

Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática



Metamodelação de Processos e Serviços

Jorge Manuel da Silva Carneiro Praça Freitas
(Licenciado)

Dissertação apresentada na Faculdade de Ciências e
Tecnologia da Universidade Nova de Lisboa para a obtenção do
Grau de Mestre em Engenharia Informática

Orientador: Professor Doutor Fernando Manuel Pereira da Costa Brito e Abreu

Monte de Caparica

2010

Resumo

Os níveis de qualidade de serviço e respectivos acordos surgiram como uma forma dos departamentos de Tecnologias de Informação e prestadores de serviços de rede medirem e gerirem a qualidade de serviço que se entrega aos clientes. A concorrência no mercado global está a obrigar muitos prestadores ou fornecedores de serviços à diferenciação, em termos de qualidade de serviço, ao estabelecerem os seus contractos. Por outro lado, a necessidade de oferecer serviços mais sofisticados e complexos força os seus fornecedores a participarem em consórcios, onde a subcontratação tem frequentemente lugar. Essas parcerias implicam que todas as partes envolvidas (incluindo clientes) devam compreender e partilhar os mesmos conceitos de serviço de TI, nomeadamente aqueles referentes à qualidade de serviço.

Apesar dos esforços em curso sobre a criação de um corpo de conhecimento em gestão de serviços de TI (Tecnologias de Informação), esses esforços ainda não fornecem uma base suficiente para a definição formal e avaliação de serviços de TI. Com efeito, revendo de forma sistemática o estado da arte, verificou-se em particular que a literatura sobre *Service Level Agreements* (SLA) no domínio genérico de TI era muito limitada, sendo a existente muito focada na gestão de infra-estruturas ou telecomunicações. Para mitigar esta falha, propõe-se nesta dissertação uma ontologia sobre serviços de TI, sobre a qual se define formalmente o conceito de SLA, bem como os conceitos necessários para a verificação do seu cumprimento.

Para modelar a dinâmica da prossecução dos serviços de TI, sobre os quais incidem os SLAs, recorre-se frequentemente à modelação de processos de negócio. Para suportar essa modelação, propõe-se nesta dissertação um metamodelo da linguagem *Business Process Modeling Notation* (BPMN). Esta linguagem permite retractar a colaboração entre todas as partes interessadas e a forma como se desenvolvem os serviços, permitindo assim a modelação da sua dinâmica.

Palavras-Chave: metamodelação de serviços, ontologia, SLA, acordos de nível de serviço, OCL, gestão e qualidade dos serviços de TI, engenharia de processos, engenharia de modelos.

[Esta página foi intencionalmente deixada em branco]

Abstract

Service quality levels and corresponding agreements emerged as a way for Information Technology departments and network service providers to measure and manage the quality of service that is delivered to customers. Competition in the global market is forcing many service providers to differentiate in terms of QoS (Quality of Service), while establishing their contracts. Furthermore, the need to offer more sophisticated and complex services forces their providers to participate in consortia, where outsourcing often takes place. Those partnerships imply that all stakeholders (including customers) should understand and share the same IT service concepts, namely those concerning the QoS.

Despite the ongoing efforts on creating a body of knowledge on IT service management, these efforts have not yet provided a sufficient basis for the formal definition and evaluation of IT services in general. Indeed, while performing a systematic survey of the state of the art, we observed in particular that the literature on *Service Level Agreements* (SLA) in the IT generic domain is very scarce, being the existing one too focused on infrastructure management or telecommunications. To mitigate this limitation, we propose in this dissertation an ontology on IT services, upon which the concept of SLA is formally defined, along with the concepts required for its compliance verification.

To model the dynamics of IT services prosecution, to which SLAs apply, business process modeling approaches are frequently used. To support that modeling, we propose in this dissertation a lightweight metamodel of the BPMN (Business Process Modeling Notation) language. This language allows depicting the collaboration among all stakeholders and the way IT services are rolled out, therefore allowing to model their dynamics.

Keywords: services metamodeling, ontology, SLA, service level agreements, OCL, IT service management and quality, process engineering, model engineering.

[Esta página foi intencionalmente deixada em branco]

Dedicatória

À minha Mãe

[Esta página foi intencionalmente deixada em branco]

Agradecimentos

Embora uma dissertação seja, pela sua finalidade académica, um trabalho individual, recebi contributos de natureza diversa que não podem nem devem deixar de ser realçados. Por outro lado os trabalhos na área das TI são processos de colaboração, quer pela sua extensão e detalhe, quer pela multiplicidade de conhecimentos envolvidos. Por essas razões, desejo expressar os meus sinceros agradecimentos:

Ao meu orientador Professor Fernando Brito e Abreu, pela sua capacidade de trabalho, pelo seu saber em múltiplos campos e valor real que acrescentou a esta obra, também pelo seu encorajamento, apoio e organização;

Ao meu colega do QUASAR Anacleto Correia, pela sua colaboração no desenho do Metamodelo ITSM, e pela sua paciência e amizade;

À minha colega do QUASAR Raquel Porciúncula, que colaborou e deu muitas e boas sugestões para o desenho do Metamodelo BPMN e pela sua ajuda na revisão da dissertação;

Ao meu Colega do QUASAR José Costa que trabalhou com o modelo de BPMN e efectuou um trabalho de validação e pela sua ajuda na revisão da dissertação;

À minha Mãe, ao meu Pai, à minha Avó, ao meu Avô, um obrigado por tudo e muito mais.

Um muito especial obrigado à minha mulher Paula pelo seu apoio nas minhas ausências e por aturar os nossos filhos sozinha.

Monte da Caparica, 30 de Setembro de 2010

Jorge Manuel da Silva Carneiro Praça Freitas

[Esta página foi intencionalmente deixada em branco]

Índice

1. INTRODUÇÃO	2
1.1 CONTEXTO	2
1.1.1 <i>Serviços e Processos</i>	2
1.1.2 <i>Ciência dos Serviços</i>	4
1.1.3 <i>Gestão de Serviços de TI</i>	5
1.1.4 <i>Modelação</i>	7
1.1.5 <i>Níveis de descrição</i>	10
1.2 MOTIVAÇÃO	12
1.2.1 <i>Importância dos SLA para os Serviços</i>	12
1.2.2 <i>Importância da descrição da dinâmica de um sistema</i>	13
1.2.3 <i>Alternativas para modelar a dinâmica de um sistema</i>	14
1.2.4 <i>Modelo de BPMN para a Dinâmica</i>	15
1.2.5 <i>UML para a Dinâmica</i>	15
1.2.6 <i>Meta modelação</i>	18
1.2.7 <i>Avaliação de qualidade de serviço SLA</i>	20
1.3 IDENTIFICAÇÃO DO PROBLEMA	24
1.4 OBJECTIVOS, SOLUÇÃO PROPOSTA E CONTRIBUIÇÕES	25
1.5 METODOLOGIA	26
1.6 CONVENÇÕES TIPOGRÁFICAS E ESTRUTURA DESTA DISSERTAÇÃO	28
2. ESTADO DA ARTE	32
2.1 AS TAXONOMIAS	32
2.2 ESTADO DA ARTE NA REPRESENTAÇÃO DO CONHECIMENTO ITSM	33
2.2.1 <i>Taxonomia ITSM</i>	33
2.2.2 <i>Trabalhos Relacionados em ITSM</i>	37
2.2.3 <i>Análise Comparativa e Lacunas ITSM</i>	49
2.3 ESTADO DA ARTE DOS METAMODELOS BPMN	50
2.3.1 <i>Taxonomia para MetaModelos BPMN</i>	50
2.3.2 <i>Trabalhos Relacionados BPMN</i>	55
2.3.3 <i>Análise comparativa e lacunas metamodelo BPMN</i>	58
3. ONTOLOGIA GERAL DE SERVIÇOS E SLA (ITSM)	62
3.1 INTRODUÇÃO	62
3.2 DESCRIÇÃO DOS PACOTES	62
3.2.1 <i>Pacote de Participantes</i>	63
3.2.2 <i>Pacote de Serviços</i>	67
3.2.3 <i>Pacote de Entregas</i>	69
3.2.4 <i>Pacote de Suporte</i>	71
3.2.5 <i>Pacote de Arquitectura de Empresa</i>	73
3.3 RESTRIÇÕES OCL ONTOLOGIA ITSM	73
3.3.1 <i>Expressões Pacote Participantes</i>	74
3.3.2 <i>Expressões Pacote Serviços</i>	74
3.3.3 <i>Expressões Pacote Entregas</i>	75
3.3.4 <i>Expressões Pacote Suporte</i>	76
3.3.5 <i>Expressões Pacote Enterprise Architecture</i>	76
3.4 VALIDAÇÃO DA CONFORMIDADE DOS SLA	77
3.5 DISCUSSÃO DOS RESULTADOS ONTOLOGIA ITSM	79

4.	REPRESENTAÇÃO DA DINÂMICA.....	82
4.1	INTRODUÇÃO	82
4.2	DESCRIÇÃO DOS PACOTES.....	84
4.2.1	<i>Pacote Modelo.....</i>	85
4.2.2	<i>Pacote Elementos Estruturais.....</i>	88
4.2.3	<i>Pacote Elementos de Suporte.....</i>	90
4.2.4	<i>Pacote Conectores.....</i>	91
4.2.5	<i>Pacote Objectos de Fluxo.....</i>	95
4.2.6	<i>Pacote Actividade.....</i>	96
4.2.7	<i>Pacote Gateway.....</i>	98
4.2.8	<i>Pacote Eventos.....</i>	100
4.2.9	<i>Pacote Artefactos.....</i>	101
4.2.10	<i>Pacote Orquestração.....</i>	104
4.2.11	<i>Pacote Simulação.....</i>	107
4.2.12	<i>Pacote Conexão Interna.....</i>	108
4.2.13	<i>Pacote Casos de Utilização.....</i>	111
4.3	RESTRICÇÕES DO METAMODELO.....	112
4.3.1	<i>Expressões do Pacote Modelo.....</i>	112
4.3.2	<i>Expressões Pacote Elementos Estruturais.....</i>	113
4.3.3	<i>Expressões do Pacote Connectores.....</i>	113
4.3.4	<i>Expressões do Pacote Actividade.....</i>	114
4.3.5	<i>Expressões do Pacote Gateway.....</i>	114
4.3.6	<i>Expressões do Pacote Event.....</i>	115
4.3.7	<i>Expressões Pacote Use Case.....</i>	117
4.4	DISCUSSÃO DOS RESULTADOS METAMODELO BPMN.....	117
4.5	VALIDAÇÃO E CONTEXTO METAMODELO BPMN.....	118
4.6	DISCUSSÃO DOS RESULTADOS RELATIVOS AO METAMODELO DE BPMN.....	121
5.	CONCLUSÕES E TRABALHO FUTURO.....	124
5.1	CONTEXTO.....	124
5.2	COMPARAÇÃO DOS OBJECTIVOS PROPOSTOS E ALCANÇADOS.....	124
5.3	REVISÃO DAS CONTRIBUIÇÕES.....	125
5.4	PUBLICAÇÕES CIENTIFICAS.....	126
5.4.1	AN ONTOLOGY FOR IT SERVICES.....	126
5.4.2	DEFINITION AND VALIDATION OF COMPLEXITY METRICS FOR ITSM PROCESS MODELS.....	127
5.4.3	IDENTIFICAR OS NÍVEIS DE APTIDÃO DA GESTÃO DE SERVIÇOS DE TI USANDO AS MELHORES PRÁTICAS.....	127
5.4.4	MGPSI – METODOLOGIA DE GESTÃO DE PROJECTOS APLICADA AO DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO 128	
5.5	TRABALHO FUTURO.....	128
	BIBLIOGRAFIA.....	129
	A : TEMPLATE DE SLA.....	135
	B : METAMODELO STP_ECLIPSE.....	141
	C: EXTRACTO PARA EXEMPLO DAS ESPECIFICAÇÕES OMG BPMN 1.2.....	143
	D: MÉTRICAS DA GESTÃO DE INCIDENTES E GESTÃO DE PROBLEMAS.....	145
	E: PROCESSO “MONITORIZAÇÃO DE INCIDENTES NO SLA” MODELADO COM BPMN.....	149

Índice de Figuras

Figura 1 - Transacções Internacionais de Serviços nos EUA (Milhões de Dólares). Extraído de [BEA - Bureau Of Economics Analysis, 2009]	3
Figura 2 - Relações entre conceptualização, abstracção, linguagem de modelação, modelo, ontologia e metamodelo. Adaptado de [Guizzardi, 2007].	11
Figura 3 - Exemplo de modelação do processo de “Gestão de Qualidade de Serviço” com um diagrama de casos de utilização	16
Figura 4 – Exemplo de um diagrama de sequência “Criação de Conta” extraído de [Alberto Silva, 2001, Booch, et al., 1998]	17
Figura 5 - Exemplo de um diagrama de estados, extraído de [Booch, et al., 1998].....	17
Figura 6 - Exemplo de diagrama de actividades para modelar um processo	18
Figura 7 - Relacionamento entre ontologia, modelo e metamodelo, extraída de [Saeki and Kaiya, 2006].....	20
Figura 8 - Exemplos de KPIs.....	24
Figura 9 - Nivel 1 eTOM, extraído de [Forum, 2007, Forum, 2006]	46
Figura 10 – SLM (Service Level Management) : Um dos principais processos de ITIL, extraído de [Loyd and Rudd, 2007]	47
Figura 11 - Pacotes ITSM.....	63
Figura 12 - Pacote Participantes ITSM.....	64
Figura 13 – Modelo de objectos de um contracto de SLA	65
Figura 14 - Participantes do fornecedor no SLA	66
Figura 15 - Participantes do cliente no SLA.....	67
Figura 16 - Pacote Serviços ITSM	68
Figura 17 - Características do Serviço.....	69
Figura 18- Pacote de Entregas ITSM	70
Figura 19 - Criação de um contracto de SLA	71
Figura 20 – Escalonamento dos problemas	71
Figura 21 - Pacote de Suporte ITSM	72
Figura 22 - Service Access Point.....	72
Figura 23 - Pacote de Arquitectura de Empresa ITSM	73
Figura 24 - Modelo de conformidade lógica OCL	77
Figura 25 - Valores Observados para o SLA.....	78
Figura 26 - Invariante OCL 12, resultado da validação OK.....	78
Figura 27 - Invariante OCL 12, resultado falhou	79
Figura 28 - Exemplo genérico de processo BPMN	84

Figura 29 - Pacotes do metamodelo de BPMN	84
Figura 30 - Diagrama de classes do pacote <i>Modelo</i>	85
Figura 31 - Diagrama de objectos do modelo “Exemplo”	88
Figura 32 - Pacote de elementos estruturais BPMN.....	88
Figura 33 - Exemplo de <i>Lanes</i> e <i>Pools</i>	90
Figura 34 - Diagrama de objectos com exemplo de Lanes e Pools.....	90
Figura 35 - Elementos de suporte BPMN.....	91
Figura 36 - Pacote de Conectores BPMN.....	91
Figura 37 - Exemplo com “Processo 1” com conectores BPMN (Nível M1).....	93
Figura 38 - Diagrama de objectos do “Processo 1”	94
Figura 39 - Pacote de <i>Flow Objects</i> de BPMN	96
Figura 40 - Pacote Actividade	97
Figura 41 - Tipos de Gateways, extraído de [OMG, 2009a]	99
Figura 42 - Pacote <i>Gateways</i> de BPMN.....	99
Figura 43 - Pacote de Eventos de BPMN	101
Figura 44- Pacote de artefactos BPMN	102
Figura 45 - Exemplo de “Processo 2” para objectos de fluxo e artefactos.....	103
Figura 46 - Exemplo de modelo de objectos do “Processo 2” para objectos de fluxo e artefactos	104
Figura 47 - Pacote de orquestração BPMN	105
Figura 48 - Extracto do processo exemplo relativo à orquestração.....	106
Figura 49 - Modelo de objectos do exemplo de orquestração.....	106
Figura 50 - Pacote de simulação BPMN	107
Figura 51 - Ligações de elementos BPMN na <i>pool</i>	108
Figura 52 - Ligações das <i>text annotation</i>	108
Figura 53 - Exemplo de Ligação Interna com Sequence Flow S1	109
Figura 54 - Exemplo de Ligação Interna Com Associações	109
Figura 55 - Modelo de objectos exemplo de ligação interna.....	110
Figura 56 - Modelo de objectos com exemplos de associações	110
Figura 57 - Diagrama do pacote Casos de utilização	111
Figura 58 - Processo de Modelação, Transformação e Instanciação do Metamodelo [Costa, 2010]	118
Figura 59 - Exemplo de processo MGPSI [Costa, 2010]	119
Figura 60 - Meta-objectos carregados MGPSI.....	120
Figura 61 - Colaboração dissertações [Porciúncula, 2010]	120

Índice de Tabelas

Tabela 1 - Dimensões de Maturidade e Níveis Correspondentes	36
Tabela 2 - Classificação de Software Maintenance from a Service Perspective.....	38
Tabela 3 - Classificação de <i>Towards generic service management concepts - MNM Model</i> ..	39
Tabela 4 - Classificação de <i>Precise Service Level Agreements</i>	40
Tabela 5 - Classificação de <i>UML Profile for QOS</i>	41
Tabela 6 - Classificação de <i>UML- Contract Negotiation and Service Level Agreements</i>	43
Tabela 7 - Classificação de Specifying Information Security	44
Tabela 8 - Classificação de <i>Enhanced Telecom Operations Map (eTOM)</i>	47
Tabela 9 - Classificação de ITIL	49
Tabela 10 - Estudo comparativo de propostas ITSM	49
Tabela 11 - Comparação de metamodelos.....	52
Tabela 12 - Tabela Resumo taxonomia BPMN.....	55
Tabela 13 - Tabela de classificação metamodelo STP_Eclipse BPMN	57
Tabela 14 - Tabela de classificação metamodelo OMG BPMN 1.2	58
Tabela 15 - Estudo comparativo de metamodelos BPMN	58
Tabela 16 – Valores observados e permitidos no SLA para o exemplo “em conformidade” ..	78
Tabela 17 - Valores observados e permitidos no SLA para o exemplo “não conforme”	79
Tabela 18 - Pools e Lanes (BPMN) extraído de [OMG, 2009b].....	89
Tabela 19 - Conectores (BPMN) extraído de [OMG, 2009b]	92
Tabela 20 - Objectos de Fluxo extraído de [OMG, 2009b].....	95
Tabela 21 - Artefactos (BPMN) extraído de [OMG, 2009b]	102
Tabela 22 – Objectivos propostos e sua prossecução.....	125

[Esta página foi intencionalmente deixada em branco]

Abreviaturas

ACM	<i>Association for Computing Machinery</i>
BPEL	<i>Business Process Execution Language</i>
BPM	<i>Business Process Management</i>
BPMI	<i>Business Process Management Initiative</i>
BPMN	<i>Business Process Modeling Notation</i>
BPMN-WG	<i>Business Process Modeling Notation Working Group</i>
BPML	<i>Business Process Modeling Language</i>
BSC	<i>Balanced Scorecard</i>
CIFS	<i>Critical Impact Factors</i>
COBIT	<i>Control Objectives for information Technologies</i>
CML	<i>Conceptual Modeling Language</i>
CMMI	<i>Capability Maturity Model Integration</i>
CRM	<i>Customer Relationship Management</i>
DSL	<i>Domain Specific Language</i>
EMF	<i>Eclipse Modeling Framework</i>
eTOM	<i>enhanced Telecom Operations Map</i>
FCT/UNL	Faculdade de Ciências e Tecnologia / Universidade Nova de Lisboa.
IBM	<i>International Business Machines</i>
ISO	<i>International Organization for Standardization</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ITIL	<i>Information Technology Infrastructure Library</i>
ITSM	<i>Information Technology Service Management</i>
KPI	<i>Key Performance Indicator</i>
KIF	<i>Knowledge Interchange Format</i>
MDA	<i>Model Driven Architecture</i>
MDE	<i>Model-driven Engineering</i>
MDT	<i>Model Development Tools</i>
MOF	<i>Meta-Object Facility</i>
OCL	<i>Object Constraint Language</i>
OLA	<i>Operational Level Agreement</i>
OMG	<i>Object Management Group</i>
OWL	<i>Web Ontology Language</i>
OMG	<i>Object Management Group</i>
QOS	<i>Quality of Service</i>
PIM -	<i>Platform Independent Model</i>
PMBOK	<i>Project Management Body of Knowledge</i>
QUASAR	<i>Quantitative Approaches on Software Engineering And Reengineering</i>
SAP	<i>Business Management Software Solutions Applications and Services</i>
SLA	<i>Service Level Agreement</i>
SLM	<i>Service Level Management</i>
SQL	<i>Structured Query Language</i>
SOA	<i>Service-oriented Architecture</i>
SPEM	<i>Software Process Engineering Meta-Model</i>
STP	<i>SOA Tools Platform</i>
TI	Tecnologias de Informação
TMForum	<i>TeleManagement Forum</i>
UML	<i>Unified Modeling Language</i>
USE	<i>UML based Specification Environment</i>
WEB	<i>World Wide Web</i>

[Esta página foi intencionalmente deixada em branco]

Introdução

1. INTRODUÇÃO	2
1.1 CONTEXTO.....	2
1.1.1 <i>Serviços e Processos</i>	2
1.1.2 <i>Ciência dos Serviços</i>	4
1.1.3 <i>Gestão de Serviços de TI</i>	5
1.1.4 <i>Modelação</i>	7
1.1.5 <i>Níveis de descrição</i>	10
1.2 MOTIVAÇÃO	12
1.2.1 <i>Importância dos SLA para os Serviços</i>	12
1.2.2 <i>Importância da descrição da dinâmica de um sistema</i>	13
1.2.3 <i>Alternativas para modelar a dinâmica de um sistema</i>	14
1.2.4 <i>Modelo de BPMN para a Dinâmica</i>	15
1.2.5 <i>UML para a Dinâmica</i>	15
1.2.6 <i>Meta modelação</i>	18
1.2.7 <i>Avaliação de qualidade de serviço SLA</i>	20
1.3 DESCRIÇÃO DO PROBLEMA	24
1.4 SOLUÇÃO PROPOSTA, CONTRIBUIÇÕES E OBJECTIVOS.....	25
1.5 METODOLOGIA.....	26
1.6 CONVENÇÕES TIPOGRÁFICAS E ESTRUTURA DESTA DISSERTAÇÃO	28

Este capítulo introduz os principais conceitos utilizados nesta dissertação e refere a motivação desta dissertação. Ele também enumera as principais contribuições e apresenta um breve resumo de cada um dos capítulos restantes.

1. Introdução

Esta dissertação aborda a Modelação de Serviços e SLA (*Service Level Agreement*) num contexto da “Ciência dos Serviços” e ITSM. A importância do tema relaciona-se com a competitividade das empresas de base tecnológica ou que dependem da tecnologia da Informação para o seu funcionamento e que são forçadas a melhorar o seu desempenho e eficiência na gestão, produção e entrega dos seus serviços. Um metamodelo na nossa opinião permite a definição formal dos sistemas e seus comportamentos que por sua vez permite estudos e observações de natureza quantitativa.

A motivação para a escolha do tema encontra-se na importância dos serviços e no facto se encontrar múltiplos trabalhos de modelação de serviços, dispersos e em abordagens específicas como serviços Web (*World Wide Web*) ou abordagens SOA (*Service-Oriented Architecture*), orientados na maior parte para arquitecturas de software TI. Nesta dissertação uma ontologia é criada para representar serviços e seus intervenientes até chegar à informação, nomeadamente informação que permita a representação do SLA. O metamodelo é dividido em sub-modelos que representam áreas distintas, mas mantém os inter-relacionamentos.

"Meta" pode significar objectivo ou designar aquilo que está além, ou mudança, passagem de um estado para outro (como em metamorfose). No decurso do nosso trabalho sentimos necessidade de representar essa mudança, a lidar com serviços e SLA lidamos com tarefas, uma sequência de actos e um ou vários resultados, ou seja processos, assim criamos um metamodelo para representar esses processos. Na secção 1.1.1 do presente capítulo definimos serviço e processo para fornecer o contexto, no capítulo 3 detalhamos os serviços com a explicação da ontologia proposta e no capítulo 4 explicamos as componentes dos processos detalhadamente ao explicar o metamodelo.

