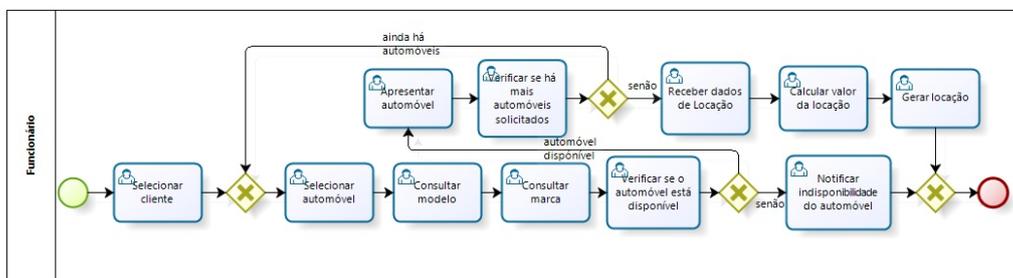


Figura 12. Modelo de processo de locação de veículo

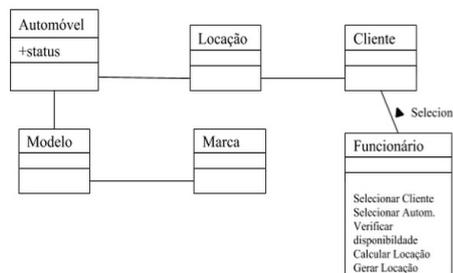


Figura 13. Fragmento de diagrama de classes referente ao modelo de processo de negócio apresentado na figura 12 conforme apresentado por Guedes (2009)

A Figura 14 apresenta os resultados sobre o conhecimento em BPMN e em diagrama de classes para cada grupo. Os resultados apontam que a maioria dos alunos tinham conhecimento nenhum ou básico em BPMN e básico ou intermediário em diagrama de classes. A figura 15 mostra os resultados obtidos após análise dos diagramas de classes extraídos pelos alunos. Dezoito alunos do grupo B, que representa 64 % do grupo, (grupo com as regras de mapeamento) obtiveram um diagrama de classes com um número de seis classes, aproximando-se das características do diagrama de classes apresentado por Guedes (2009) (Figura 13). Além disso, seis alunos do grupo B identificaram mais do que seis classes, o que representam outros 21 % do grupo. Ao total, 86 % dos alunos do grupo B identificaram seis ou mais classes. Por outro lado, para o grupo A, apenas três alunos (ou seja, 12 %) identificaram seis classes, e 20 alunos (ou seja, 80 % do grupo) identificaram menos do que seis classes. A Figura 16 apresenta o resultado de satisfação

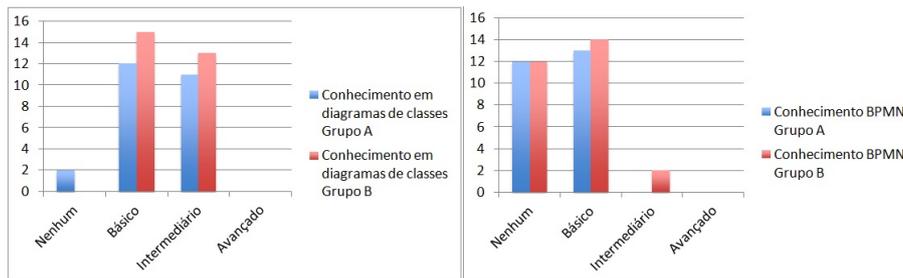


Figura 14. Conhecimento dos alunos em BPMN e em diagrama de classes

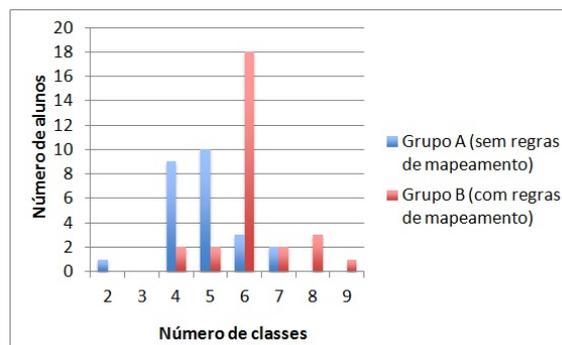


Figura 15. Classes vs número de alunos

dos alunos do grupo B em relação às regras de mapeamento que foram fornecidas. A maioria concordou parcialmente que as regras de mapeamento auxiliam na extração de diagramas de classes a partir de um modelo de processo de negócio. Somente dois alunos não concordaram com o uso das regras.