1.1 Contexto

1.1.1 Serviços e Processos

A sociedade actual é acompanhada com o progressivo crescimento do sector dos serviços, no contexto das economias mais desenvolvidas.

Com o incremento dos serviços, tal como se mostra na **Figura 1**, com a globalização e a concorrência acrescida, há uma procura por parte das universidades, indústria e governos em tornar mais sistemática a inovação no sector de serviços, na maioria dos países industrializados, e os serviços estão rapidamente a tornar-se o maior sector também em nações em desenvolvimento.

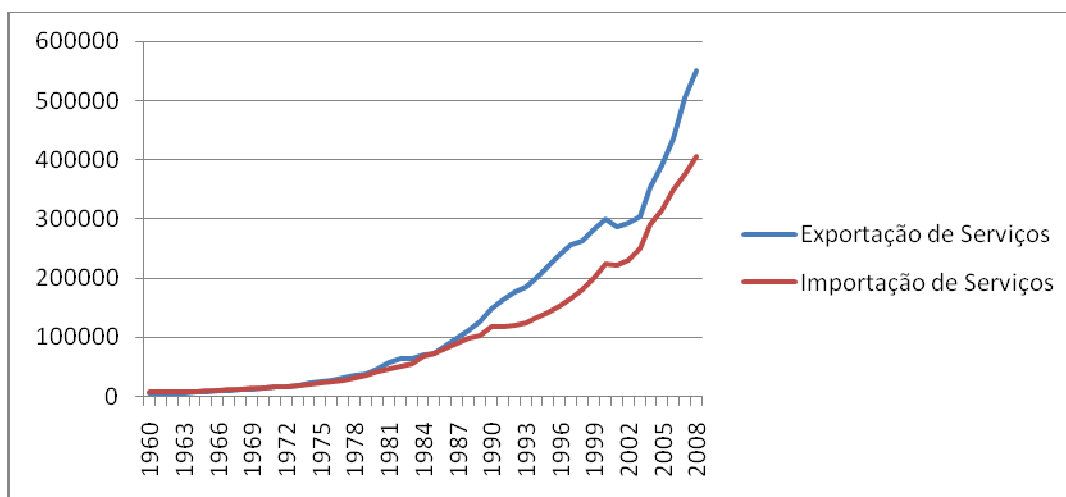


Figura 1 - Transacções Internacionais de Serviços nos EUA (Milhões de Dólares). Extraído de [BEA - Bureau Of Economics Analysis, 2009]

A sociedade contemporânea tem evoluído com base na informação, tendo sido designada por Sociedade da Informação. “Este é um modelo que assenta num modo de desenvolvimento social e económico onde a informação é um meio criação de conhecimento, desempenha um papel fundamental na produção de riqueza e na contribuição para o bem-estar e qualidade de vida dos cidadãos” [Wikipedia, 2009a].

A Sociedade do Conhecimento constitui uma evolução natural da Sociedade da Informação. O termo “Sociedade do Conhecimento” deve-se a Peter Drucker e corresponde a uma etapa no desenvolvimento da civilização que se caracteriza por uma proporção alta de trabalhadores do conhecimento (profissionais que criam, modificam e ou sintetizam conhecimento como parte integrante das suas ocupações) e onde a educação constitui a pedra angular da sociedade [Drucker, 2003].

O conceito de serviço foi definido de formas diferentes no campo de marketing, operações, economia e tecnologias de informação e não existe uma definição comum [Arun Rai and Sambamurthy, 2006]. Alguns exemplos;

- I. Os serviços são capacidades ou competências que um pessoa, organização, empresa ou sistema fornece por outro [Stephen L. Vargo and Lusch, 2004].

- II. Um serviço é uma mudança na condição de uma pessoa, ou um bem que pertence a uma entidade económica, trouxeram um resultado de alguma outra entidade económica, com a aprovação da primeira pessoa ou entidade económica [Hill, 2005].
- III. Um serviço é uma interação cliente-fornecedor que cria e captura de valor [IBM, 2004].

Na nossa opinião, as definições de Serviço enfatizam, a transformação da experiência, o valor, uma relação com os fornecedores, e uma fronteira difícil de perceber entre bens tangíveis e intangíveis.

Ao longo da história, cada sociedade encontrou seu modo específico de produzir serviços ou bens tangíveis. Adam Smith (1723-1790) já abordava a divisão de tarefas [Wikipedia, 2007]. Taylor (1856-1915) estudou métodos para produzir melhor, analisando as tarefas, separando e encadeando-as [Wikipedia, 2006].

A noção de processo é mais recente; algumas definições são;

- I. Qualquer actividade ou conjunto de actividades que toma um *input*, adiciona valor a ele e fornece um *output* a um cliente específico. Os processos utilizam os recursos da organização para oferecer resultados objectivos aos seus clientes [Harrington, 1991].
- II. Um processo é um grupo de actividades realizadas numa sequência lógica com o objectivo de produzir um bem ou um serviço que tem valor para um grupo específico de clientes [Michael Hammer and Champy, 1994].

1.1.2 Ciência dos Serviços

A Ciência dos Serviços é uma nova abordagem interdisciplinar para o estudo, concepção e implementação de sistemas de serviços. Estes são sistemas complexos em que uma combinação de pessoas e tecnologias criam valor [Abe, 2005].

O que caracteriza a área de Ciência dos Serviços é uma maior colaboração interdisciplinar e o foco na inovação e nos serviços. Já se trabalha há alguns anos de forma multidisciplinar, em especial nas operações comerciais, marketing e recursos humanos. Hoje,

alarga-se de forma a tornar ainda mais transversal a multidisciplinaridade que abrange as ciências sociais, a engenharia e a informática [Bitner, et al., 2008]. A chave para a Ciência de Serviços é a abordagem multidisciplinar e considerá-la como um sistema de interacção onde se incluem pessoas, tecnologia e negócios.

Os esforços de investigação em Ciência dos Serviços foram liderados por empresas como a IBM e a Fujitsu [Research, 2005] [Abe, 2005]. Também académicos e entidades governamentais também se interessam por esta ciência. Segundo os mesmos autores existe uma consciência crescente de que a economia e a competitividade, na era do conhecimento e da globalização, estão associadas ao:

- (i) Uso de tecnologias;
- (ii) Interligação com modelos de negócio;
- (iii) Formação;
- (iv) Rápida adaptação a novos produtos;
- (v) Serviços;
- (vi) Concorrentes;
- (vii) Ciclos económicos.

Por estes motivos, existe a necessidade de se concentrarem esforços sobre a inovação dos serviços. Para o efeito, espera-se que a investigação crie novos modelos para impulsionar a inovação dos serviços [Research, 2005].

A Ciência de Serviços baseia-se nas disciplinas já existentes, incluindo Gestão, Ciência da Computação, Ciência Cognitiva, Economia, Gestão de Recursos Humanos, Marketing, Investigação Operacional e outras, visando integrá-las num conjunto coerente [Research, 2005].

1.1.3 Gestão de Serviços de TI

As TI têm como objectivo suprir as necessidades da corporação gerando satisfação interna e externa, criando, gerindo e mantendo sistemas, redes, bases de dados, discos, *outsourcing*, e-mail, acesso à internet, etc.

Com os aspectos técnicos é intrinsecamente gerido um orçamento e como as TI são um factor de produção, este orçamento é normalmente visto como um custo pela organização.

Quando se fala em diminuir os custos, fala-se na necessidade de justificá-los [Mendes and Mira da Silva, 2009], assim como eliminação de erros e de desperdícios. Esta redução é contrariada, normalmente, pelo argumento dos executivos das TI de que tudo é necessário, importante para o negócio e que uma redução de custos poderá ser danosa, gerando perdas financeiras, insatisfação interna e externa.

Procura-se também saber se o orçamento existente é bem aplicado, ou seja, se o dinheiro que se gasta compra a melhor TI que se pode comprar. Para suportar esta afirmação as TI mostram indicadores de disponibilidade de servidores, redes e hiperligações, tempos de resposta de sistemas, quantidade de incidentes, tempo médio de resolução e índices de actualização tecnológica.

O problema neste tipo de relatórios é quem pergunta não está muito interessado em saber se os servidores são bons ou não. De facto, o que se procura saber é qual o valor que as TI acrescentam aos negócios nesse sentido. Pode-se discutir o valor acrescentado pela TI ao negócio através dos serviços que são efectivamente fornecidos [Mendes and Mira da Silva, 2009].

Para se organizar e oferecer os serviços das TI às organizações existem diversas metodologias ou *frameworks*, cada uma com focos diferentes, tais como CobiT (*Control Objectives for information Technologies*), CMMI (*Capability Maturity Model Integration*), ISO¹ 9001 (*Quality Management System*), PMBOK (*Project Management Body of Knowledge*), BSC (*Balanced Scorecard*), ISO 17799 (*IT Security Techniques*) e Six Sigma². No entanto, uma das mais utilizadas é o ITIL (*Information Technology Infrastructure Library*). O seu objectivo é, simplesmente, a implementação e gestão de serviços de TI utilizando uma abordagem da organização por processos. Esta abordagem é importante, porque a percepção do que é a qualidade em serviços não depende só do resultado final mas também da forma como o serviço é produzido [Carter-Stell, et al., 2006].

Identificar os serviços de TI e quantificar o seu valor não é fácil, pois normalmente os utilizadores da área de negócio não sabem o que necessitam ou usam e, muito menos, o valor que acrescentam ao negócio.

¹ ISO (International *Organization for Standardization*) - Organização Internacional de Normalização.

² Six Sigma é um conjunto de práticas originalmente desenvolvidas pela Motorola para melhorar sistematicamente os processos, produtos e serviços.

Entendemos que gestão de serviços de TI é a organização de um conjunto de capacidades e o seu controlo de forma a proporcionar valor aos clientes e a demonstrá-lo.

As capacidades assumem a forma de funções e processos organizadas na forma de serviços. O acto de transformar recursos em serviços que acrescentam valor está no centro da gestão dos serviços. Sem este saber, uma organização de serviços é apenas um conjunto de recursos que, por si, só tem valor relativamente baixo para os clientes.

Para gerir os serviços ao longo de um ciclo de vida, são necessárias competências em estratégia, desenho, transição, operação e melhoria contínua.

A implementação dos processos, por exemplo com ITIL, começa normalmente com a elaboração de um catálogo de serviços, discute-se o valor de cada serviço, definem-se os níveis de serviço e, por fim, o custo e o investimento a serem realizados em cada serviço. Depois, definem-se os processos até um nível de maturidade e detalhe adequado, inclusive com a nomeação de responsáveis pelos processos.

Uma vez definidos os serviços, os processos, os níveis de serviço adequados e os custos é elaborado um acordo de nível de serviço (SLA), elemento central na gestão dos serviços.

1.1.4 Modelação

Independentemente de qual a tecnologia que irá ser utilizada, uma das grandes perguntas que fazemos ao criar serviços é o que realmente devemos disponibilizar e como devemos organizar e implementar o que queremos.

Segundo Rumbaugh [Rumbaugh, et al., 1991], um modelo é uma abstracção de alguma coisa, cujo propósito é permitir que se conheça essa coisa antes de construí-la. Como um modelo omite os detalhes não essenciais, a sua manipulação é mais fácil do que a da entidade original. *“Um modelo é uma simplificação da realidade”* [Booch, et al., 1998].

Construímos modelos de todos os tipos de projectos em Engenharia e Arquitectura, casas, prédios, cadeiras, etc., para auxiliar os seus utilizadores a perceberem qual será o produto final.

A modelação não se restringe a grandes sistemas. Porém, é verdade que, quanto maior e mais complexo for o sistema, maior será a importância da modelação, por uma razão muito simples: Conforme refere Rumbaugh, nós construímos modelos de sistemas complexos porque não é possível compreendê-los na sua totalidade.

Com a modelação alcançamos vários objectivos: Visualização do sistema como ele é ou como desejamos que seja, Especificação da estrutura e do comportamento de um sistema; Guião para a construção do sistema; Registo e documentação das decisões tomadas; Comunicação entre pessoas; Aprendizagem; Persuasão; Análise de situações; Gestão de conformidade; Desenvolvimento de requisitos de *software*; Execução directa dos modelos através de mecanismos de *software*; Gestão do conhecimento; Reutilização; Simulação; Experimentação de novas ideias.

Os “*modelos ajudam na implementação de mudanças. Se nada muda, não são precisos modelos, assim como, não são precisos mapas se não se pretende viajar para lugar nenhum*” [David and Zahavi, 2009] .

Os processos que utilizamos para os modelos consiste numa combinação de técnicas de decomposição do domínio em análise *top-down*, *bottom-up* e *middle-out*, análise de trabalhos existentes e, por fim, meta-modelação.

Um outro importante aspecto dos modelos é que uma mesma realidade pode ser modelada de diferentes maneiras, representando diferentes aspectos do problema, daí apresentarmos um modelo estático e um modelo dinâmico.

O processo *top-down* é muitas vezes referido como a decomposição de domínio, que consiste na divisão do processo de negócio nas suas áreas funcionais e subsistemas, incluindo o seu fluxo de decomposição de processos, sub-processos e de alto nível, nos casos de utilização de negócios. Os casos de utilização (UML) são candidatos muito bons para uma visão de alto nível dos serviços e dos seus componentes.

A abordagem *bottom-up* utiliza-se para a análise dos sistemas que já existem, por exemplo aplicações informáticas.A análise *middle-out* permite capturar aspectos não identificados por outras aproximações. Além disso, permite efectuar uma validação, dado que o seu uso em conjunto com outras abordagens constitui uma reflexão e permite efectuar a ligação entre serviços, objectivos e métricas [Arsanjani, 2004].

O Cobit, por exemplo, descreve as especificações dos sistemas de forma informal com a língua natural (por exemplo, Inglês) com diagramas e com outras notações (por exemplo, caixas), para as quais não existe uma forma e significados bem definidos.

Uma especificação deve ser precisa, não podendo dar margem a interpretações erradas. Em [Meyer, 1985] o autor aponta os sete pecados do especificador: redundância, omissão, sobre-especificação, inconsistência, ambiguidade, referência antecipada e descrição subjectiva. As linguagens informais são ambíguas, o que dá margem a uma

interpretação errónea da especificação e, conseqüentemente, uma implementação errónea do objecto especificado. Uma linguagem formal possui uma forma (sintaxe) e significado (semântica) bem definidos.

O uso de métodos formais ou semi-formais no desenvolvimento de *software* apresenta várias vantagens, salientando-se a possibilidade de prototipagem. Os protótipos podem ser gerados automaticamente e formalmente a partir das suas especificações. Tais protótipos, algumas vezes chamados de especificações executáveis, servem de base para exibir as funcionalidades do futuro sistema. Pode-se provar também que programas satisfazem determinadas propriedades e que um programa é uma realização da sua especificação [Moura, 1995]. Estes métodos formais permitem obter a segurança de ausências de erros no *software* e são baseados na semântica dos programas, ou seja, sobre descrições matemáticas formais do sentido de um programa dado pelo seu código fonte (ou, às vezes, o seu código objecto). Contudo, são geralmente dispendiosos em recursos (humanos e materiais) e actualmente reservados aos *softwares* mais críticos. A sua melhoria e o alargamento dos seus âmbitos de aplicação prática são a motivação de numerosas investigações científicas em informática.

Em conclusão, para modelar SLAs e os próprios serviços alvo precisamos de uma semântica precisa. Metodologias como Cobit dizem-nos o que é necessário, mas não a natureza processual (modelada na dinâmica) e mesmo na modelação da parte estática, são imprecisos, ou seja, fornecem diagramas de modelos mas a semântica é imprecisa. Exemplos de imprecisão são caixas cujo significado não é normalizado e compreendido por todos, ou descrições em linguagem natural em vez de fórmulas precisas. Se queremos dar um passo em frente no âmbito da avaliação de serviços, precisamos de uma semântica precisa. Dado que não encontramos nenhuma definição de serviços de TI genérica, elaborámos uma ontologia de serviços, usando para tal uma linguagem bem definida para clarificar conceitos (UML), que é aceite tanto pela comunidade científica como empresarial, e que está formalmente definida através de uma norma do OMG [OMG, 2009b].

1.1.5 Níveis de descrição

Em filosofia, o termo ontologia tem sido usado desde o Século XVII para se referir tanto a uma disciplina filosófica (Ontologia com um "O" maiúsculo), como a conceptualização de um determinado domínio científico.

Nas últimas décadas tem havido um crescente interesse pelo tema da ontologia em informática e ciências da informação. Nos últimos anos, esse interesse tem aumentado consideravelmente no contexto das iniciativas designadas por *Semantic Web* [Wikipedia, 2009c] e MDA (*Model-Driven Architecture*) [Guizzardi, 2007].

Uma ontologia pode ser vista como um sistema de categorias (o estudo das características mais gerais da realidade e objectos reais) e é independente da forma de especificação. Normalmente, em sistemas de informação é um artefacto de engenharia projectado para uma finalidade específica e representado em um idioma específico. Muitas linguagens têm sido usadas para representar ontologias em domínios específicos. Exemplos incluem cálculo de predicados [Guizzardi, 2007], KIF (*Knowledge Interchange Format*), UML, LINGO [Ricardo A. Falbo, 2002], CML (*Conceptual Modelling Language*), OWL (*Web Ontology Language*). A **Figura 2** descreve a relação entre ontologia e metamodelo. Uma ontologia é usada para descrever conceptualizações, tal como um metamodelo é usado para descrever linguagens de modelação. As conceptualizações descritas pelas ontologias permitem construir abstracções, enquanto as linguagens de modelação descritas pelos metamodelos permitem construir modelos.

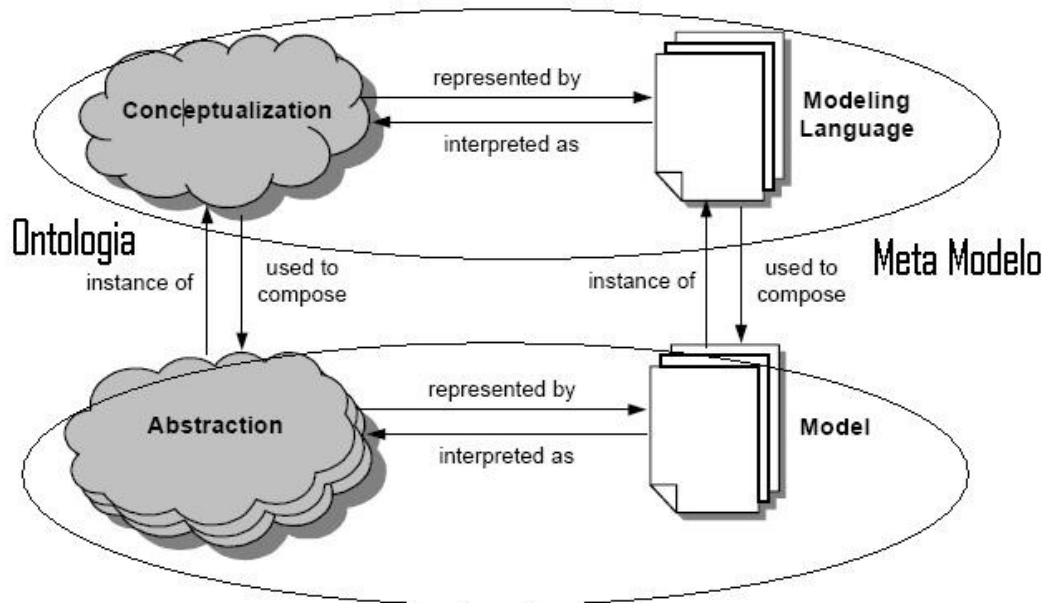


Figura 2 - Relações entre conceptualização, abstracção, linguagem de modelação, modelo, ontologia e metamodelo. Adaptado de [Guizzardi, 2007].

O conjunto de símbolos que compõem uma linguagem, bem como as regras para a formação de combinações válidas dos símbolos, constituem a sintaxe da linguagem. A sintaxe é definida, pela primeira vez, em termos de um conjunto de caracteres (alfabeto) que podem ser agrupados em sequências válidas formando palavras. Isso é chamado de léxico.

A camada do léxico é geralmente definida usando expressões. As palavras podem ser agrupadas em frases, de acordo com as regras definidas com precisão, existentes numa gramática que pode ser escrita num metamodelo.

Finalmente, o metamodelo da linguagem é enriquecido com condições de contexto que se destinam a restringir a sintaxe da linguagem, com o objectivo de criar sentenças bem formadas. Essas restrições são criadas em uma linguagem como seja, por exemplo, OCL (*Object Constraint Language*).

Estas restrições são motivadas por considerações de ordem semântica (leis do domínio a ser modelado), enquanto outras podem ser motivadas por questões pragmáticas. No entanto, um metamodelo é uma descrição da sintaxe abstracta da linguagem, uma vez que define:

(i) Um conjunto de construções seleccionadas para realizar um conjunto de tarefas específicas;

(ii) Um conjunto de regras de boa formação para combinar essas construções a fim de criar modelos gramaticalmente válidos na linguagem.

1.2 Motivação

1.2.1 Importância dos SLA para os Serviços

Cada vez mais as empresas estão a contratar serviços baseados em SLA [Hiles, 2002], [Masche, et al., 2006], [Wetzstein, et al., 2008], alterando-se, assim, o relacionamento entre clientes e fornecedores, o que está a acontecer desde há 10 anos, devido à crescente onda de *outsourcing* [Cavalcanti, et al., 2007].

Para gerir esses contractos é necessário que, tanto os prestadores de serviços, como as empresas contratantes, possuam uma forma de gerir os parâmetros dos mesmos. A simplicidade e conhecimento pleno dos parâmetros contratados são fundamentais para a boa gestão do contrato, evitando conflitos pessoais e contratuais entre empresas e fornecedores dos serviços [Fagundes, 2004].

A elaboração de um SLA deve iniciar-se com a identificação dos requisitos do negócio da empresa em que o serviço irá ser executado. Os interessados pelo serviço, normalmente os utilizadores finais da empresa, devem compreender, opinar e concordar com os parâmetros definidos, devem definir os objectivos e avaliar as variáveis financeiras e específicas do negócio (prazo, qualidade, etc.).

Os tipos de contractos dependem muito da organização existente, da dimensão da empresa, do processo produtivo e dos objectivos do cliente. Tais factores deverão ser objecto de análise específica para cada caso, de modo a garantir que os objectivos pretendidos são alcançados.

Deve verificar-se a real necessidade dos parâmetros contratados, pois existe a tendência de subestimá-los para reduzir os custos ou sobrestimá-los para garantir um alto nível de serviço, com custos altos e desnecessários. Um SLA bem dimensionado significa garantir custos adequados para o negócio da empresa e para o contrato de serviços.

A gestão dos SLA, conhecida como SLM (*Service Level Management*), é realizada através do acompanhamento dos parâmetros do contrato. Esse acompanhamento exige o

uso de ferramentas de *software* que registem a abertura dos chamados *trouble-tickets*³ no *service desk*, que regista tempos de paralisação de equipamentos e outros eventos que tornam indisponível o serviço. Além desses parâmetros técnicos, outros aspectos do serviço do fornecedor devem ser acompanhados para garantir a continuidade do contrato e evitar conflitos entre as partes, tais como: nível de satisfação geral do cliente com o fornecedor e qualidade da gestão do serviço.

Um contrato de SLA inclui informações sobre: a definição dos serviços, performance, gestão de conflitos, responsabilidade de ambas as partes, garantias, planos alternativos, planos para soluções temporárias, relatórios de monitorização, segurança, confidencialidade e cancelamento do contrato. Caso o SLA não seja cumprido, um cliente pode cobrar multas ao fornecedor do serviço, se estes estiverem definidos contratualmente na secção de gestão de conflitos. Ver o anexo A : *Template* de SLA .Um SLA é uma forma de garantir a prestação de serviços de uma forma transparente e verificável, quer por fornecedores, quer por clientes.

1.2.2 Importância da descrição da dinâmica de um sistema

Uma importante dimensão dos sistemas trata da sua relação com o tempo. Modelos que se modificam (evoluem) com o tempo são conhecidos como dinâmicos.Os Processos de negócio são normalmente associados aos objectivos operacionais e relação de negócio (colaboração) entre entidades. Como exemplo, consideremos o processo de realização de um seguro ou processo de fornecimento de um bem. Pode estar contido somente dentro de uma organização, ou contemplar diversas organizações, como no caso dos relacionamentos entre clientes e fornecedores.

Os processos de negócio, definidos em ITIL [Loyd, 2007] representam um conjunto de um ou mais procedimentos ou actividades que colectivamente realizam objectivos, normalmente no contexto de uma estrutura organizacional, definindo papéis funcionais e relacionamentos.

³ *Trouble-Ticket* (“nota de problema” ou “relatório de problemas”) é um mecanismo usado numa organização para acompanhar a detecção, comunicação e resolução de algum tipo de problema. Normalmente é implementado em *software* de *Help-Desk/Service Desk* ou CRM.

Pode-se perceber da necessidade de se modelar os processos de uma empresa para podermos controlar e colaborar na hora em que for necessária uma intervenção. Um SLA é, por definição, um contracto que implica controlo e colaboração.

“A escolha dos modelos a serem criados tem profunda influência sobre a maneira como um determinado problema é atacado e como uma solução é definida.” [Booch, et al., 1998]

Por outras palavras, uma boa escolha do modelo permitirá conclusões que, simplesmente, não seriam possíveis com outros modelos que desviam a atenção para questões a que não se pretende dar relevância, ou que criam confusão.

Modelar a dinâmica significa ter a possibilidade de se criarem processos relacionados com os SLA e de poder geri-los, já que grande parte dos processos de SLA são exactamente isso: processos de negócio envolvendo geralmente mais do que uma parte, com uma noção temporal de casualidade e resposta, por exemplo, nos *help-desks* já referidos.

1.2.3 Alternativas para modelar a dinâmica de um sistema

Existem diversas técnicas e notações de modelação de processos, tais como:

- (i) UML;
- (ii) BPMN;
- (iii) UML + EPBE (Eriksson-Penker Business Extensions) [Penker and Eriksson, 2000];
- (iv) Learn [Coelho, 2003].

Sem considerar outras alternativas vamos fazer uma comparação entre os modelos BPMN e UML (*Unified Modeling Language*), tendo como principal objectivo expandir o nosso conhecimento, explicar as suas vantagens, diferenças e outros aspectos relevantes.

O motivo que nos leva a considerar o BPMN é o suporte da OMG (*Object Management Group*) ao mesmo e a possibilidade de se usar uma modelação formal. A comparação com UML é efectuada para efeitos de referencial, dada a sua elevada utilização, tanto na indústria, como no meio académico.

1.2.4 Modelo de BPMN para a Dinâmica

O BPMN é, na sua origem, uma notação gráfica com o objectivo de criar padrões e uma arquitectura comum para gerir processos de negócio, foi criada pela BPMI (*Business Process Management Initiative*)⁴, uma organização sem fins lucrativos, iniciada pela *Intalio Inc.* em 2000 e que recebeu o suporte de gigantes da indústria como a IBM (*International Business Machines*), SAP (*Business Management Software Solutions Applications and Services*), BEA (*BEA Systems*), Fujitsu (*Fujitsu Technology Solutions*), WebMethods⁵ e IDS Scheer AG (*Software AG*). Em Agosto de 2001, o *Business Process Modeling Notation Working Group* (BPMN-WG), da BPMI.org, foi formado por 35 empresas e iniciou os trabalhos para criar o BPMN (*Business Process Modeling Notation*).

A versão 1.0 da especificação, escrita por Stephen White da IBM, no âmbito do BPMI, surgiu em Maio de 2004 [BPMNI, 2004] e rapidamente se estabeleceu como notação padrão para modelar processos de negócio. Em Junho de 2005, a BPMI anunciou a sua junção ao OMG (*Object Management Group*), associação sem fins lucrativos, desde 1989, e que desenvolve e mantém padrões e especificações, entre elas, a notação UML. A versão 1.1 da BPMN é de Fevereiro de 2008 [OMG, 2008]. A versão 1.2 surgiu em Março de 2009 [OMG, 2009b] e a última versão, a 2.0 Beta 1, é de Agosto de 2009 [OMG, 2009c]. As alterações que foram introduzidas prenderam-se, em grande medida, com a extensão da notação, introduzindo por exemplo outros tipos de eventos. A última versão apresenta um metamodelo completo e, além de extensões à notação, tem uma aproximação muito clara ao mapeamento em BPEL. A notação e seu significado serão descritos detalhadamente no capítulo 4.

1.2.5 UML para a Dinâmica

A dinâmica em UML pode ser modelada através de:

- (i) Diagramas de Casos de Utilização;
- (ii) Diagramas de Sequência;
- (iii) Diagramas de Estados;
- (iv) Diagramas de Actividades.

⁴ <http://www.bpmi.org>

⁵ Empresa que pertence desde 1997 à Software AG.

Todos estes diagramas representam comportamentos e podem ser utilizados para modelar processos. Encontramos um exemplo de estudo de descrição de processos com casos de utilização em [Nawrocki, et al., 2006]. No entanto, são os diagramas de actividades que, correntemente, são utilizados para modelar os processos de negócio.

O diagrama de caso de uso descreve o que faz um sistema (ou parte deste) sem se preocupar em descrever como é realizado (tem o foco na visão externa do sistema). É um documento que descreve a sequência de eventos de um actor, que usa um sistema, para completar um processo.

Apresentamos na **Figura 3** um exemplo de processo modelado com um diagrama de casos de utilização.

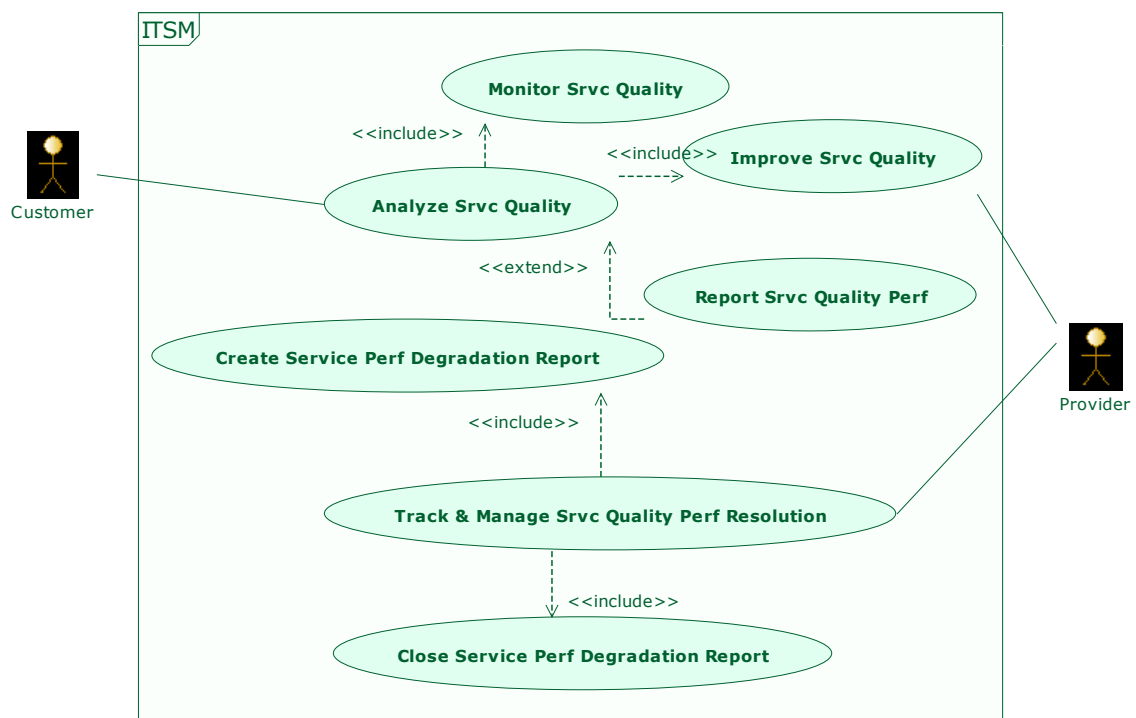


Figura 3 - Exemplo de modelação do processo de “Gestão de Qualidade de Serviço” com um diagrama de casos de utilização

O diagrama de sequência descreve um padrão de interacção entre objectos, apresentado de uma forma cronológica. O diagrama de sequência preocupa-se com a ordem temporal em que as mensagens são trocadas entre os objectos envolvidos em um determinado processo, como ilustramos na **Figura 4**.

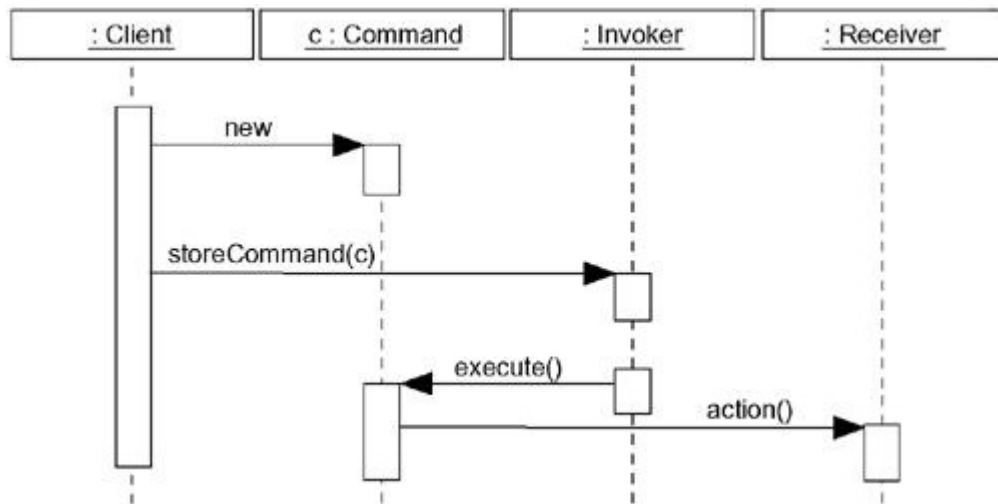


Figura 4 – Exemplo de um diagrama de sequência “Criação de Conta” extraído de [Booch, et al., 1998]

O diagrama de estados representa as alterações de estado dos objectos do sistema (objectos mudam de estado como resposta a eventos e à passagem de tempo), como ilustramos na **Figura 5**.

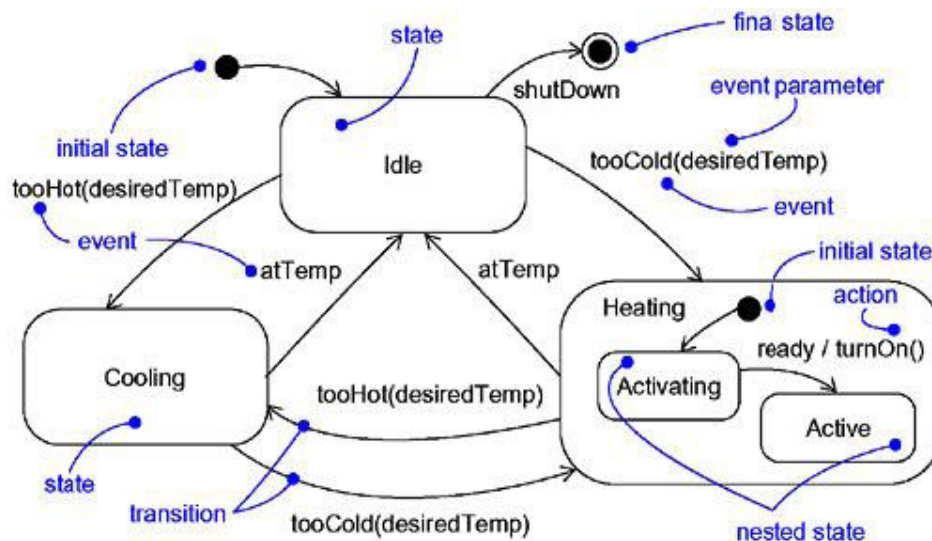


Figura 5 - Exemplo de um diagrama de estados, extraído de [Booch, et al., 1998]

Um diagrama de actividades fornece uma visão simplificada do fluxo de controlo de um caso de uso, de uma operação ou de um processo e é o meio normalmente utilizado para descrever os processos de negócio de um sistema (modelação da lógica do sistema e fluxos de dados). Mostram o fluxo entre actividades (e adicionalmente também o fluxo de informação), numa forma de aplicação simples, tal como o BPMN e até com algumas semelhanças de notação (aspecto do diagrama), como ilustramos na **Figura 6**.

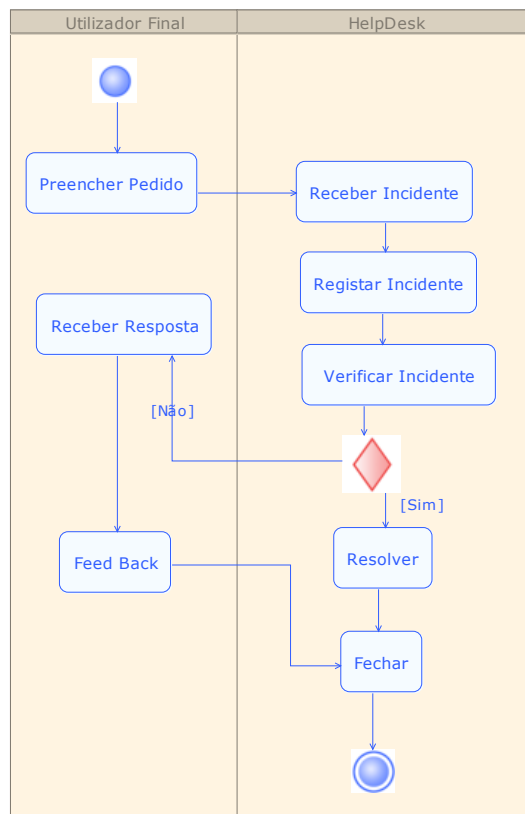


Figura 6 - Exemplo de diagrama de actividades para modelar um processo

Não obstante UML providenciar meios para representar processos como explicámos nesta secção optámos por BPMN, devido à sua riqueza semântica e precisão formal, que a torna apropriada como técnica de modelação dinâmica de serviços. Como exemplo apresentamos no **Anexo E** com o *exemplo de Processo "Monitorização de Incidentes no SLA" modelado com BPMN*. Este tema é abordado novamente na secção 4.1.

1.2.6 Meta modelação

A meta modelação é a construção de um conjunto de "conceitos" (coisas, termos, ideias, etc.) dentro de um determinado domínio. Um modelo é uma abstracção dos fenómenos do mundo real. Um metamodelo é ainda uma outra abstracção, destacando as propriedades do próprio modelo. Um metamodelo constitui uma gramática para o domínio que foi construído [Wikipedia, 2009b].

Um modelo está em conformidade com um metamodelo, por exemplo, um diagrama desenhado em UML está em conformidade com o metamodelo UML. Um dos ramos mais activos do MDE (*Model Driven Engineering*) é a abordagem chamada arquitectura orientada a modelos (MDA) proposta pelo OMG. Esta abordagem é baseada na utilização de uma linguagem para escrever metamodelos chamada MOF (*Meta Object Facility*). Exemplos de metamodelos típicos propostos pelo OMG são UML e SPEM (*Software Process Engineering Meta-Model*) [Martins and Silva, 2004].

Metamodelos estão intimamente relacionados com ontologias. Ambos são frequentemente utilizados para descrever e analisar as relações entre os conceitos.

Uma ontologia procura perceber um domínio e como se descrevem e interligam os conceitos envolvidos.

Os metamodelos são utilizados para descrever um domínio. São nestes que são definidas as DSL (*Domain Specific Language*) que englobam os conceitos da área e o universo do discurso. São utilizados para definir e traduzir conceitos em termos computacionais e a base de processos de MDE (*Model-driven Engineering*) [Cibran, 2008].

Uma ontologia providencia um dicionário num determinado domínio e semântica, ou seja, define ideias (palavras) para que todos tenham uma única interpretação num determinado domínio.

Segundo [Saeki and Kaiya, 2006], uma ontologia providencia estrutura (conceitos e relações) e é mais abstracta que um metamodelo, dado que contém mecanismos que permitam calcular as propriedades semânticas na ontologia e assim verificar faltas ou inconsistências no modelo.

No mesmo artigo [Saeki and Kaiya, 2006] é apresentada uma relação entre modelos, metamodelos e ontologia. Ainda nesse artigo, são apresentados dois tipos de ontologia:

- (i) A ontologia do metamodelo;
- (ii) Ontologia do modelo, também chamado de ontologia do domínio.

Um modelo é uma instanciação de um metamodelo e é semanticamente interpretado pela ontologia do domínio. A ontologia de domínio é uma instanciação da ontologia do metamodelo. A **Figura 7** mostra relações entre uma ontologia, um modelo e um metamodelo.

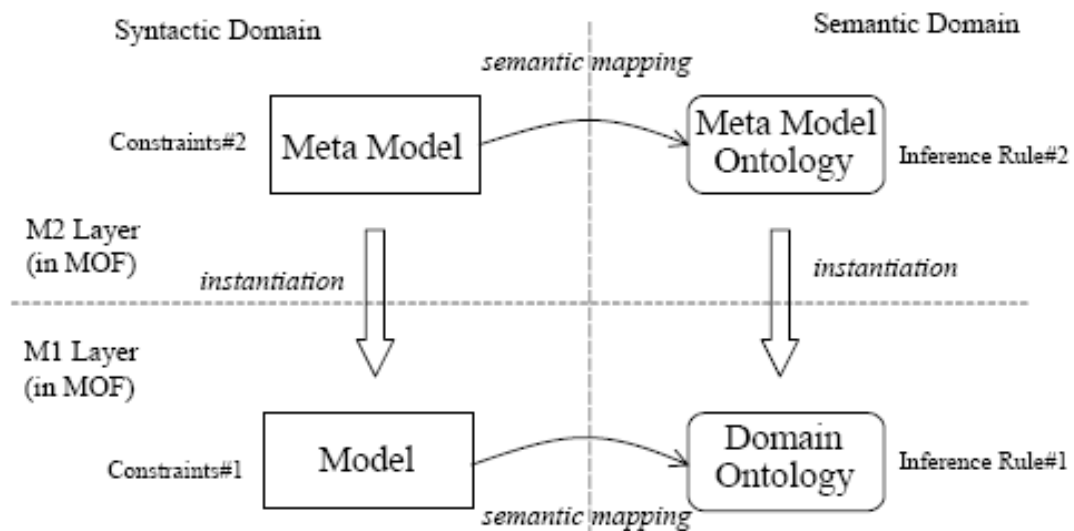


Figura 7 - Relacionamento entre ontologia, modelo e metamodelo, extraída de [Saeki and Kaiya, 2006]

A ontologia do metamodelo contém, por exemplo, a sintaxe da interpretação SQL (*Structured Query Language*) e a ontologia do domínio as expressões SQL no domínio. Modelos são instâncias de metamodelos e as suas estruturas lógicas e sintáticas devem obedecer aos respectivos metamodelos.

Um metamodelo de ITSM (*Information Technology Service Management*) irá ajudar a:

- (i) Definir conceitos e serviços de TI;
- (ii) Relacionar serviços;
- (iii) Sistematizar SLA;
- (iv) Definir associações e conceitos;
- (v) Definir o Universo do discurso;
- (vi) Formalizar o Universo do discurso.

Existe também a necessidade de se modelar os processos, ou seja, de descrever a dinâmica.

1.2.7 Avaliação de qualidade de serviço SLA

1.2.7.1 SLA

Segundo [Araújo, 2005] e [Lee and Ben-Natan, 2002] um *Service Level Agreement* (SLA), ou acordo de nível de serviço, é um contracto ou acordo que formaliza um relacionamento comercial, ou, uma parte da relação entre duas entidades. A maior parte das vezes, assume a forma de um contrato negociado, feito entre um prestador de serviços e um cliente (empregador ou empregado, departamentos de uma empresa ou departamento de TI, empresa contratada para *outsourcing* ou empresa contratante, etc.) e define um preço pago em troca de um direito a um produto ou serviço a ser prestado, sob certos termos, condições e com certas garantias financeiras. Neste documento consta, entre outros:

- (i) O serviço fornecido bem como uma descrição deste – o que é feito (em termos do que é visível para o cliente, não em termos de processos internos), qual a duração, horários, etc.;
- (ii) O nível de qualidade esperado, exemplos; 1 minuto de tempo máximo de resposta, 99% de disponibilidade do servidor, máximo de 2% de produtos com defeito, etc.;
- (iii) As métricas para a avaliação do nível de qualidade, exemplos são a disponibilidade, velocidade, tempo de resposta, etc.;
- (iv) As possíveis recompensas e ou penalizações no caso da qualidade subir ou descer para lá dos níveis propostos.

No anexo “A : Template de SLA ” podemos observar as secções que constituem um SLA. Os acordos do nível de serviço (SLA) foram inicialmente desenvolvidos para o fornecimento de serviços internos das empresas, sendo que os pioneiros foram os serviços de informática. A seguir, foram utilizados para a regulamentação da conectividade de rede entre grandes fornecedores do mesmo nível e, posteriormente, nos serviços externos, onde passaram a ser tratados por meio de contratos, com as especificações de níveis de serviços requeridas pelos clientes.

1.2.7.2 Estrutura e Conteúdo de um SLA

Não existe uma forma generalizada ou normalizada de descrição de um SLA que conheçamos, até porque as aplicações de SLA são muito diversificadas o que conduz naturalmente a formatos e conteúdos muito diversos. No entanto, a partir do nosso conhecimento, exemplos e descrições que se encontram em [Leader, 2009], [Petit, 2004], [Software, 2008], [Wustenhoff, 2002] e [Medeiros, 2007] vamos sugerir uma descrição de um contrato de SLA genérico com a enumeração dos itens a que pode fazer referência. Para o efeito apresentamos o anexo “A : Template de SLA ” para ilustrar aspectos essenciais da composição de um SLA.

De seguida, falaremos um pouco sobre alguns aspectos de um SLA, que servem para introduzir as características do metamodelo ITSM, nomeadamente qualidade dos serviços ou qualidade de produtos, que estão fortemente ligadas aos níveis de qualidade de serviços ou produtos prestados pelos fornecedores, ou sub-fornecedores. Estas considerações são também abordadas nos processos de qualidade de serviço descritos nas normas e padrões internacionais do TeleManagement Forum ou ITIL [Forum, 2007], [Commerce, 2007].

Na grande maioria dos casos, o fornecedor de serviços tem pouco conhecimento dos negócios do cliente ou de futuros clientes, o que torna difícil estabelecer o dimensionamento correcto dos SLA. Assim, é geralmente necessária a elaboração do acordo de nível de serviço para a necessidade específica de cada cliente, ou seja, o SLA será implementado de forma personalizada e com capacidade de se ajustar automaticamente às novas necessidades que surgirem durante a sua vigência (SLA dinâmico). No entanto, é de referir que também existem SLA padronizados, que todos utilizamos de forma quase invisível, como é o caso das telecomunicações, cujos parâmetros de qualidade são estabelecidos e verificados pela Anacom (Autoridade Nacional para as Comunicações).

Observa-se que a política de SLA é quase sempre determinada pelo fornecedor e é intrínseca ao produto ou serviço oferecido. Muitas vezes o cliente não possui processos nem procedimentos para a elaboração de um SLA adequado ao seu modelo de negócio ou plano de negócios, resultando daí a inadaptação do SLA e insatisfação de clientes e, por sua vez, de fornecedores.

O SLA pode ser considerado em duas partes distintas: uma contendo as informações que não dependem de particularidades dos serviços, tais como, as informações sobre disponibilidade, fiabilidade, privacidade e aspectos de segurança; a outra, contendo dados e processos específicos do serviço, a forma de atendimento, tempos de resposta, locais, etc.