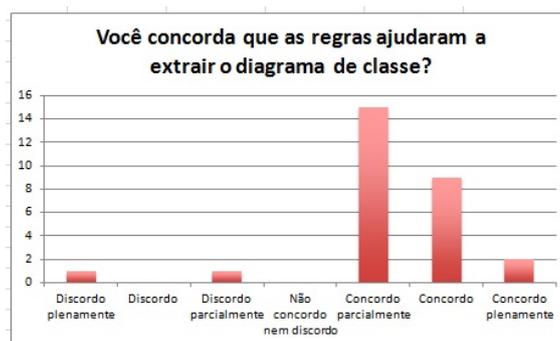


Figura 16. Satisfação dos alunos com as regras de mapeamento

5. Conclusão

As regras de mapeamento de modelos em nível de processo de negócio para modelos em nível de sistemas apresentadas neste trabalho demonstram que é possível obter fragmentos de diagramas de classes a partir de um modelo de processo de negócio em BPMN. Além disso, pelos resultados obtidos, com o experimento de validação realizado, pode-se concluir que as regras auxiliam na extração de tais diagramas de classes. O diagrama de

classes obtido poderá ser uma versão inicial a ser posteriormente melhorado. Os especialistas podem na sequência validar e completar o diagrama de classes obtido. O estudo realizado apresentou limitações quanto à validação dos modelos de processo. Não foi realizada uma verificação explícita do modelo de processo para avaliar sua correteza. Como trabalho futuro, sugere-se a criação de uma ferramenta que auxilie na extração de diagramas de classes com base no algoritmo apresentado na figura 9. Dessa forma, o mapeamento poderia ser realizado de forma semi-automática facilitando cada vez mais a tarefa de análise do projetista.

Referências

- Barros, J. P. and Gomes, L. (2000). From activity diagrams to class diagrams. In *Workshop Dynamic Behaviour in UML Models: Semantic Questions In conjunction with Third International Conference on UML, York, UK*. Citeseer.
- Belaunde, M., Casanave, C., DSouza, D., Duddy, K., El Kaim, W., Kennedy, A., Frank, W., Frankel, D., Hauch, R., Hendryx, S., et al. (2003). Mda guide version 1.0.
- Booch, G., Rumbaugh, J., and Jacobson, I. (1999). Unified modeling language user guide, the.
- Dumas, M., La Rosa, M., Mendling, J., and Reijers, H. A. (2013). *Fundamentals of business process management*. Springer.
- Elmasri, R., Navathe, S. B., Pinheiro, M. G., Canhette, C. C., Melo, G. C. V., Amadeu, C. V., and de Oliveira Morais, R. (2005). Sistemas de banco de dados.
- Force, U. R. T. (1999). Omg uml specification.
- Mendling, J., Reijers, H. A., and Recker, J. (2010). Activity labeling in process modeling: Empirical insights and recommendations. *Information Systems*, 35(4):467–482.
- Okawa, T., Hirabayashi, S., Kaminishi, T., Koizumi, H., and Sawamoto, J. (2007). 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*, pages 458–465. IEEE.
- Recker, J. (2010). Opportunities and constraints: the current struggle with bpmn. *Business Process Management Journal*, 16(1):181–201.
- Rodríguez, A., de Guzmán, I. G.-R., Fernández-Medina, E., and Piattini, M. (2010). Semi-formal transformation of secure business processes into analysis class and use case models: An mda approach. *Information and Software Technology*, 52(9):945–971.
- Specification, B. P. M. N. (2006). Omg final adopted specification. *Object Management Group (February 2006)*. Online available: http://www.omg.org/bpmn/Documents/OMG_Final_Adopted_BPMN_1-0_Spec_06-02-01.pdf (accessed: 23.08.14).
- Suarez, E., Delgado, M., and Vidal, E. (2008). Transformation of a process business model to domain model. In *Proceedings of the World Congress on Engineering*, volume 1.
- Weske, M. (2012). *Business process management: concepts, languages, architectures*. Springer Science & Business Media.