1.2.7.3 Tipos de visões de SLA

Os serviços e SLA associado podem ter duas visões distintas:

- (i) *Black Box* – Descrevem relações entre clientes e fornecedores; uma aplicação típica é a utilização em aplicações Web, onde se reforça a não competição por recursos dos serviços. Quando várias aplicações Web partilham um servidor, um desafio consiste em aplicar o isolamento de desempenho, ou seja, uma aplicação não deve ser capaz de roubar recursos de outras aplicações. O isolamento leva à especificação de níveis de serviço (SLA) que quantificam a garantia de execução com determinada performance, através da utilização de KPI (*Key Performance Indicador*) [Philippe, et al., 2006];
- (ii) Engenharia de Processos – Um exemplo de SLA deste tipo é o *Service Level Agreement Management* do ITIL que existe nas versões v2 (parte do *Service Delivery*) [OGC, 2003] e v3 (parte do *Service Design*) [Lloyd and Rudd, 2007]. O objectivo é a gestão *Service Level Agreements*, de forma que os requisitos do cliente são reflectidos e os contratos são coordenados e harmonizados. O requisito básico é o de equilibrar o valor e qualidade para o cliente com os custos do serviço. *Service Level Agreement Management* é responsável pela criação, revisão e cancelamento de *Service Level Agreements* (SLA) com o cliente. É um processo de adaptação interactivo.

1.2.7.4 Indicadores de desempenho

Indicadores ou chaves de Desempenho, KPI (*Key Performance Indicador*), medem o nível de desempenho do processo, indicando como e quando os processos de tecnologia da informação permitem que o objectivo seja alcançado. Pode-se gerar indicadores para qualquer etapa de um processo e medir o seu resultado [Wustenhoff, 2002].

Os KPI tiveram a sua aplicação estendida às mais diversas questões referentes aos negócios e empresas. Podem-se gerar indicadores para qualquer etapa de um processo e medir o seu resultado, apresentamos alguns exemplos na **Figura 8**. Muitas empresas trabalham com KPI como instrumentos de sua navegação. Eles vão além das tradicionais métricas financeiras e passam a medir o sucesso dos processos ou o alcançar de um

objectivo estratégico nas organizações através da combinação de indicadores. Podem existir diversos indicadores que apontam diferentes objectivos e apoiam diagnósticos.

Outro facto importante referente a KPI, é que, quando forem compatíveis e disponibilizados, é possível comparar desempenhos entre empresas ou processos.

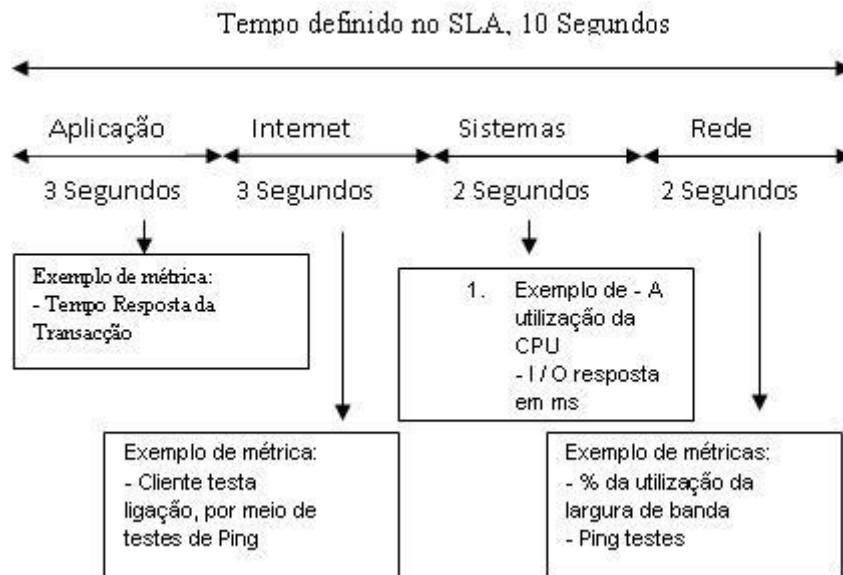


Figura 8 - Exemplos de KPIs

1.3 Identificação do Problema

Tal como indicado anteriormente, os serviços e correspondentes acordos de nível de serviço (SLAs) são usualmente tratados de forma informal, desconexa e dispersa entre várias disciplinas, e de forma não abrangente em termos de ciclo de vida. Não existe também uma linguagem ou representação comum de definição de conceitos. Esta situação tem várias consequências, em particular dificultando:

- i) a colaboração entre as partes envolvidas (*stakeholders*);
- ii) a capacidade de automação dos processos de ITSM;
- iii) a comparabilidade entre as diferentes metodologias de gestão de ITSM;
- iv) a implementação de acções de avaliação e melhoria dos processos de ITSM;
- v) a investigação sobre as relações de causalidade existentes no domínio de ITSM, usando o método científico, o que implica a replicação de experiências.

1.4 Objectivos, solução proposta e contribuições

Os objectivos que pretendemos atingir com esta dissertação são:

- (i) Definir e representar formalmente conceitos de ITSM;
- (ii) Representar os conceitos abrangidos pelos SLA no contexto de ITSM;
- (iii) Representar a dinâmica dos serviços de ITSM;
- (iv) Compatibilizar conceitos como produtos e serviços não só de um ponto de vista estático, mas também dinâmico.

Em suma, esperamos trazer alguma luz ao universo da representação de conceitos no domínio de ITSM, melhorando a modelação de conceitos, nomeadamente clarificando as relações entre os diversos conceitos e as mudanças que ocorrem ao longo dos seus ciclos de vida. Na senda destes objectivos, prevemos as seguintes contribuições:

- (i) Estudo comparativo de diferentes abordagens de ITSM e definição de SLAs;
- (ii) Criação de um método de representação flexível, abrangente e formal capaz de representar todos os conceitos relevantes de ITSM, como recursos, produtos e serviços, para permitir uma representação formal de um SLA, em particular segundo duas perspectivas:
 - a. Componente Estática: através da definição de uma Ontologia de ITSM;
 - b. Componente Dinâmica: através da definição de um metamodelo da linguagem BPMN, capaz de suportar a representação de processos inerentes a ITSM.

Interessa desde já esclarecer a necessidade do metamodelo de BPMN. Quando esta dissertação se iniciou, este metamodelo estava ainda em fase de elaboração na OMG. No ano transacto foi finalmente publicada uma versão supostamente estável [OMG, 2009a]. Este metamodelo de BPMN, publicado pelo OMG, é uma extensão (perfil) do metamodelo de UML. Tal como este último, é suportado no referencial designado por *Meta Object Facility (MOF)* [OMG, 2002], para a criação de metamodelos. Daqui resulta uma complexidade excessiva para o contexto dos nossos trabalhos de investigação. Optou-se assim por desenvolver uma versão independente, mais leve, de um metamodelo para o

BPMN, que garantisse a cobertura de todas as abstrações de modelação, sem sacrificar a necessária facilidade de compreensão e utilização.

1.5 Metodologia

Nesta secção, vamos definir a metodologia de investigação que vamos usar e o domínio da dissertação. Esta enquadra-se no domínio da Epistemologia, dado que procura desenvolver o conhecimento sobre Serviços, SLA e Processos associados. A Epistemologia estuda a origem, a estrutura, os métodos e a validade do conhecimento, motivo pelo qual também é tipicamente conhecida por filosofia do conhecimento [Wikipédia, 2010].

A Ontologia refere-se a uma realidade externa ao ser humano, uma visão sobre um domínio particular ou uma determinada perspectiva da realidade. A Epistemologia preocupa-se com o conhecimento e como ele pode ser adquirido [Farren, 2005].

Recordamos que abordámos anteriormente o que é uma ontologia e um metamodelo do ponto de vista da Engenharia Informática (secção 1.2.6). Uma ontologia é uma generalização e sistematização do conhecimento num diverso universo de discurso, no caso corrente o de ITSM. Um metamodelo é também, em nosso entender, uma ontologia, em que o domínio é o de uma linguagem de modelação (BPMN neste caso).

A investigação depende de um conjunto de procedimentos para que os objectivos sejam atingidos; os métodos científicos. No nosso caso o método científico que fornece a base lógica da investigação é o método indutivo. O método indutivo considera que o conhecimento é fundamentado na experiência, não levando em conta princípios preestabelecidos. No raciocínio indutivo a generalização deriva de observações de casos da realidade concreta, partindo de casos particulares para generalizações [GIL, 1999]. As constatações particulares levam à elaboração de generalizações. No nosso caso a partir do estudo de exemplos como o ITIL, o eTOM, o MNM Service Model ou o BPMN generalizamos para a ontologia e metamodelo apresentados nos capítulos 3 e 4.

Segundo [Silva and Menezes, 2001] uma investigação é um procedimento reflexivo e crítico de busca de respostas para problemas ainda não solucionados. Uma investigação faz parte de um processo sistematizado que inclui etapas e que pode ser detalhado da seguinte forma, de acordo com os mesmos autores:

- 1) Escolha do tema;

- 2) Revisão de literatura;
- 3) Justificação da escolha do tema;
- 4) Formulação do problema;
- 5) Determinação dos objectivos;
- 6) Metodologia;
- 7) Recolha de dados;
- 8) Apresentação de dados;
- 9) Análise e discussão dos resultados;
- 10) Conclusão da análise dos resultados;
- 11) Redacção e apresentação do trabalho científico.

Vamos apresentar cada fase e descrever o que fizemos e como se adequa ao nosso trabalho:

1) Escolha do tema – Este encontra-se definido na introdução em termos gerais.

2) Revisão de literatura - A revisão da literatura é uma das fases mais importantes [Silva and Menezes, 2001]. A revisão de literatura resulta do processo de levantamento e análise do que já foi publicado sobre o tema e o problema de investigação escolhidos, permitindo determinar o “estado da arte”. O traçar desse perfil permite identificar as principais limitações e lacunas existentes. Para o efeito elaborámos uma taxonomia composta por um conjunto de critérios que se pretendem objectivos, a qual permitiu classificar e comparar os trabalhos de forma sistemática, como se pode encontrar no capítulo 2.

3) Justificação da escolha do tema – Partindo das limitações constatadas no estado da arte, identificámos oportunidades para a sua melhoria, consubstanciadas na proposta de um conjunto de contribuições, tal como justificado na secção 1.2.1.

4) Definição do problema – Segundo [Silva and Menezes, 2001] nesta etapa reflectimos sobre o problema que pretendemos resolver na investigação, se é realmente um problema e se vale a pena tentar encontrar uma solução para ele. A investigação científica depende da formulação adequada do problema, isto porque se pretende encontrar uma solução para ele. Essa identificação foi realizada na secção 1.3.

5) Determinação dos objectivos – Os mesmos foram apresentados na secção 1.4.

6) Metodologia – Segundo [Silva and Menezes, 2001] nesta etapa devemos definir onde e como será realizada a investigação. Define-se o tipo de investigação, a população

(universo da investigação), a amostragem, os instrumentos de recolha de dados e a forma como se pretende apresentar e analisar esses dados. Nesta dissertação não realizamos tratamento estatístico dado que o objectivo foi o estudo de casos, sua generalização e definição de conceitos. As amostras foram não-probabilísticas devido a serem amostras acidentais, compostas por acaso, com casos que apareceram na revisão da literatura e amostras intencionais, ou seja foram escolhidos casos para a amostra que representam os melhores exemplos que encontramos.

7) Recolha de dados – A recolha de dados foi efectuada em estudos de casos na literatura; para além disso foi realizada por observação em equipa, dado que esta dissertação foi desenvolvida no âmbito do QUASAR, em conjunto com mais duas dissertações [Costa, 2010, Porciúncula, 2010]. Estas últimas, por sua vez, utilizaram técnicas como entrevistas e questionários. Na secção 4.5 encontra uma descrição da interacção entre estas 3 dissertações.

8) Apresentação de dados – Produzimos várias tabelas, quadros e gráficos como usual, no capítulo 2.

9) Análise e discussão dos resultados – A forma de representação escolhida nesta dissertação para apresentação de resultados foi uma das mais amplamente usadas na construção de ontologias e metamodelos: diagramas de (meta)classes em UML, com recurso a regras de boa formação expressas com recurso à linguagem de restrições OCL, como realizado nos capítulos 3 e 4. No caso particular desta dissertação não há confirmação ou rejeição de hipóteses, mas os modelos formulados foram testados com exemplos e amplamente utilizados em casos reais nas supracitadas dissertações [Costa, 2010, Porciúncula, 2010].

10) Conclusão da análise dos resultados – Sintetizamos os resultados obtidos na secção 5.3 onde revemos as contribuições esperadas, fazemos uma comparação entre os resultados obtidos e sumariamos, na secção 5.4, as principais contribuições para o meio científico.

12) Redacção e apresentação do trabalho científico – Escrita da presente dissertação.

1.6 Convenções tipográficas e estrutura desta dissertação

Para distinguir os elementos semanticamente diferentes e fornecer uma orientação visual para o leitor, esta dissertação utiliza as seguintes convenções tipográficas:

- Em *itálico* distinguimos palavras-chave, variáveis, termos científicos (especialmente em inglês), fórmulas, métodos e ferramentas com um significado especial na literatura técnica ou científica;
- **Negrito** indica cabeçalhos de assunto, cabeçalhos de tabelas, as questões de pesquisa e itens em enumerações.

Esta dissertação está organizada num conjunto de capítulos e resume-se a:

Capítulo 1. Introdução - Este capítulo introduz os principais conceitos utilizados nesta dissertação e refere a motivação desta dissertação. Ele também enumera as principais contribuições e apresenta um breve resumo de cada um dos capítulos restantes.

Capítulo 2. Estado da arte - Apresenta uma visão geral dos trabalhos relacionados assim como a taxonomia utilizada para classificá-los e exibe no final um estudo de oportunidades, dividido em duas partes: (i) serviços e SLA, (ii) representação da dinâmica.

Capítulo 3. Ontologia geral de serviços e SLA - Descreve e apresenta o primeiro resultado obtido, ou seja, a proposta de ontologia ITSM para aplicação em SLA.

Capítulo 4. Representação da dinâmica - Detalha um metamodelo de BPMN para representação da dinâmica.

Capítulo 5. Conclusões e trabalho futuro – A dissertação finaliza com as conclusões e abre um debate para os futuros trabalhos nesta área.

[Esta página foi intencionalmente deixada em branco]

Estado da Arte

2.	ESTADO DA ARTE.....	32
2.1	AS TAXONOMIAS	32
2.2	ESTADO DA ARTE NA REPRESENTAÇÃO DO CONHECIMENTO ITSM	33
2.2.1	<i>Taxonomia ITSM</i>	33
2.2.2	<i>Trabalhos Relacionados em ITSM</i>	37
2.2.3	<i>Análise Comparativa e Lacunas ITSM</i>	49
2.3	ESTADO DA ARTE DOS META MODELOS BPMN	50
2.3.1	<i>Taxonomia para Meta Modelos BPMN</i>	50
2.3.2	<i>Trabalhos Relacionados BPMN</i>	55
2.3.3	<i>Análise comparativa e lacunas meta modelo BPMN</i>	58

Este capítulo descreve as taxonomias definidas para avaliar trabalhos relacionados e usa as mesmas para identificar as limitações desses trabalhos e a necessidade de novos desenvolvimentos.

2. Estado da Arte

Para apoiar a nossa investigação procurámos trabalhos relacionados com a área de modelação, serviços, gestão de TI, qualidade, representação da dinâmica, metamodelos e BPMN, recorrendo, para tal, a várias bibliotecas digitais, tais como, ACM⁶ (*Association for Computing Machinery*), IEEE⁷ (*Institute of Electrical and Electronics Engineers*), Springer⁸ (*Springer Science+Business Media Publishing Company*) e Elsevier⁹ (*Publishing Company*). Encontrámos apenas alguns artigos sobre Modelação de ITSM e a dificuldade foi ainda maior na área da meta modelação de processos com BPMN.

A escolha dos artigos ficou assim limitada e, como consequência, foram também revistos artigos de subtemas relacionados e, nalguns casos, já com algum tempo de publicação.

2.1 As Taxonomias

A taxonomia, segundo [Campos and Gomes, 2008], é a ciência da sistematização e classificação e é entendida, no âmbito da Ciência da Informação, como uma ferramenta de organização intelectual. O mesmo termo designa também (e é nesta acção que é usado nesta dissertação) compostos de unidades taxonómicas, critérios ou categorias que são frequentemente organizados numa estrutura hierárquica. Cada uma destas categorias deve ser descrita de tal forma que, para um determinado assunto, será simples identificar se pertence, ou não, à categoria, reduzindo desta forma a ambiguidade.

As taxonomias são utilizadas em portais institucionais e bibliotecas digitais como um novo mecanismo de consulta, ao lado de ferramentas de pesquisa. Além destas aplicações, a taxonomia é um dos componentes em ontologias. A organização das informações através do conceito de taxonomia permite alocar, recuperar e comunicar informações dentro de um sistema de uma forma lógica [Campos and Gomes, 2008].

⁶ <http://www.acm.org/>

⁷ <http://www.ieee.org>

⁸ <http://www.springer.com>

⁹ <http://www.elsevier.com>

Uma taxonomia para a classificação de trabalhos relacionados permitirá uma utilização mais objectiva de um conjunto de critérios de comparação, facilitando assim a visão do actual estado da arte nesta área.

2.2 Estado da Arte na Representação do Conhecimento ITSM

2.2.1 Taxonomia ITSM

A taxonomia ITSM proposta é composta pelos seguintes seis critérios de classificação:

- (i) **Âmbito** – Refere o principal domínio de aplicação da proposta;
- (ii) **Intenção** – Refere os seus objectivos subjacentes;
- (iii) **Técnica** – Descreve as principais etapas da metodologia da proposta e a forma pela qual as regras do modelo são elaboradas;
- (iv) **Crítica** – É uma avaliação qualitativa das características mais notáveis da proposta, incluindo as suas insuficiências;
- (v) **Maturidade** – Um conjunto de descritores com sete dimensões. Cada dimensão é classificada num intervalo [0, 3], tal como se verifica na **Tabela 1** (página 36).

A classificação de 0 a 3, atrás referida, foi escolhida porque é, na nossa opinião, simples, intuitiva e permite níveis de avaliação gerais do tipo Muito Bom, Bom, Médio e Mau, aceites geralmente e bem compreendidos, além de que número par de níveis o que permite fugir das avaliações no ponto médio.

No caso em apreço de revisão bibliográfica a descrição da taxonomia permite reduzir a ambiguidade na clarificação do estado da arte.As dimensões de maturidade que analisámos são as seguintes:

1. **Domínio de Aplicação (DoA);**
2. **Precisão das Técnicas (PoT);**
3. **Composição de Serviços (SvC);**
4. **Definição de SLA (SLA);**
5. **Validação do Modelo (VAL);**

6. Cobertura do Ciclo de Vida (LCC);

7. Definição de Serviço (SEC).

Cada proposta de ITSM estudada foi classificada de acordo com as citadas dimensões e respectivas classificações. Vamos de seguida caracterizar estas dimensões.

2.2.1.1 Domínio da Aplicação

A dimensão **DoA (Domain of Application)** procura estabelecer em que medida é que a proposta estudada se enquadra num panorama global de representação de serviços de TI, e quais as áreas de cobertura que referem (por exemplo, cliente ou fornecedor, serviço, SLA, QoS, etc.). Procurámos medir o grau de aplicabilidade das propostas à definição de Serviços, então, quanto mais amplo o domínio de aplicação, melhor. As categorias identificadas, aumentando o nível de maturidade são as seguintes:

- 0 - Muito específica e técnica – a proposta é construída para uma área muito específica e abrange questões técnicas concretas (por exemplo, segurança);
- 1 - Específica – a proposta centra-se num domínio parcial ITSM (por exemplo, SLA);
- 2 - Parcial – a proposta abrange diversos, mas não todos os domínios de ITSM;
- 3 - Geral – a proposta abrange a grande maioria dos domínios relevantes de ITSM.

2.2.1.2 Dinâmica do Sistema

A dimensão **PoT (Precision of Techniques)** refere-se à precisão das técnicas utilizadas para capturar o elemento dinâmico dos sistemas (por exemplo, diagramas de actividades, estado gráfico, BPMN, redes de Petri, etc.). A verificação cruzada da validação dinâmica entre os modelo estático e dinâmico deverá ser adequada. As categorias identificadas, aumentando o nível de maturidade são as seguintes:

- 0 - Nenhum – sem modelos que representem a dinâmica ITSM;
- 1 - Parcial – há alguns diagramas alusivos à dinâmica ITSM, mas exemplificam as questões de uma maneira superficial e parcial;
- 2 - Definida – apesar de existirem modelos dinâmicos que cobrem quase todo o sistema, são muito genéricos, e não permitem a simulação ou a verificação de conformidades com SLA;
- 3 - Formal – é fornecido um modelo integrado, global e formal e estão previstas verificações de conformidades com SLA.

2.2.1.3 Composição de Serviços

A dimensão **SVC (Services Composition)** refere-se ao grau de precisão e rigor com que se caracterizam as componentes dos serviços (processos, actividades e acções) utilizadas em ITSM e relacionados com SLA, de acordo com a seguinte escala ordinal:

- 0 - Natural – os conceitos são apresentados em linguagem natural de forma bastante informal e não estão estruturados;
- 1 - Estruturada – é utilizada uma linguagem natural, mas são blocos hierárquicos e conceitos definidos de cima para baixo (*top-down*);
- 2 - Definida – há uma ontologia subjacente, mas a semântica é difusa devido à falta de formalidade;
- 3 - Formal – é utilizada uma linguagem formal como OCL, BPMN ou BPEL para conceder a precisão e formalismo aos processos de ITSM.

2.2.1.4 Especificação de SLA

A dimensão **SLA (SLA Activities Definition)** representa a precisão do processo de especificação de SLA e é definida a seguinte escala ordinal:

- 0 - Inexistente – não se fala de SLA;
- 1 - Informal – o processo relativo ao SLA é informal e difuso;
- 2 - Definida – existe um modelo geral para o processo de definição de SLA, mas com falta de detalhes;
- 3 - Formal – um procedimento formal é adoptado na especificação do SLA, por exemplo recorrendo a uma linguagem de restrições.

2.2.1.5 Validação do Modelo

A dimensão **VAL (Model's Validation)** representa o grau de validação, sobre a forma de um exemplo (mais ou menos completo e complexo) realizada nos processos proponentes. Esta validação é muito importante para clarificar a sua semântica:

- 0 - Sem Validação – a proposta é apenas conceitual;
- 1 - Exemplificativa – apenas são fornecidos pequenos exemplos;
- 2 - Amplamente testada – o modelo é instanciado com alguns exemplos consistentes;
- 3 - Totalmente testada – o modelo é instanciado de um modo geral consistente com vários exemplos.

2.2.1.6 Cobertura do Ciclo de Vida

A dimensão **LCC (Life Cycle Coverage)** representa a extensão em que o "ciclo de vida"¹⁰ do produto ou serviço é abrangido pela proposta dos processos de SLA e é definida sobre a seguinte escala ordinal:

- 0 - Nenhuma – não há referência ao ciclo de vida explícita ou implicitamente;
- 1 - Parcialmente coberta – apenas é considerada uma fase do ciclo de vida e, como tal, é coberto muito parcialmente o ciclo de vida;
- 2 - A maior parte – várias fases do ciclo de vida (não todas) são cobertas;
- 3 - Totalmente – todo o ciclo de vida é explicitamente abrangido.

2.2.1.7 Verificação de Conformidade

A dimensão **SEC (Service Compliance)** representa o processo de verificação de conformidade entre os serviços contratados e serviços prestados e é definida sobre a seguinte escala ordinal:

- 0 - Nenhuma – esta questão não é tratada;
- 1 - Informal – esta questão é tratada informalmente;
- 2 - Definida – esta questão é modelada genericamente;
- 3 - Formal – está definido um modelo de processos formal.

2.2.1.8 Tabela Resumo da Taxonomia ITSM

Nível	DOA	POT	SVC	SLA	VAL	LCC	SEC
0	Muito específica e técnica	Nenhum	Natural	Inexistente	Sem Validação	Nenhuma	Nenhuma
1	Específica	Parcial	Estruturada	Informal	Exemplificativa	Parcialmente coberta	Informal
2	Parcial	Definida	Definida	Definida	Amplamente testada	A maior parte	Definida
3	Geral	Formal	Formal	Formal	Totalmente testada	Totalmente	Formal

Tabela 1 - Dimensões de Maturidade e Níveis Correspondentes

¹⁰ Ciclo de vida de um produto ou serviço é definido normalmente como 1 Introdução, 2 Crescimento, 3 Maturidade, 4 Declínio, ou seja criação e apresentação ao mercado, aceitação no mercado, entrada de concorrentes ou saturação, desatualização e saída de produção.

2.2.2 Trabalhos Relacionados em ITSM

Para perceber o actual estado da arte nos diversos documentos pesquisados seleccionámos e estudámos alguns mais profundamente, naturalmente procurámos os mais específicos, importantes (no sentido de serem utilizados e reconhecidos) e recentes. No entanto como já foi dito tivemos alguma dificuldade em encontrar artigos sobre o assunto específico que procurámos. É importante salientar que o nosso objectivo, ao analisar esses trabalhos publicados, era tirar conclusões sobre os méritos relativos, e sistematizar e avaliar a metodologia de avaliação, utilizando a taxonomia anterior definida, além de procurar ideias que fossem um ponto de partida para a inovação que pretendíamos. Além da categorização, para cada obra, apresentamos um resumo e uma crítica.

2.2.2.1 *Software Maintenance from a Service Perspective*

[Niessink and Vliet, 2000]

Âmbito – Manutenção de Software.

Intenção – Investigar as diferenças entre manutenção de software e desenvolvimento de software.

Técnica – Os autores identificam quatro processos que dizem respeito à qualidade de manutenção de software:

- (i) Traduzir as expectativas dos clientes, no que diz respeito à manutenção, em acordos claros;
- (ii) Utilizar estes acordos de serviços como base de planeamento e implementação de actividades de manutenção;
- (iii) Assegurar de que a manutenção é feita de acordo com planos e procedimentos;
- (iv) Gerir a comunicação das actividades realizadas.

Crítica – Este trabalho apresenta SLA num contexto muito reduzido e focado (relativo a um software de manutenção). Não é feita referência a modelos estruturais que possam ser validados e instanciados.

Tal como definimos na secção 2.2.1 em que criamos uma Taxonomia para ITSM, apresentamos de seguida na **Tabela 2** a nossa classificação do trabalho.

	DOA	POT	SVC	SLA	VAL	LCC	SEC
Niessink et al., 2000	1	1	0	2	0	2	2

Tabela 2 - Classificação de Software Maintenance from a Service Perspective

2.2.2.2 The MNM Service Model

[Garschhammer , et al., 2001b] , [Garschhammer , et al., 2001a]

Âmbito – Meta modelação de gestão de serviços em geral.

Intenção – Desenvolver o metamodelo apresentado no documento “*Towards Generic Service Management Concepts*” [Garschhammer , et al., 2001a, Garschhammer , et al., 2001b], que continua a ser desenvolvido em “*The MNM Service Model – Refined Views on Generic Service Management*” [Garschhammer , et al., 2001b], após múltiplas aplicações na indústria, acrescentando a noção de pontos de vista.

Técnica – O modelo de base é refinado, adicionando a “vista de serviço” e a “vista de realização”. O modelo básico é definido como a combinação de serviços e de papéis (*roles*). Um Serviço é definido pelas suas funcionalidades e é explicitamente visto como estando separado da sua implementação. A funcionalidade do serviço é definida por um conjunto de interacções que ocorrem entre os papéis. Os papéis identificados são: utilizador, cliente e fornecedor.

O modelo de base é dividido em três partes independentes: o lado do cliente, o lado do fornecedor e o serviço.

1. Vista de serviço – Abstrai os detalhes da implementação, define o serviço independentemente de qualquer fornecedor ou cliente. O seu objectivo é o de servir de base para SLA.
2. Vista de realização – Detalhes no âmbito do prestador, a concentrar-se na implementação. O objectivo é o de servir como ponto de partida para a fase de análise. Um exemplo é uma aplicação de serviço de *help desk*.

Crítica – A proposta de metamodelo é conceitual e não é destinado para implementação. A falta de formalidade e detalhe deixa algumas questões em aberto na definição de conceitos. Não obstante, o artigo apresenta boas ideias, como por exemplo a recursividade dos recursos para modelação da subcontratação.

Tal como definimos na secção 2.2.1 em que criámos uma taxonomia para ITSM, apresentamos de seguida na **Tabela 3** a nossa classificação do trabalho.

	DOA	POT	SVC	SLA	VAL	LCC	SEC
Garsch. et al., 2001	3	2	2	2	1	3	1

Tabela 3 - Classificação de *Towards generic service management concepts - MNM Model*

2.2.2.3 *Precise Service Level Agreements*

[Skene, et al., 2004]

Âmbito – Automação de negociação sobre a composição qualitativa dos sistemas distribuídos, com o foco em serviços web, em áreas genéricas como aplicação de gestão da cadeia de abastecimento (*Supply Chain Management*).

Intenção – Definir o conceito de compatibilidade SLA e estender um perfil UML que permite a modelação de serviços, como um precursor para análise e implementação. É apresentada a concretização de um SLA.

Técnica – Os autores propõem a SLANG, uma linguagem formal para a definição de SLA. As fases para definir a semântica da SLANG foram:

- (i) Modelar a sintaxe da linguagem em UML;
- (ii) Relacionar o modelo da linguagem com um modelo que descreve a estrutura e o comportamento de serviços;
- (iii) Impor restrições definidas formalmente com OCL sobre o comportamento dos elementos.

É apresentado um metamodelo em UML com o uso de OCL.

Crítica – Este artigo constitui um bom exemplo de criação de uma DSL. No entanto o seu âmbito de aplicação é *Web Services*, com SLAs sobre servidores, tempos de resposta, etc. e por isso de âmbito muito restrito.

Tal como definimos na secção 2.2.1 em que criamos uma Taxonomia para ITSM, apresentamos de seguida na **Tabela 4** a nossa classificação do trabalho.

	DOA	POT	SVC	SLA	VAL	LCC	SEC
Skene et al., 2004	3	3	2	3	2	0	2

Tabela 4 - Classificação de *Precise Service Level Agreements*

2.2.2.4 UML Profile for QoS

[OMG, 2006]

Âmbito – Um perfil UML para representar conceitos QoS (*Quality of Service* / Qualidade de Serviço).

Intenção – Incluir requisitos não-funcionais em modelos de análise, associando-os com elementos do modelo da arquitectura.

Técnica – Esta abordagem define uma linguagem abstracta que suporta conceitos de modelação QoS em modelos UML. A *QoS Modeling Language* oferece suporte para a especificação das características QoS. Estas últimas são quantificadas com alguns parâmetros específicos e métodos, e com outras características com um menor nível de abstracção.

As características QoS podem ser agrupadas em categorias que partilham atributos comuns. Existem também restrições que definem qualquer tipo de restrições impostas sobre características QoS. As restrições expressam limitações sobre os parâmetros e métodos das características.

Eles identificam intervalos de valores permitidos para um ou vários parâmetros, os métodos e as suas dependências.

Às vezes, os elementos funcionais básicos do modelo QoS destinam-se a suportar diferentes modos de execução, com diferentes níveis de qualidade. Cada modo tem diferentes restrições QoS associadas e as suas funcionalidades podem ser diferentes.

Em geral, a concepção da arquitectura funcional deve ter em conta estes modos, e existem diferentes funções e componentes para cada modo. A transição de um modo de execução para outro requer algumas acções na execução do pedido, e alguns tipos de transições não podem ser permitidas. A proposta também engloba o conceito de monitores para detectar a não realização de algumas restrições QoS e as acções a desenvolver neste caso.

Crítica – A abordagem apresenta em detalhe um modelo estrutural genérico, mas apenas para QoS de requisitos não funcionais. A especificação precisa de QoS é alcançada por meio de restrições OCL. O modelo não se destina a gerir explicitamente contractos SLA, mas pode ser adaptado com o objectivo de se implementar a noção de QoS num modelo de contractos com níveis de serviço. Por outro lado, está fora do âmbito a definição do modelo de serviços de gestão e do seu ciclo de vida.

Tal como definimos na secção 2.2.1 em que criamos uma Taxonomia para ITSM, apresentamos de seguida na **Tabela 5** a nossa classificação do trabalho.

	DOA	POT	SVC	SLA	VAL	LCC	SEC
OMG, 2006	1	3	0	1	3	0	3

Tabela 5 - Classificação de *UML Profile for QOS*

2.2.2.5 Contract Negotiation and SLA

[Teyssié, 2006]

Âmbito – Especificações de QoS e SLA automatizados para redes, onde a própria rede garante que determinados parâmetros (atrasos, largura de banda, perdas, etc.) estão dentro de limites bem definidos (valor mínimo, valor máximo).

Intenção – Negociações automatizadas entre dispositivos de rede, numa aproximação desenhada em UML, com conceitos de QoS com uma abordagem orientada para o desenho de rede através da especificação de SLAs. A Qualidade de Serviço (QoS) é um requisito das aplicações para a qual se exige que determinados parâmetros (atrasos, largura de banda, perdas, etc.) estejam dentro de limites bem definidos. A QoS é garantida pela rede, suas componentes e equipamentos utilizados. Do ponto de vista dos programas de aplicação, a QoS é tipicamente expressa e solicitada em termos de uma "Solicitação de Serviço" ou "Contrato de Serviço".

Técnica – O autor considera que a negociação pode ser automatizada ou não. No caso de não ser automatizada o responsável pela rede deve modelar ele mesmo o QoS. No caso da negociação automatizada o artigo sugere três processos de negociação:

- (i) UCA (*User Centered Adaptation*) – Adaptação centrada no utilizador. A rede oferece o mais fraco QoS que responda às restrições do utilizador;
- (ii) PCA (*Provider Centered Adaptation*) – Adaptação centrada no fornecedor. O utilizador escolhe entre as diversas classes de serviço que o fornecedor disponibiliza;
- (iii) WBA (*Well Balanced Adaptation*) – Adaptação balanceada, máxima qualidade fornecida face aos requisitos mínimos do utilizador.

São descritos diversos algoritmos para automatização de SLA utilizando lógica de 1.ª ordem. Os SLA são definidos como composições de SLS (*Service Level Specifications*), que representa o QoS negociado com TCS (*Traffic Conditioning Specification*).

Os SLA são descritos por 6 pontos:

- (i) Identificador;
- (ii) Nome;
- (iii) Condição de validação de tempo (Contém o SLS que deve ser alcançado);
- (iv) TCA (*Traffic Conditioning Agreement*), que é o conjunto de restrições que representa as restrições e as prioridades QoS;
- (v) Melhorar a definição de SLA, para permitir o alargamento da definição dos novos elementos;

- (vi) A validação do SLA é feita através da verificação de compatibilidade e consistência dos requisitos de QoS, criticidade das restrições e prioridade de verificação.

Crítica – O modelo e as fórmulas são apresentadas no âmbito de um metamodelo demasiado abstracto. Uma vez que este não é instanciado, é difícil entender os conceitos. O autor não identifica o trabalho futuro.

Tal como definimos na secção 2.2.1 em que criamos uma Taxonomia para ITSM, apresentamos de seguida na **Tabela 6** a nossa classificação do trabalho.

	DOA	POT	SVC	SLA	VAL	LCC	SEC
Teyssié, 2006	0	2	0	3	1	0	3

Tabela 6 - Classificação de UML- *Contract Negotiation and Service Level Agreements*

2.2.2.6 Understanding and Specifying Information Security Needs to Support the Delivery of High Quality Security Services

[Su, et al., 2007]

Âmbito – Nível de serviço do processo de gestão de serviços de segurança.

Intenção – Especificar e quantificar SLA.

Técnica – Os passos propostos para a especificação e quantificação de SLA para serviços de segurança são os seguintes:

- (i) A unidade empresarial define a sua visão empresarial;
- (ii) São identificados os factores críticos de impacto (CIFS) no negócio com base numa combinação de CIFS específicos da indústria, CIFS semelhantes e envolvimento da própria organização;

- (iii) Os activos são listados e prioritizados utilizando uma técnica de análise de impacto (CIFS), para a prioritização dos activos do negócio e a unidade de segurança de TI identifica activos críticos de TI que são necessários para suportar a actividade e identifica as propriedades de segurança desejadas para a TI;
- (iv) São preparados SLA pela unidade de segurança de TI e propostos à unidade de negócio;
- (v) Devido a preocupações orçamentais ou outras causas, a unidade comercial pode decidir aceitar um maior risco de certos produtos, em troca de um serviço de menor preço;
- (vi) Finalmente, o novo SLA deve ser revisto pela TI para se certificar de que os riscos mais elevados numa unidade de negócio não impõem altos riscos inaceitáveis para outras unidades empresariais. Isso pode acontecer por causa da interdependência dos processos entre unidades de negócio.

Crítica – Esta abordagem é centrada principalmente na ligação entre os requisitos de segurança e na visão empresarial que os motiva. Ela mostra como este modelo pode ser usado para priorizar os requisitos de segurança. Outros factores importantes relacionados com serviços TI estão ausentes neste modelo, porque este foca-se apenas nos aspectos de segurança.

Tal como definimos na secção 2.2.1 em que criamos uma Taxonomia para ITSM, apresentamos de seguida na **Tabela 7** a nossa classificação do trabalho.

	DOA	POT	SVC	SLA	VAL	LCC	SEC
Su et al., 2007	1	2	0	2	1	3	2

Tabela 7 - Classificação de Specifying Information Security

2.2.2.7 Enhanced Telecom Operations Map (eTOM)

[Forum, 2007]

Âmbito – Modelação de Processos de negócio para utilização pelos prestadores de serviços e outros, dentro da indústria das telecomunicações.

Intenção – Especificar modelos de desenho de processos.

Técnica – O eTOM (*enhanced Telecom Operations Map*) é um referencial de processos de negócio para a área de telecomunicações que disponibiliza modelos numa estrutura padrão, terminologia e sistema de classificação para descrever processos de negócio em blocos de construção. Fornece uma base para a aplicação de disciplina a nível empresarial e conhecimentos para o desenvolvimento de processos de negócio, ao longo de todo o processo de negócio.

Detalhe – eTOM começa com uma vista geral do conjunto da empresa, e define os processos empresariais dentro desta, numa série de agrupamentos hierárquicos designados de níveis usando a decomposição para desenvolver a estrutura. Define processos, descrições, entradas e saídas, bem como outros elementos-chave. A vista inicial designada por "*Conceptual Level*" mostra a estrutura global da empresa, conhecida como "Nível 0 – Agrupamento de Processos", a qual está dividida em:

- (i) Estratégia e Infraestruturas - Abrange a gestão do ciclo de vida do produto e o planeamento;
- (ii) Operações - Abrange o núcleo de gestão operacional;
- (iii) Gestão da empresa - Abrange a gestão corporativa ou o negócio.

Para ajudar no posicionamento desses processos (modelos fornecidos) dentro do seu contexto de uso dentro da empresa, o processo de modelação eTOM inclui agrupamentos de processos verticais e horizontais (Nível 1).

Os processos agrupados verticalmente estão relacionados com o cliente final, ou seja destinam-se a produzir um bem ou serviço (por exemplo, satisfação, fiabilidade e facturação).

Os processos agrupados horizontalmente representam funcionalidades interprocessos relacionadas com a forma como se trabalha e coopera, por exemplo, CRM (*Customer Relationship Management*). Devido aos diferentes focos nas Operações e Estratégia, Infra-estruturas e Produto, é feita uma separação também a este nível.

Alguns processos foram analisados e desenvolvidos em decomposição mais detalhadamente.

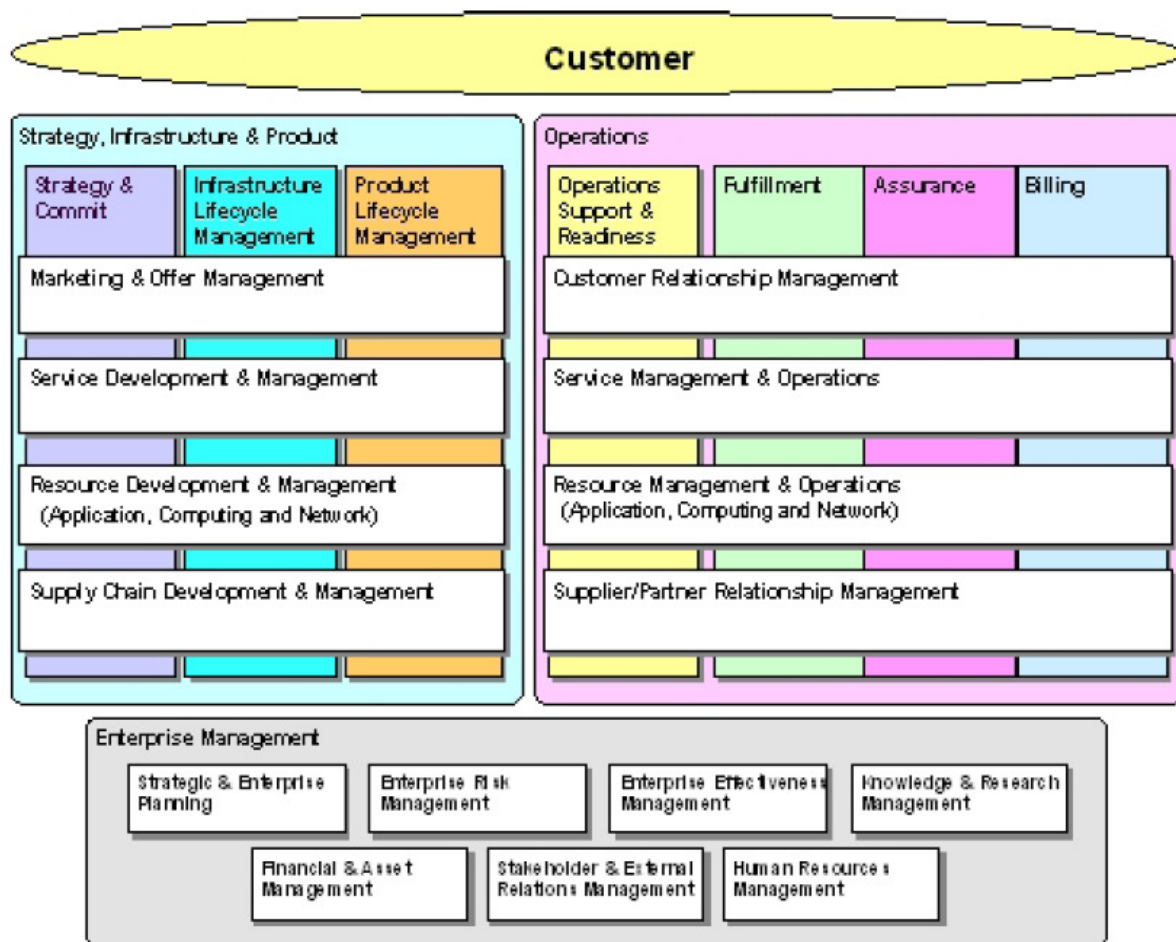


Figura 9 - Nível 1 eTOM, extraído de [Forum, 2007, Forum, 2006]

Crítica – O referencial eTOM inclui a definição para a estrutura do processo (hierarquia), decomposição de processos (níveis), fluxos de processo (ligações) e dinâmica do processo (comportamento). No entanto, a representação não é suficiente para o eTOM ser entendido como um modelo completo, uma vez que existe formalismo e diagramas que ajudem a compreender a informação. O TMForum (*TeleManagement Forum*), responsável pelo eTOM, deseja superar isso com o projecto SID (*Shared Information Data Model*) [Faurer, et al., 2003]. Não existe formalismo na modelação de processos. A sua representação não é feita por uma linguagem de modelação geralmente aceite como UML (*Unified Modeling Language*) ou BPMN.

Tal como definimos na secção 2.2.1 em que criamos uma Taxonomia para ITSM, apresentamos de seguida na **Tabela 8** a nossa classificação do trabalho.

	DOA	POT	SVC	SLA	VAL	LCC	SEC
e-TOM 7	2	2	2	2	1	1	1

Tabela 8 - Classificação de *Enhanced Telecom Operations Map (eTOM)*

2.2.2.8 ITIL SLM Versão 3.0

[Loyd and Rudd, 2007]

Âmbito – *Service Management*, entendida como um conjunto de capacidades organizacionais especializadas para fornecer valor aos clientes sob a forma de serviços na área de sistemas de informação, em particular gestão de infra-estruturas.

Intenção – Especificar modelos de desenho de processos.

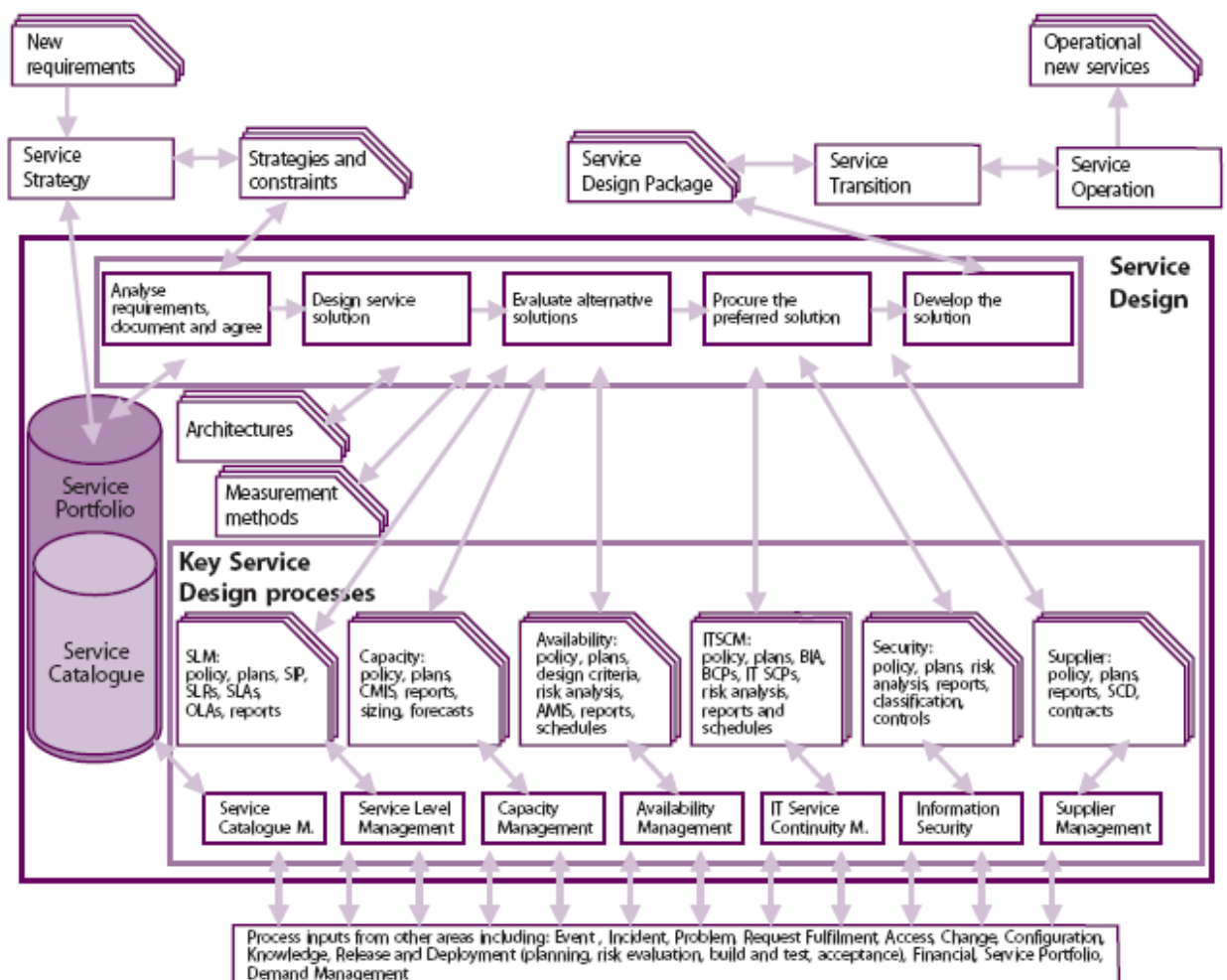


Figura 10 – SLM (Service Level Management) : Um dos principais processos de ITIL, extraído de [Loyd and Rudd, 2007]

Técnica - Existem 5 fases para gestão dos serviços ao longo do ciclo de vida do serviço (estratégia, desenho, transição, operações e serviços de melhoria contínua). Cada uma das fases tem processos como SLM (*Service Level Management*), como está ilustrado na **Figura 10**

SLM é o nome dado aos processos de planeamento, coordenação, elaboração, acordo, acompanhamento e elaboração de relatórios de SLA, e da revisão constante do serviço a fim de garantir que os resultados são os esperados, os custos justificáveis e, em termos gerais, que é um serviço de qualidade, mantido e melhorado gradualmente. É sob o processo de SLM que as metas são especificadas com os representantes da empresa e traduzidas para os requisitos e SLA.

Se os objectivos especificados no SLA forem adequados e reflectirem as exigências da empresa, o serviço prestado pelos prestadores de serviços estará alinhado com as necessidades de negócios e satisfará as expectativas dos clientes e utilizadores. O SLA é, por conseguinte, uma garantia no que diz respeito ao nível de qualidade dos serviços fornecidos pelo prestador do serviço. De realçar que para o próprio prestador de serviços é uma garantia que fornece os serviços adequados ao cliente face ao valor pago pelo mesmo.

O SLM é desenvolvido através do desenvolvimento do SLA e acordos de gestão a nível operacional, designados por OLA (*Operational Level Agreement*), com a TI interna ou através de contractos externos, a fim de garantir que os objectivos internos são alinhados com os objectivos do SLA.

O sucesso do SLM está dependente da qualidade do catálogo de serviços e seu conteúdo, porque fornecem a informação necessária sobre os serviços que devem ser geridos no âmbito do processo SLM.

Detalhe – A abordagem ITIL é muito elaborada. Para cada processo é especificado a sua finalidade, âmbito, valor para o negócio, as políticas aplicadas, actividades que o compõem e os principais métodos e técnicas empregados. O que desencadeia o início do processo de SLM, bem como as suas entradas, saídas e interfaces são também mencionados. Os principais indicadores de desempenho KPI (*Key Performance Indicator*) são delineados para monitorizar e medir QoS de serviços de TI.

Crítica - Embora ITIL v3 constitui um passo em frente na abordagem orientada para o serviço e na redução do fosso entre negócios e TI, há ainda uma falta de pormenor e de formalismo exigido para a avaliação das conformidades com SLA.

Tal como definimos na secção 2.2.1 em que criamos uma Taxonomia para ITSM, apresentamos de seguida na **Tabela 9** a nossa classificação do trabalho.

	DOA	POT	SVC	SLA	VAL	LCC	SEC
ITIL v3	1	2	2	2	1	3	2

Tabela 9 - Classificação de ITIL

2.2.3 Análise Comparativa e Lacunas ITSM

Existe falta de definição formal de serviços ou a utilização reduzida de SLA fora das áreas de telecomunicações ou gestão de redes. Mesmo que os trabalhos relacionados sejam escassos, delinear o actual estado-da-arte. Para o efeito, elaborámos um resumo dos trabalhos apresentados na tabela seguinte:

	DOA	POT	SVC	SLA	VAL	LCC	SEC
Niessink et al., 2000	1	1	0	2	0	2	2
Garsch. et al., 2001	3	2	2	2	1	3	1
Skene et al., 2004	3	3	2	3	2	0	2
OMG, 2006	1	3	0	1	3	0	3
Teyssié, 2006	0	2	0	3	1	0	3
Su et al., 2007	1	2	0	2	1	3	2
e-TOM 7	2	2	2	2	1	1	1
ITIL v3	1	2	2	2	1	3	2

Tabela 10 - Estudo comparativo de propostas ITSM

Uma ontologia genérica deve abranger todas as sete dimensões. Como podemos ver na Tabela 10, nenhuma das propostas revista faz isso. A proposta contida no capítulo 3 tem a intenção de atenuar esse problema.

2.3 Estado da Arte dos MetaModelos BPMN

Pretende-se estudar os metamodelos existentes para modelação de processos BPMN e concluir sobre as vantagens e desvantagens, a fim de compreender os conceitos envolvidos.

Os metamodelos estudados foram:

- (i) **BPMN STP_Eclipse**¹¹ – Trata-se de um metamodelo de um produto de modelação de processos a funcionar sobre o Eclipse, com a designação de *BPMN Modeler* [Foundation, 2009c].
- (ii) **OMG BPMN 1.2** – As especificações e modelos (classes e dependências) fornecidos pela OMG [OMG, 2009b].

Os conceitos de BPMN que são referidos, por exemplo *lane*, *pool*, *gateway*, actividade, evento etc., são apresentados e discutidos com exemplos no capítulo 4 Representação da Dinâmica.

2.3.1 Taxonomia para MetaModelos BPMN

Para se efectuar esse estudo foi feito um mapeamento dos conceitos e os mesmos agrupados em diferentes dimensões. Essas dimensões são:

- (i) *Swimlanes* – Este grupo inclui *Pools* e *Lanes*;
- (ii) *Activities & Transactions* – Este grupo inclui Actividades e Fluxos de Comunicação;
- (iii) *Gateways* – Este grupo inclui as decisões (excepto *Sequence Flow* condicionais);
- (iv) *Events* – Este grupo inclui os acontecimentos;
- (v) *Data & Documentation* – Este grupo inclui Artefactos;
- (vi) *Simulation* – Este grupo permite atribuir recursos a uma actividade ou processo e distribuir o fluxo de trabalho através das gates;
- (vii) *Use Case* – Este grupo inclui ligação dos processos aos casos de utilização, que incluía definição dos papéis (*roles*) dos actores e a sua responsabilidade nas unidades organizacionais;
- (viii) *Auxiliary & Structural* – Inclui todas os conceitos não incluídos nos outros grupos, como data, modelo, diagramas, parâmetros de entrada e saída de processos ou

¹¹ *Plug-in* do Eclipse que suporta a modelação em BPMN.

actividades, conteúdos de mensagens, etc., em resumo todos os conceitos complementares ou para além da utilização da notação gráfica de BPMN.

Para melhor entender as valências e limitações de cada modelo, será efectuada uma comparação crítica, global e grupo a grupo. Assim, para cada grupo indicamos:

- (i) Onde há concordância nos metamodelos;
- (ii) Onde é que os metamodelos aparentemente são discordantes;
- (iii) Quais são as consequências dessas discordâncias e porque é que isso é mais ou menos importante.

Tal como para os estudos de meta modelação para ITSM introduzimos uma taxonomia. Os três grandes critérios utilizados são:

- (i) Cobertura dos conceitos;
- (ii) Qualidade do metamodelo;
- (iii) Complexidade do modelo.

Para o efeito desenvolveram-se e analisaram-se as seguintes aproximações;

- (i) Número de classes – Critério de Dimensão da Cobertura;
- (ii) Número de associações – Critério de Complexidade;
- (iii) Número de heranças – Critério de Complexidade;
- (iv) Número de pacotes no modelo – Critério de Complexidade;
- (v) Coincidência de Classes, Enumerados e Grupos – Critério de Cobertura de Conceitos.

Foram listadas as classes, contadas e comparadas no conjunto e divididas por grupos. Associações e heranças foram contabilizadas. Daqui resultou a **Tabela 11** que apresentamos de seguida:

	OMG BPMN 1.2	BPMN STP_Eclipse
Pacotes	8	0
Classes	100	20
Associações	Não disponível ¹²	10
Especializações	99	24
Enumerados	45	37

Tabela 11 - Comparação de metamodelos

Notas à obtenção dos dados: Os números são aproximados e foram extraídos por métodos diferentes, e nalguns casos dependentes de alguma interpretação.

Para comparar os metamodelos que estudámos, desenvolvemos uma matriz em que a avaliação inclui as características qualitativas e quantitativas descritas num sistema baseado numa escala de maturidade, ou seja, um conjunto de descritores ao longo de diferentes dimensões definidas sobre uma escala ordinal [0,1,2,3] tal como se realiza no trabalho realizado na parte de estudos sobre ITSM, apresentado na secção 2.1.1 Taxonomia.

As dimensões de maturidade são as seguintes:

- (i) **Facilidade de Implementação (FIM);**
- (ii) **Tradução de Conceitos (CON);**
- (iii) ***Swimlanes* (LAN);**
- (iv) **Actividades (ACT);**
- (v) **Decisões (DEC);**
- (vi) **Eventos (EVE);**
- (vii) **Artefactos (ART);**
- (viii) **Simulação (SIM);**
- (ix) **Casos de utilização (USC);**
- (x) **Execução (EXE).**

Vamos nas secções seguintes analisar estas dimensões.

2.3.1.1 Facilidade de Implementação

A dimensão **FIM (Facilidade de Implementação)** representa o grau de facilidade de implementação do metamodelo no desenvolvimento de software. Quanto mais simples o

¹² O Meta modelo de BPMN disponibilizado pelo OMG, versão 1.2, não está detalhado, dos diagramas apenas costumam especializações C: Extracto para exemplo das especificações OMG BPMN 1.2.

metamodelo e com menos classes e relações, mais fácil será implementá-lo. No entanto menor será a sua versatilidade e capacidade de desenvolvimento futuro:

- 0 - Difícil – O modelo é altamente conceptual, com um alto número de associações e classes;
- 1 - Limitado – O modelo utiliza algumas soluções de compromisso, por exemplo algumas associações com o objectivo de tornar mais fácil a implementação;
- 2 - Fácil – O modelo utiliza várias soluções de compromisso;
- 3 - Muito Fácil – O modelo é muito operacional, apresentando-se muito *flat*, com poucas classes, poucos níveis de especialização e utiliza muitos enumerados.

2.3.1.2 Tradução de Conceitos

A dimensão **SEQ (Sequência)** representa o grau de detalhe da modelação no metamodelo estudado das especificações de BPMN 1.2 definidas pela OMG, em termos de sequência de fluxo nomeadamente orquestração e fluxos de sequência.

- 0 - Nada – Não segue as especificações do BPMN;
- 1 - Pouco – Traduz só as especificações básicas da OMG, ou seja permite a criação de processos mas sem serem muito sofisticados;
- 2 - Médio – O modelo utiliza mais especializações que no caso anterior, permite nomeadamente comunicação entre *pools* e fluxos de sequência condicionais;
- 3 - Muito – Traduz completamente as especificações da OMG, permite nomeadamente mensagens entre um largo número de objectos em diferentes *pools*.

2.3.1.3 Aplicação das Especificações da OMG - (LAN), (ACT), (DEC), (EVE), (ART)

As dimensões que vamos analisar de notação BPMN são as seguintes:

- (i) *Swimlanes* (LAN);
- (ii) Actividades (ACT);
- (iii) Decisões (DEC);
- (iv) Eventos (EVE);
- (v) Artefactos (ART).

Procuramos reflectir sobre o nível de tradução das especificações da OMG, ou seja, a capacidade dos metamodelos traduzirem as regras e instanciarem meta classes de acordo com as especificações:

- 0 - Nada – Este é um nível pouco provável de ser utilizado, já que todos os metamodelos devem traduzir alguns tipos básicos de objectos de BPMN;
- 1 - Pouco – Os objectos básicos podem ser criados, mas com bastantes limitações, quer quanto à combinação de alguns objectos, como atributos dos mesmos, ou ainda com ausência de regras;
- 2 - Médio – O modelo traduz mais especificações que no caso anterior;
- 3 - Muito – O modelo traduz completamente as especificações da OMG.

2.3.1.4 Simulação

A dimensão **SIM (Simulação)** representa o grau de inserção de dinamismo nos processos, o qual necessita de suporte no metamodelo, em termos de classes, associações etc., já que envolvem novos conceitos (especificamos alguns conceitos no secção 2.3.1.4 Simulação):

- 0 - Nada – Não dispõe de nenhum dos conceitos associados;
- 1 - Pouco – Alguns conceitos existem, por exemplo recursos;
- 2 - Médio – O metamodelo dispõe de estruturas para realizar a simulação e encaminhamento de processos ou seja condições sobre as gates, funções distribuição fluxos, etc.;
- 3 - Muito – Apresenta um vasto suporte para funções de simulação.

2.3.1.5 Casos de Utilização

A dimensão **USC (Casos de Utilização)** representa a ligação entre casos de utilização e processos:

- 0 - Nada – Não existe;
- 1 - Pouco – Existe, mas não foi muito desenvolvida;
- 2 - Médio – Um pouco mais desenvolvido que no caso anterior;

3 - Muito – Amplamente desenvolvida com múltiplas relações com elementos de BPMN e estruturais.

2.3.1.6 Tabela Resumo da Taxonomia BPMN

Nível	FIM	SEQ	LAN, ACT, DEC, EVE, ART	SIM	USC
0	Difícil	Nada	Nada	Nada	Nada
1	Limitado	Pouco	Pouco	Pouco	Pouco
2	Fácil	Médio	Médio	Médio	Médio
3	Muito Fácil	Muito	Muito	Muito	Muito

Tabela 12 - Tabela Resumo taxonomia BPMN

2.3.2 Trabalhos Relacionados BPMN

2.3.2.1 Metamodelo STP_Eclipse para BPMN

[Foundation, 2009c]

[Cibran, 2008]

Âmbito – Metamodelo de uma ferramenta de desenho BPMN *open source* na plataforma Eclipse.

Intenção – O seu objectivo é a implementação e funcionamento do BPMN *Modeler*. Este é um editor de diagramas de processo de negócio que existe desde 2006, foi desenvolvido como um componente da Plataforma SOA [Foundation, 2009b] e evoluiu como um subprojecto Eclipse em 2008. Utiliza o EMF (*Eclipse Modeling Framework*) [Foundation, 2009a].

Técnica – O metamodelo para BPMN do STP_Eclipse caracteriza-se por um elevado número de associações, com uma topologia em estrela, poucas relações de herança e preocupações orientadas para a implementação de uma ferramenta de desenho com a notação BPMN. Permite a criação de *lanes* e *pools*, e *lanes* dentro de *pools*. Permite a criação de actividades,

gateways, eventos, etc. Aliás, no STP_Eclipse a mesma classe *Activity* é utilizada para criar objectos para actividades, eventos e *gateways*, o que traz limitações (ver crítica) e não permite a criação de condições associada a uma *gateway* ou a noção de *gate*.

Crítica – A simplicidade do modelo mantém o tamanho do código gerado pelo *Model Development Tools* (MDT) sustentável, ou seja, não se torna demasiado grande e complexo. No entanto, introduz dificuldades à evolução do BPMN *Modeler* quando se pretende ir para além de ser editor gráfico de BPMN ou mesmo em relação a algumas especificações de detalhe da OMG. O modelo do STP_Eclipse utiliza enumerados para eventos, tarefas e *gateways* a partir da mesma classe *Activity*. A consequência é falta de flexibilidade no STP_Eclipse e algumas limitações, por exemplo, o metamodelo que encontramos não prevê tipo de Actividades do tipo *Looping* ou *Multiinstance*, tal poderia ser inserido aumentando-se o número de enumerados (*ActivityType*) ligados à classe *Activity*, no entanto pelas especificações do OMG cada actividade pode ter até dois tipos, por exemplo *loop* e *multiinstance*, com 2 símbolos gráficos em simultâneo, numa estrutura como utiliza o STP_eclipse implicava uma alteração na estrutura de classes. O desenho STP_Eclipse é mais difícil de compreender e evoluir, menos flexível portanto, embora para implementação dos casos mais simples seja mais rápido (existem menos classes e menos heranças).

A utilização de enumerados no STP_Eclipse para eventos, tarefas e *gateways* é feito a partir da mesma classe *Activity*, isto é particularmente sentido a nível dos eventos dado o elevado número de eventos que existem em BPMN, cerca de 25. No entanto tal como nas actividades, o verdadeiro problema são as características diferentes de cada evento que, numa eventual evolução, não se conseguem modelar sem alterar a estrutura de classes. O STP_Eclipse apresenta, mesmo só em termos de notação gráfica algumas dificuldades em relação às especificações do OMG. Outro exemplo é que os eventos do tipo *IntermediateEvent* devem poder ser ligados às actividades, até mais que um evento numa só actividade, e tal não está previsto.

Tal como definimos na secção 2.3.1 em que criamos uma Taxonomia para MetaModelos BPMN apresentamos na **Tabela 13** a nossa classificação para este trabalho;

	FIM	SEQ	LAN	ACT	DEC	EVE	ART	SIM	USC	EXE
STP_Eclipse	3	1	2	2	1	1	3	0	0	0

Tabela 13 - Tabela de classificação metamodelo STP_Eclipse BPMN

2.3.2.2 Especificações OMG 1.2

[OMG, 2009a]

Âmbito – Caderno de especificações BPMN criado pelo OMG versão 1.2.

Intenção – Especificações de BPMN. Normalizar o uso de BPMN e especificar as regras da linguagem BPMN. São descritas as classes, enumerados, dependências e restrições de integridade.

Técnica – O metamodelo da OMG preocupa-se com a definição dos conceitos e apresenta grande número de especializações. O metamodelo permite a criação de *lanes* e *pools*, assim como o conceito de que um processo pode ter várias *pools*. Implementam a noção de condição associada à *gateway* e a noção de *gate* (saída ou entrada da *gateway*). Estes dois conceitos são fundamentais para permitir a animação e simulação de processos.

Crítica – O Documento extenso e denso. São fornecidas apenas as classes, especializações sem outro tipo de associação o que muito ajudaria a compreender os conceitos, C: Extracto para exemplo das especificações OMG BPMN 1.2. O modelo, no caso do domínio BPMN, tenderá a ser muito complexo com vários graus de especialização e alguma dificuldade de implementação por via dessa complexidade. O Modelo da OMG é muito desenvolvido e como especializações de elementos de suporte. Apresenta mensagens (conteúdo e de quem para quem), entidade, categoria, evento, parâmetros de input e output para processos, etc., cujos objectivos são a execução de processos, motores de *workflow* e a utilização de BPEL (*Business Process Execution Language*)¹³.

Tal como definimos na secção 2.3.1 em que criamos uma Taxonomia para MetaModelos BPMN apresentamos na **Tabela 14** a nossa classificação para este trabalho;

13 BPEL (*Business Process Execution Language*) - É a designação curta de *Web Services Business Process Execution Language* (WS-BPEL).

	FIM	SEQ	LAN	ACT	DEC	EVE	ART	SIM	USC	EXE
BPMN 1.2	0	3	3	3	3	3	3	1	0	2

Tabela 14 - Tabela de classificação metamodelo OMG BPMN 1.2

NOTA: Na versão de Especificações de BPMN 2.0 Beta1 [OMG, 2009c] que surgiu em Agosto de 2009 (posteriormente à elaboração deste capítulo) o metamodelo é apresentado com maior detalhe e este é desenvolvido não tanto no sentido da introdução de mais conceitos a modelar (embora existem alguns novos conceitos de SLA, como escalonamento), mas, fundamentalmente, no sentido de execução de processos e mapeamento BPEL. O capítulo não foi actualizado devido à extensão das especificações, 316 páginas na versão 1.2, que exigiu vários meses de análise e 496 páginas na versão 2.0 Beta 1, que deverá ser trabalho futuro.

2.3.3 Análise comparativa e lacunas metamodelo BPMN

Como se viu no trabalho relacionado de meta modelação BPMN, não são muitas as publicações de Metamodelos nesta área. Os dois trabalhos encontrados perspectivaram características muito diferentes. O metamodelo de BPMN do STP_Eclipse caracterizou-se por uma total ausência de documentação excepto uma página com o diagrama que se apresenta no anexo “B : Metamodelo STP_Eclipse”. Por outro lado, o metamodelo da OMG BPMN 1.2 os diagramas encontram-se incompletos e existem 316 páginas de especificações escritas.

Apesar dos trabalhos relacionados serem escassos, procurámos olhar colectivamente para tirar algumas conclusões. Para o efeito, elaborámos um resumo dos trabalhos na **Tabela 15** que apresentamos de seguida:

	FIM	SEQ	LAN	ACT	DEC	EVE	ART	SIM	USC	EXE
STP_Eclipse	3	1	2	2	1	1	3	0	0	0
BPMN 1.2	0	3	3	3	3	3	3	1	0	2

Tabela 15 - Estudo comparativo de metamodelos BPMN

Ambos cobrem os conceitos básicos de BPMN: *Activities*, *Gateways*, *Data*, *Transactions*, *Documentation*, *Swimlanes* e *Events* e permitem orquestração. Os dois modelos apresentam uma abordagem diferente de implementação dos diferentes subtipos de meta classes BPMN. O modelo do STP_Eclipse utiliza enumerados para eventos, tarefas e *gateways* a partir da mesma classe *Activity*. O OMG BPMN 1.2 utiliza um grande número de classes e especializações (outro tipo de especificações é textual). A consequência é falta de flexibilidade no STP_Eclipse e algumas limitações, quer de notação, quer de regras de boa formação do modelo. Por exemplo, o STP_Eclipse BPMN não prevê actividades do tipo *Looping* ou *Multiinstance*. Tal poderia ser inserido aumentando-se o número de enumerados (*ActivityType*) ligados à classe *Activity*. O problema é que cada actividade, pelas especificações da OMG, pode ter até dois tipos, por exemplo *loop* e *multiinstance*, com 2 símbolos gráficos em simultâneo. Na estrutura do STP_eclipse não está previsto e implicava uma alteração na estrutura de classes.

O que foi dito nas *Activities* & *Transactions* aplica-se também nas *Gateways*. Aliás, no STP_Eclipse também a mesma classe é utilizada para criar objectos para actividades, eventos e *gateways*. Existem também algumas diferenças no nível de implementação dos modelos, enquanto o modelo do STP_Eclipse fica-se pela implementação gráfica das *gateways*, o metamodelo da OMG implementa a noção de condição associada à *gateway* e a noção de *gate* (saída ou entrada da *gateway*). Estes dois conceitos são fundamentais para permitir a animação e simulação de processos, mas a simulação de processos apenas existe na OMG e de uma forma incipiente, pois apesar de o fluxo poder ser dirigido através de gates, não existe outro tipo de conceitos ou seja, atribuir recursos a uma actividade ou processo, distribuir o fluxo de trabalho através das *gateways* e obter, assim, uma visão dinâmica, custos, tempos e ocupação de recursos.

A utilização de enumerados no STP_Eclipse para eventos, é feita novamente a partir da mesma classe *Activity*. Isto é particularmente sentido a nível dos eventos, dado o elevado número de eventos que existem em BPMN, cerca de 25. No entanto, tal como nas actividades, o verdadeiro problema são as características diferentes de cada evento que, numa eventual evolução, não se conseguem modelar sem alterar a estrutura de classes. O STP_Eclipse apresenta problemas em relação às especificações da OMG, mesmo só ao nível gráfico, os eventos do tipo *IntermediateEvent* devem poder ser ligados às actividades, até mais que um evento numa só actividade, e tal não está previsto.

Os metamodelos são idênticos em termos de *Data* e *Documentation*, ou seja capacidade de criar grupos, *data objects* e *text annotation*.

Em relação aos casos de utilização, deverá existir uma relação com os processos, no entanto nenhum dos metamodelos estudados estabelece uma ponte.

O desenho STP_Eclipse é mais denso do que o metamodelo da OMG BPMN 1.2, embora para implementação dos casos mais simples seja mais rápido (existem menos classes e menos heranças). O modelo da OMG tem especificações bastante completas e é constituído especialmente por classes e especificações textuais mas bastante extenso, torna-se de difícil operacionalização pelo seu grande número de meta classes, especializações e detalhe. Sentimos necessidade da criação de um metamodelo numa situação intermédia, não totalmente “*flat*” nem tão profundo, com mais regras que as especificações do STP_Eclipse BPMN mas menos detalhado que o do BPMN 1.2, que permita investigação e seja de fácil operacionalização, mas mantendo um alto nível de rigor em relação às especificações BPMN 1.2 da OMG.

Ontologia de Serviços e SLA (ITSM)

3. ONTOLOGIA GERAL DE SERVIÇOS E SLA (ITSM)	62
3.1 INTRODUÇÃO	62
3.2 DESCRIÇÃO DOS PACOTES	62
3.2.1 <i>Pacote de Participantes</i>	63
3.2.2 <i>Pacote de Serviços</i>	67
3.2.3 <i>Pacote de Entregas</i>	69
3.2.4 <i>Pacote de Suporte</i>	71
3.2.5 <i>Pacote de Arquitectura de Empresa</i>	73
3.3 RESTRIÇÕES OCL ONTOLOGIA ITSM	73
3.3.1 <i>Expressões Pacote Participantes</i>	74
3.3.2 <i>Expressões Pacote Serviços</i>	74
3.3.3 <i>Expressões Pacote Entregas</i>	75
3.3.4 <i>Expressões Pacote Suporte</i>	76
3.3.5 <i>Expressões Pacote Enterprise Architecture</i>	76
3.4 VALIDAÇÃO DA CONFORMIDADE DOS SLA	77
3.5 DISCUSSÃO DOS RESULTADOS ONTOLOGIA ITSM	79

Este capítulo apresenta uma Ontologia de Serviços TI genérica. São descritos os pacotes de participantes, entregas, serviços; suporte, arquitectura de empresas e resultados.

3. Ontologia Geral de Serviços e SLA (ITSM)

3.1 Introdução

Neste capítulo, propomos uma ontologia genérica de serviços de TI. Temos como objectivo a representação do conhecimento de forma precisa e partilhável. Para fins de modelação escolhemos a notação mais utilizada na modelação da indústria das TI nos dias de hoje. O UML proporciona a necessária formalização para regras e restrições (com OCL), e prevê também a necessária modularização para organizar um complexo domínio em várias partes independentes.

Vamos utilizar um pequeno exemplo ilustrado com dados para estabelecer um SLA de help desk e alojamento de um *site*. O objectivo é exemplificar e demonstrar a viabilidade da ontologia com criação de cenários. Para o efeito, apresentamos um instantâneo, primeiro caso a caso, após uma visão geral de todo o sistema. O exemplo apresentado irá permitir efectuar a verificação do cumprimento e violação das especificações do SLA.

3.2 Descrição dos Pacotes

A ontologia proposta está organizada por vários pacotes, como representado na **Figura 11** onde são mostradas as dependências. A modelação de integração de serviços com a arquitectura de empresa é brevemente abordada na secção 3.2.5.

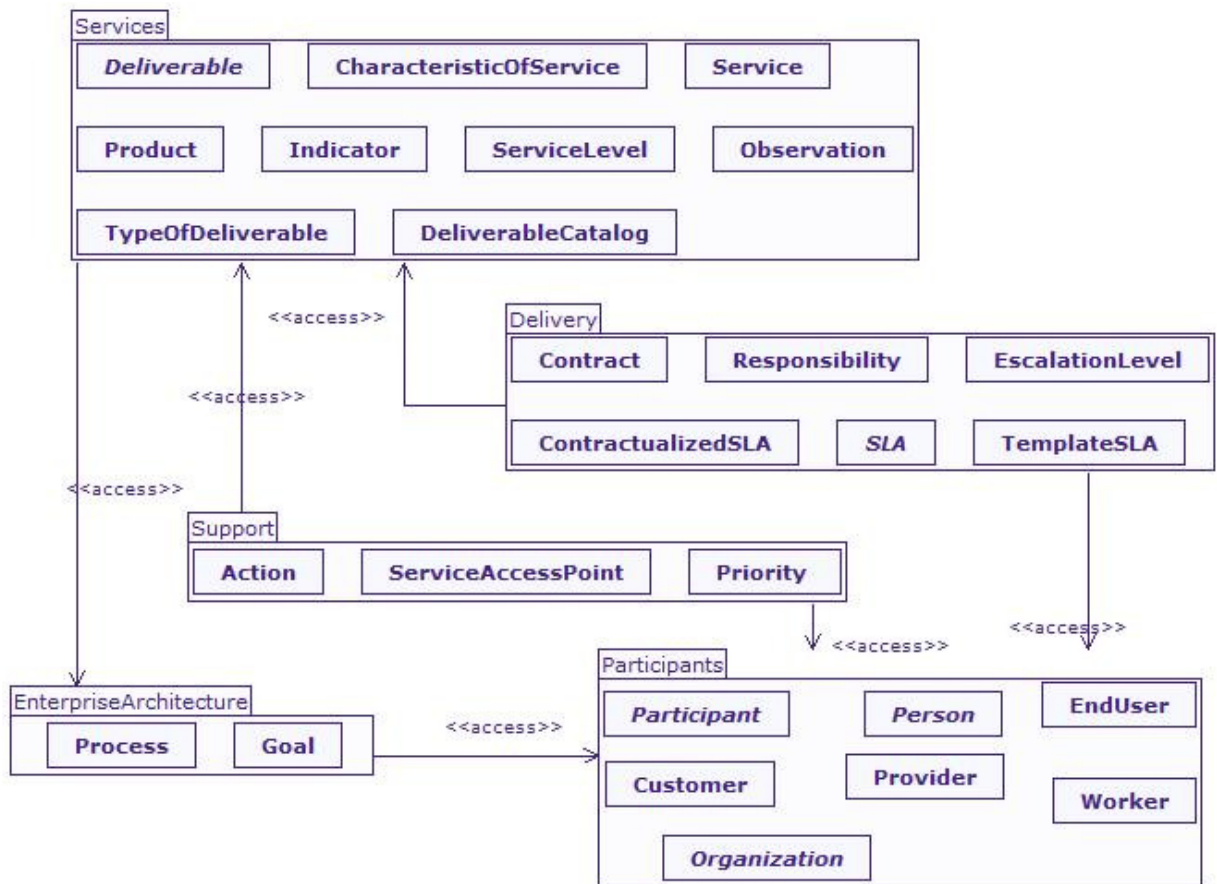


Figura 11 - Pacotes ITSM

O exemplo mostrado está simplificado, dado que é uma vista geral. Nas subsecções seguintes faremos uma breve descrição de cada um dos pacotes representados. São usados invariantes [OMG, 2003] para reforçar a consistência da ontologia.

3.2.1 Pacote de Participantes

Tal como representado na **Figura 12** os participantes podem ser indivíduos ou organizações. Estes últimos podem ser clientes ou fornecedores. O conceito de cadeia de fornecimento (subcontratação) é estabelecido no modelo apresentado, pela associação reflexiva na organização (*organization*). Os participantes podem ser os trabalhadores ou os utilizadores finais e podem pertencer ou não a uma organização. Por exemplo, um cliente em casa, usando um serviço de *video-on-demand*, seria modelado como um utilizador final.

Uma organização pode oferecer vários pontos de acesso ao serviço. Vários trabalhadores do fornecedor podem ser colocados num determinado ponto de acesso.

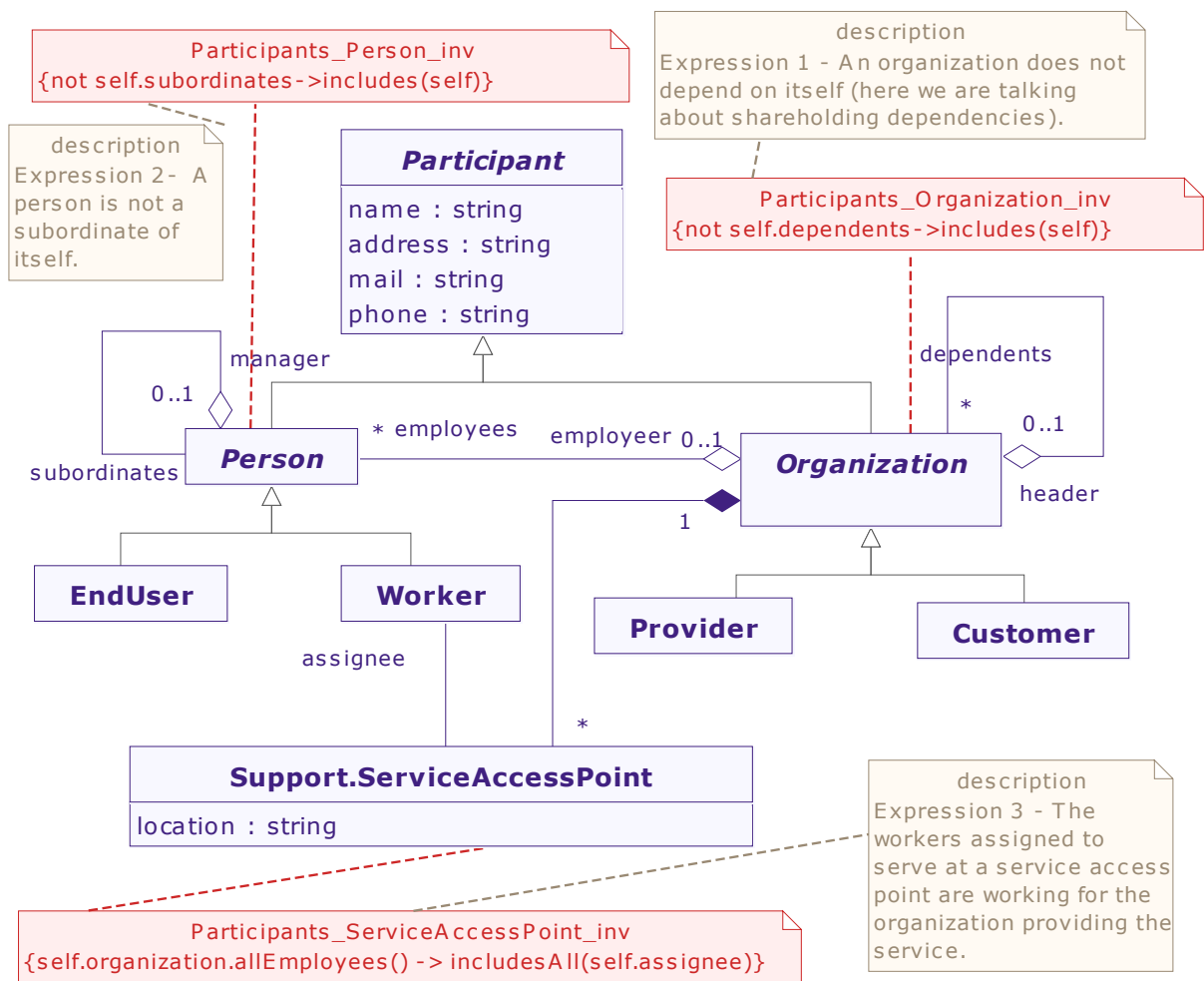


Figura 12 - Pacote Participantes ITSM

Para teste da Ontologia e criação do modelo de objectos criamos um exemplo de um de um contracto de SLA que mostramos de seguida na **Figura 13**, e que iremos descrever por partes nas secções seguintes.

3. Ontologia ITSM

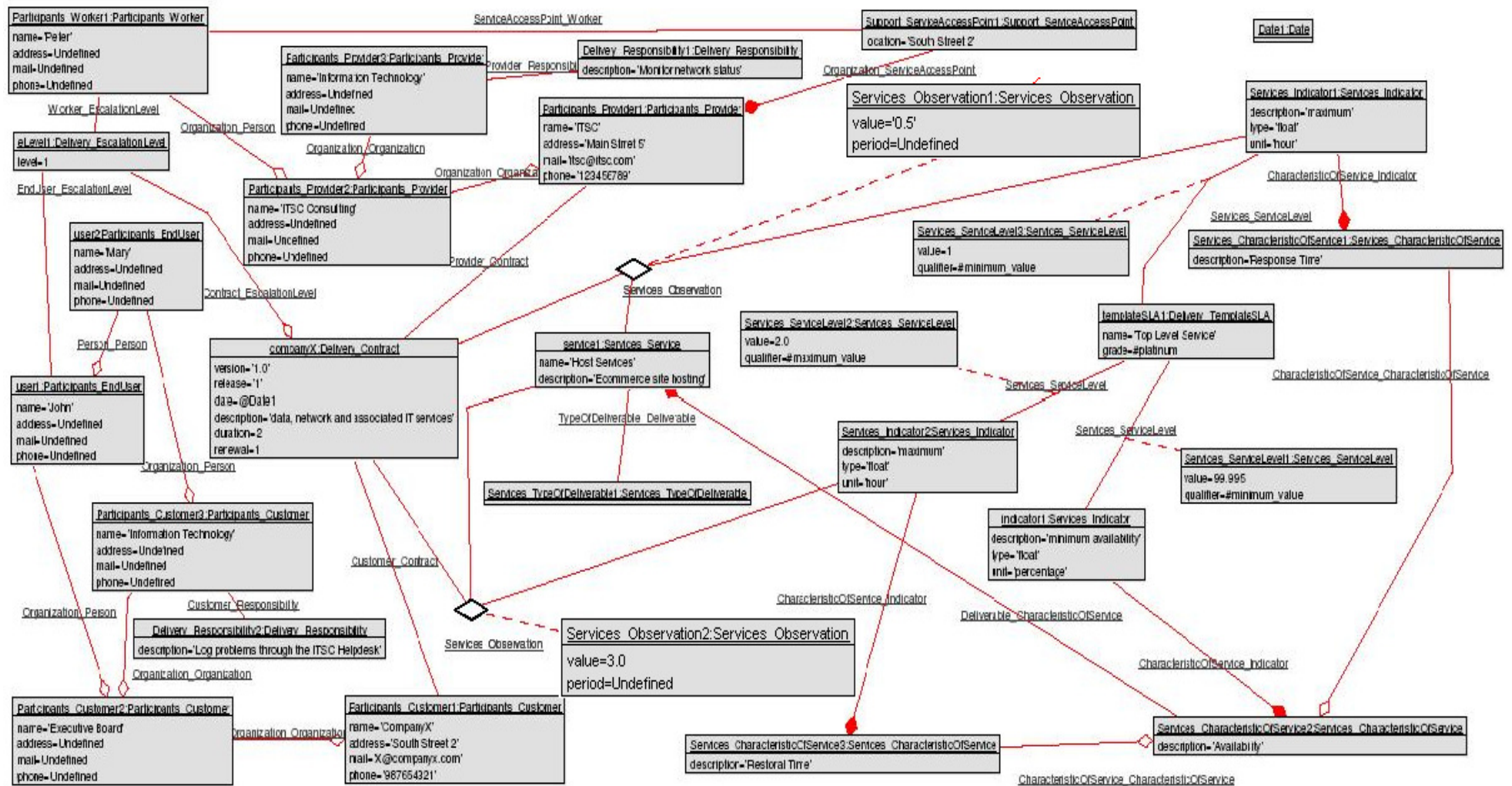


Figura 13 – Modelo de objetos de um contrato de SLA

No cliente existem duas pessoas, Mary e John que trabalham no departamento de TI. O John é o CEO (*Chief Executive Officer*). John é responsável pela resolução de problemas ao nível 1 de escalonamento do SLA, ilustramos esta estrutura na **Figura 14**

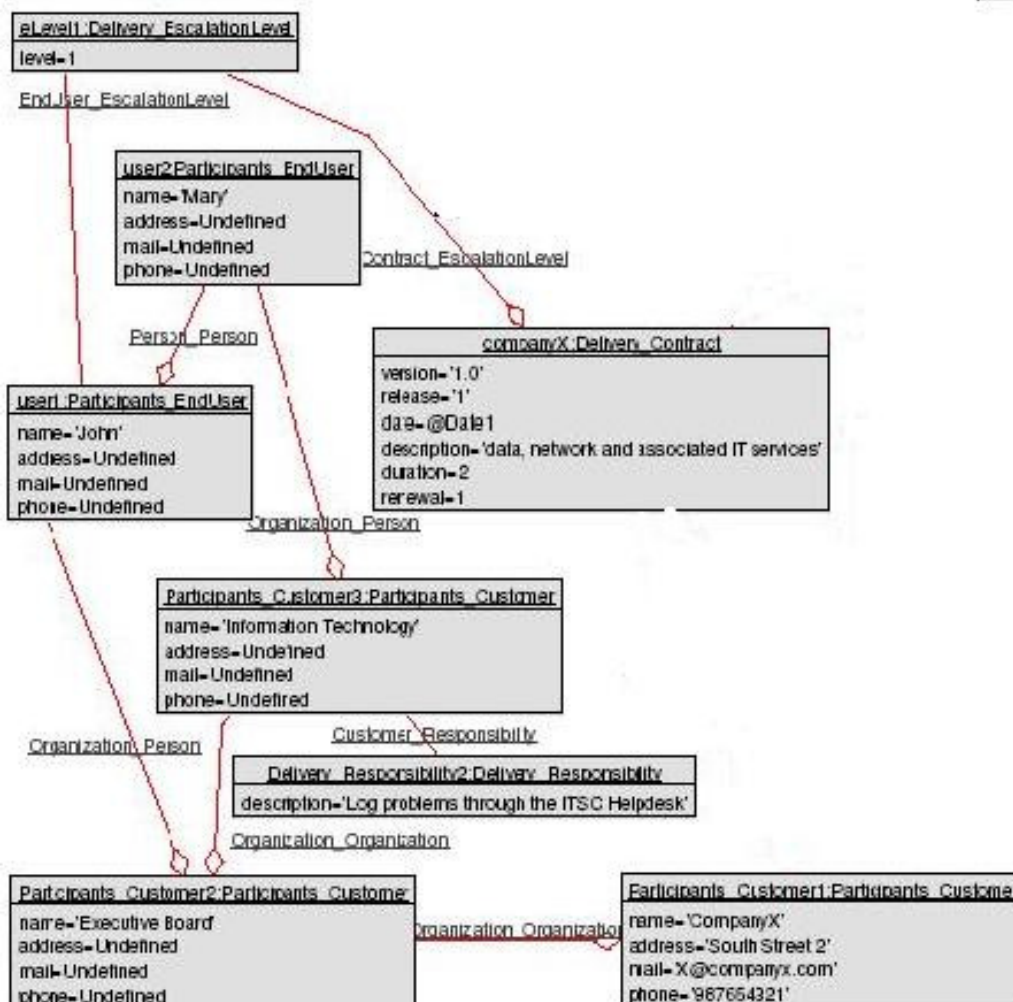


Figura 14 - Participantes do fornecedor no SLA

No lado do fornecedor existe uma unidade de negócio denominada ITSC Consulting, com um responsável chamado Peter e um departamento de TI. O departamento de TI tem a responsabilidade de acompanhar o *status* da rede, mas não é o responsável por Peter, tal como é ilustrado na **Figura 15**.

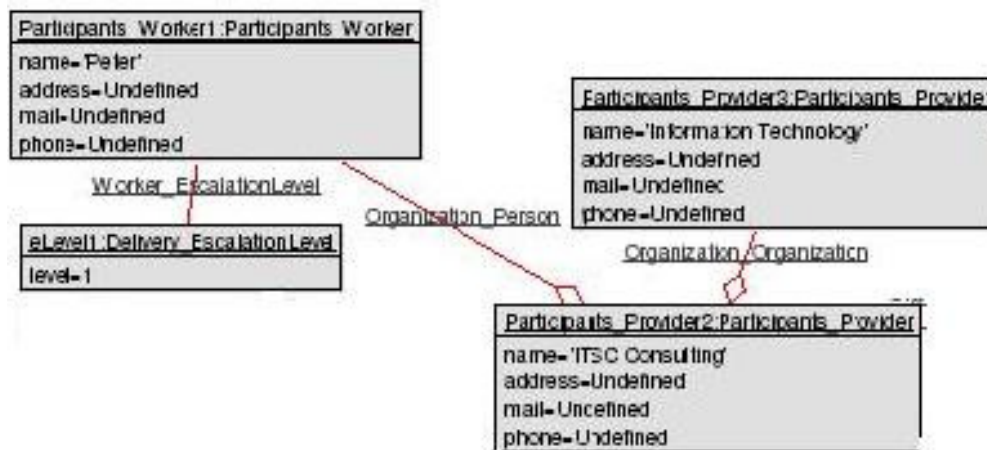


Figura 15 - Participantes do cliente no SLA

3.2.2 Pacote de Serviços

Uma vez que os serviços podem ser associados com entregas de produtos, generalizamos o conceito de entregáveis para incluir quer os serviços, quer os produtos, apresentamos na **Figura 16** o correspondente diagrama de classes.

Os entregáveis podem:

- I. Ser escolhidos a partir de um catálogo de serviços;
- II. Ser compostos hierarquicamente;
- III. Ser categorizadas em função de um conjunto de tipos.

Os entregáveis podem ser definidos por um conjunto de características (por exemplo, desempenho e disponibilidade). Estas características podem ser decompostas hierarquicamente e quantitativamente e são avaliadas por meio de um conjunto de indicadores (por exemplo, resposta a tempo de inatividade). Cada SLA define os valores aceitáveis para algumas estatísticas sobre os indicadores calculados. Esses valores são comparados com aqueles obtidos a partir das observações dos indicadores sobre o mesmo contexto do SLA, quando o serviço está a ser prestado.

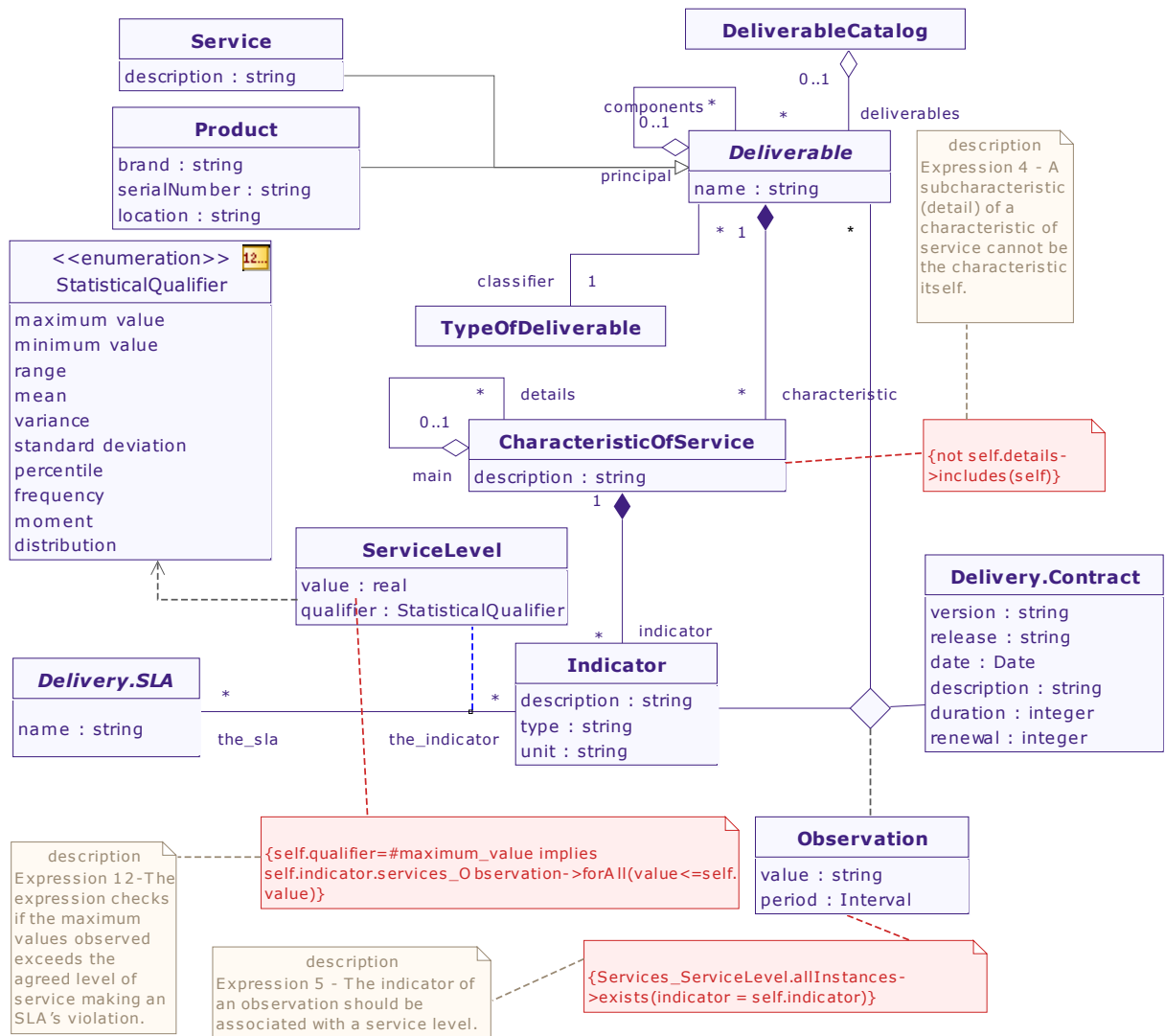


Figura 16 - Pacote Serviços ITSM

Ilustramos na **Figura 17** um exemplo de uma prestação de serviços da ITSC Consulting que se refere ao alojamento com características como disponibilidade (a acessibilidade do *site*) e sub-características como tempo de reposição do sistema (o tempo necessário para restaurar o serviço depois de uma pausa para baixo), e tempo de resposta (tempo entre o final de uma consulta em um computador e o início de uma resposta). Todas as características do serviço têm um indicador associado que define o nível de serviço. Por exemplo, a disponibilidade do indicador é a disponibilidade mínima, e é definido no contrato que o site estar disponível pelo menos, 99.995% do tempo. No caso de falha de serviço este deve ser restabelecido no máximo em duas horas e o tempo de resposta deve ser de 1 milissegundo.

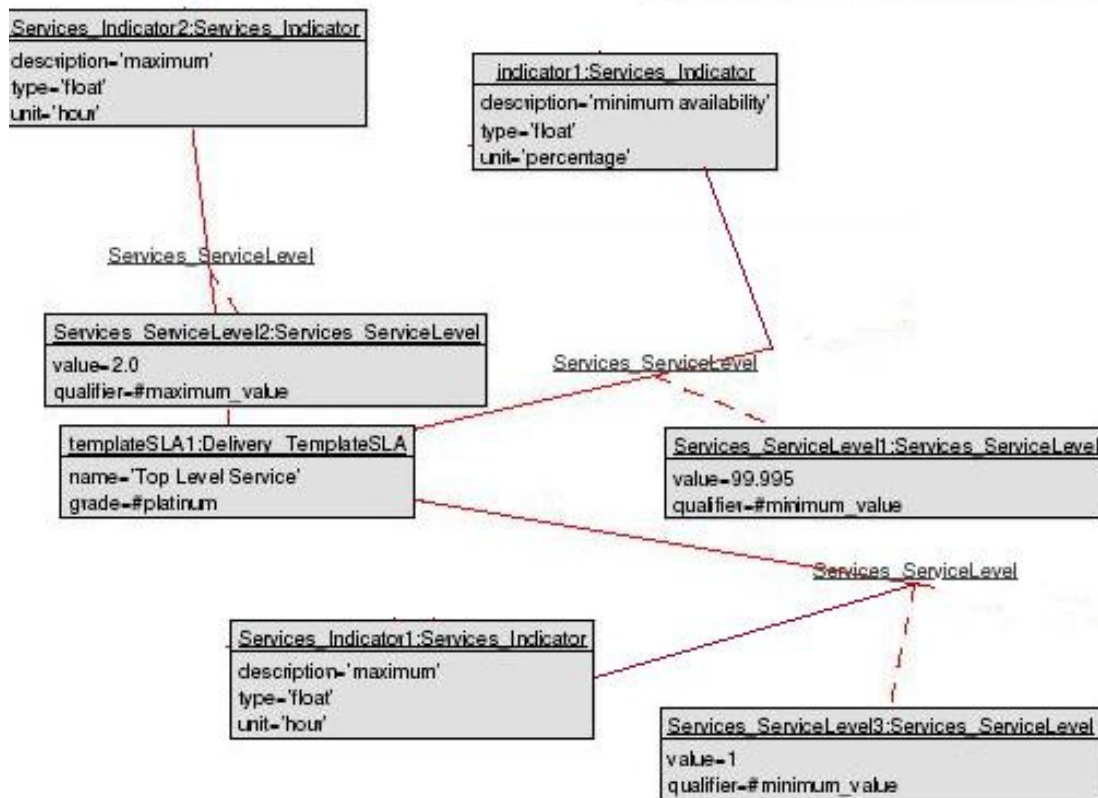


Figura 17 - Características do Serviço

3.2.3 Pacote de Entregas

Apresentamos na **Figura 18** o diagrama de classes que se refere a este pacote. Normalmente, os criadores de um serviço definem um conjunto de SLA padronizados, por exemplo, criando diferentes alternativas comparando o custo e o benefício (por exemplo, serviços platina, ouro, prata e bronze).

Estes SLA padronizados podem ser usados como “*templates*” no estabelecimento de um contrato entre um cliente e um fornecedor.

Um conjunto de resultados pode ser explicitamente incluído (automaticamente ou sob pedido) ou excluídos de um contrato. Todos os serviços abrangidos por um SLA têm responsabilidades que devem ser assumidas pelo cliente ou pelo fornecedor. O escalonamento fornece uma base, tanto para o prestador, como para os clientes, para a resolução de conflitos.

3. Ontologia ITSM

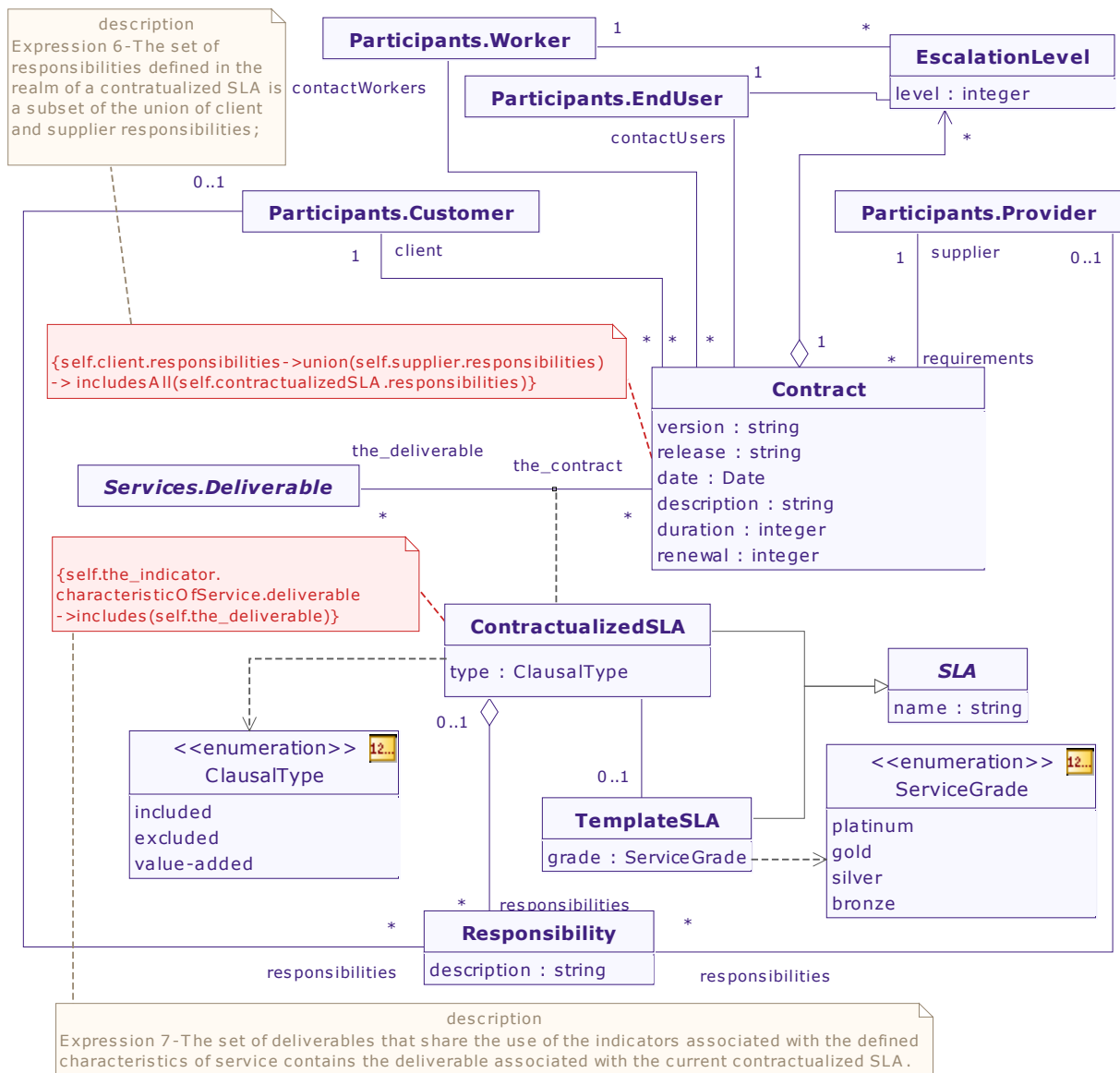


Figura 18- Pacote de Entregas ITSM

No nosso exemplo, tal como representado na **Figura 19**, a Empresa “Company X” fez um contrato de SLA sobre os dados, redes e serviços de TI associados especificamente à hospedagem de um site de comércio electrónico com ITSC (*Information Technology Supply Company*) por 2 anos, e a renovação é feita após 1 ano.

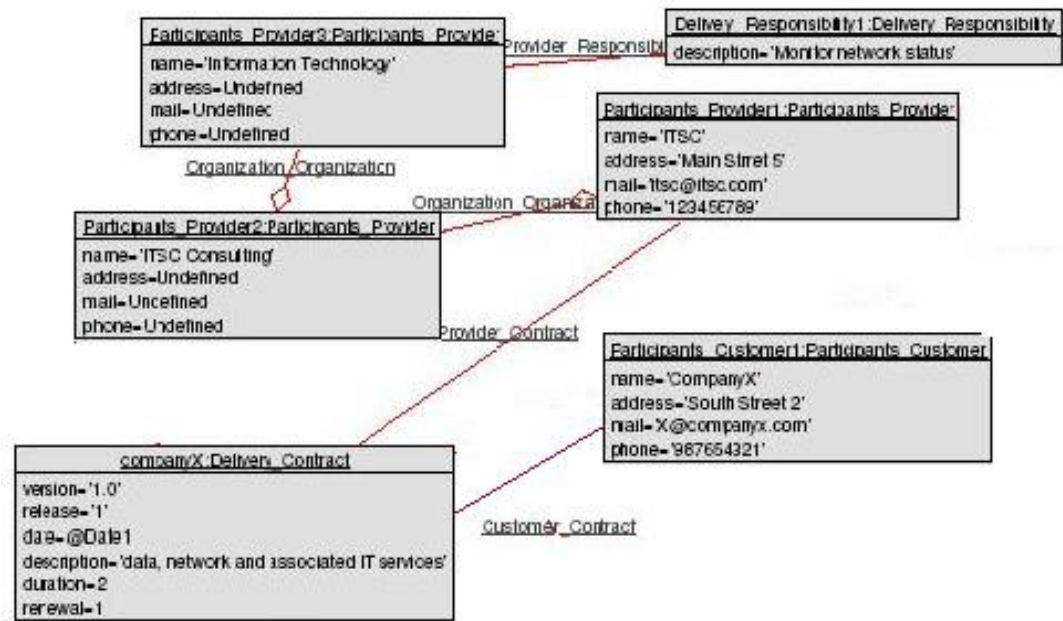


Figura 19 - Criação de um contrato de SLA

Peter também é definido como a pessoa para resolver problemas no nível 1 (existe um mecanismo para escalar os problemas para as pessoas no SLA, tal como ilustramos na **Figura 20**), da parte do fornecedor.

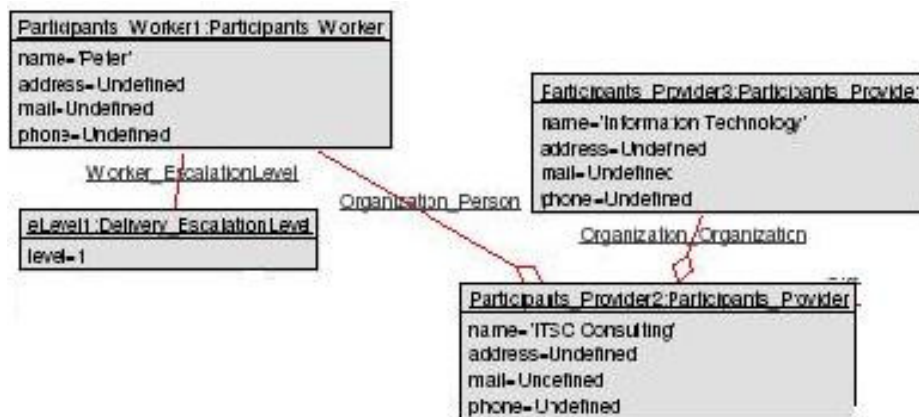


Figura 20 – Escalonamento dos problemas

3.2.4 Pacote de Suporte

Um ponto de acesso (por exemplo, *call center*, *help-desk*, ou sistema *on-line* de *trouble-tickets* (resolução de problemas ou assistência) pode lidar com vários entregáveis e o mesmo entregável ser tratado em vários pontos de acesso. Os utilizadores finais podem gerar acções (por exemplo, incidentes, sugestões de melhoria, consultas para a assistência) utilizando um serviço de ponto de acesso. Essas acções também podem ser recebidas e priorizadas por

um trabalhador, que também regista a acção, resolução ou adiamento, afectando assim o estatuto (referimo-nos ao estado da resolução do problema, por exemplo novo, aberto, resolvido, etc.) da acção como se pode ver na **Figura 21, Action**.

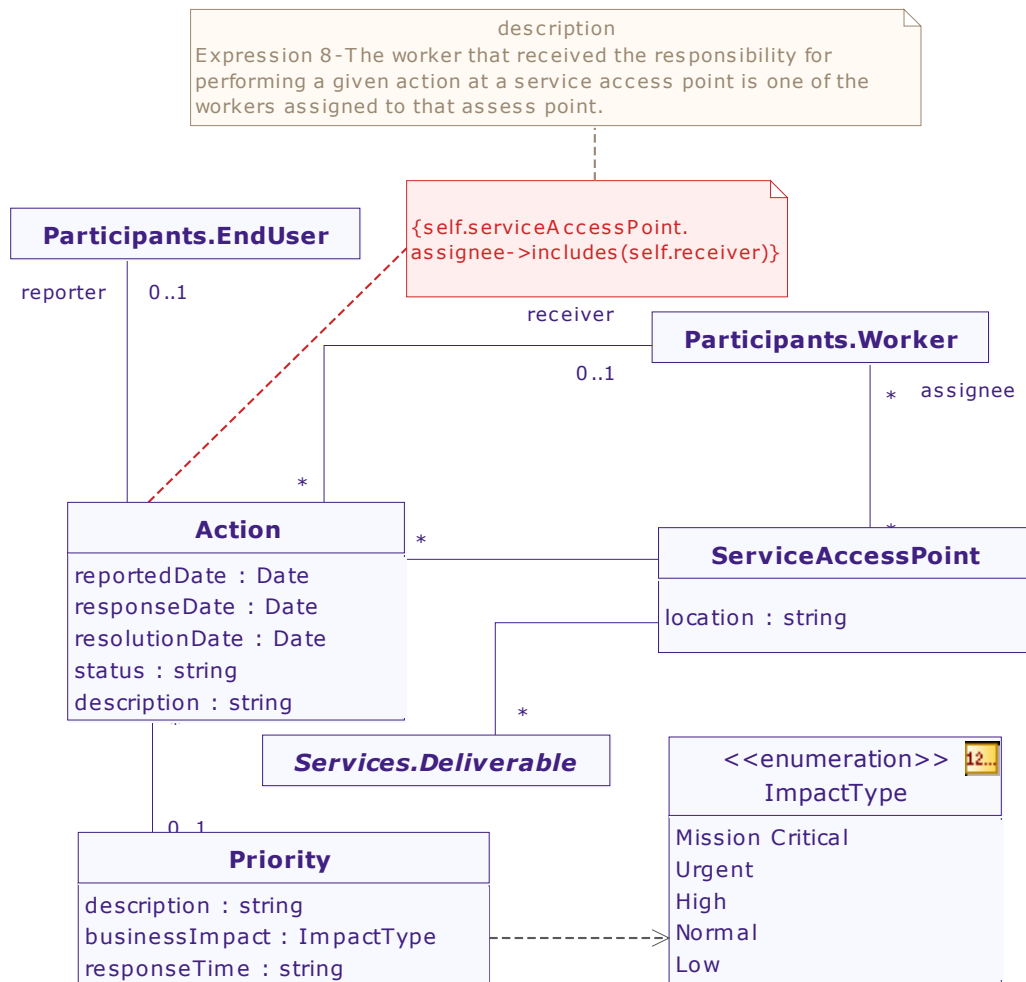


Figura 21 - Pacote de Suporte ITSM

No nosso exemplo Pedro trabalha no ponto de acesso de serviço que é *helpdesk*, cujo endereço é *South Street 2*, exemplo na **Figura 22**.

```

Support ServiceAccessPoint :Support ServiceAccessPoint
location='South Street 2'
    
```

Figura 22 - Service Access Point

3.2.5 Pacote de Arquitetura de Empresa

Os objectivos organizacionais promovem a implementação de processos que devem ser alinhados com esses objectivos. Os processos podem ter interdependências que podem ser representadas como um grafo dirigido. Os processos consomem alguns entregáveis (*inputs*) e produzem saídas (*outputs*). Quer os processos quer os resultados podem ser decompostos hierarquicamente. Na **Figura 23** apresentamos o diagrama de classes.

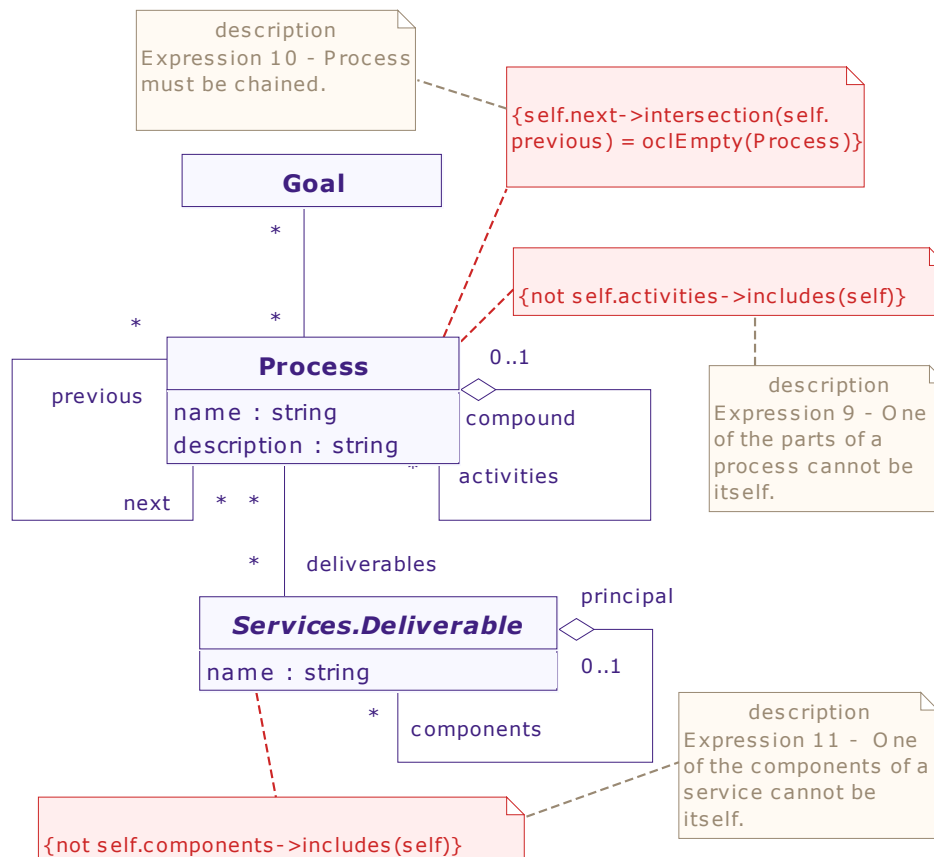


Figura 23 - Pacote de Arquitetura de Empresa ITSM

3.3 Restrições OCL Ontologia ITSM

As vantagens de utilizar OCL são aumentar a precisão através da redução da ambiguidade nas especificações, resultando em melhor (inequívoca) comunicação entre as partes envolvidas.

O OCL permite expressar três tipos de restrições sobre as partes do modelo: invariantes, pré-condições e pós-condições. OCL é uma linguagem declarativa cujas expressões não alteram o estado dos objectos. Estas expressões podem variar de

comparações simples (por exemplo, um atributo com um limite superior) para navegações complexas em um diagrama de classe (da nossa ontologia) através de suas associações. O contexto de pré e pós-condições e nos invariantes é a Classe onde as mesmas são definidas. É possível verificar as expressões de validade durante a modelação utilizando ferramentas como USE (*UML based Specification Environment*) [Group, 2008]. Para obter a descrição da sintaxe e semântica da OCL, consulte [OMG1997].

3.3.1 Expressões Pacote Participantes

Expressão 1 - Uma organização não depende só de si (depende dos *stakeholders*);

```
context Organization
inv Participants_Organization_inv :
    not self.dependents->includes(self)
```

Expressão 2 - Uma pessoa não é um subordinado de si mesmo;

```
context Person
inv Participants_Person_inv :
    not self.subordinates->includes(self)
```

Expressão 3 - O trabalhador que recebeu a responsabilidade de executar uma determinada acção em um ponto de acesso ao serviço, é um dos trabalhadores afectos a esse ponto de serviço;

```
context Support_Action
inv Support_Action_inv :
    self.serviceAccessPoint.assignee->includes(self.receiver)
```

3.3.2 Expressões Pacote Serviços

Expressão 4 - A sub-característica (detalhe) de uma característica do serviço não pode ser a própria característica;

```
context CharacteristicOfService  
inv Services_CharacteristicOfService_inv :  
    not self.details->includes(self)
```

Expressão 5 - O indicador de uma observação deve ser associado a um nível de serviço;

```
context Services_Observation  
inv Services_Observation_inv :  
    Services_ServiceLevel.allInstances->exists(indicator = self.indicator)
```

Expressão 12 – Esta expressão verifica se os valores máximos observados excedem o valor para o nível de serviço acordado, realizando ou não uma violação do SLA (Ver secção 3.4 Validação da Conformidade dos SLA);

```
context Services_ServiceLevel  
inv Services_ServiceLevel_inv :  
    self.qualifier=#maximum_value implies  
    self.indicator.services_Observation->forAll(value<=self.value)
```

3.3.3 Expressões Pacote Entregas

Expressão 6 - O conjunto de responsabilidades definidas no âmbito de um SLA contratado é um subconjunto da união das responsabilidades de cliente e fornecedor;

```
context Contract  
inv Delivery_Contract_inv:  
    self.client.responsibilities->union(self.supplier.responsibilities)->  
    includesAll(self.delivery_ContractualizedSLA.responsibilities)
```

Expressão 7 - O conjunto de entregáveis que partilham o uso de indicadores associados com as características definidas de serviço contém a prestação associada ao actual SLA contratualizado;

```
context ContractualizedSLA  
  
inv Delivery_SLA_inv :  
    self.the_indicator.characteristicOfService.deliverable ->includes(self.the_deliverable)
```

3.3.4 Expressões Pacote Suporte

Expressão 8 – Os trabalhadores designados para servir em um ponto de acesso de serviço, estão a trabalhar para a organização que presta o serviço;

```
context ServiceAccessPoint  
  
inv Support_ServiceAccessPoint_inv :  
    self.organization.allEmployees() -> includesAll(self.assignee)
```

3.3.5 Expressões Pacote Enterprise Architecture

Expressão 9 - Uma das partes de um processo não pode ser ele mesmo;

```
context Process  
  
inv EnterpriseArchitecture_Process_inv :  
    not self.activities->includes(self)
```

Expressão 10 - O Processo necessita de fazer parte de uma cadeia.

```
context Process  
  
inv Process_inv2 :  
    self.next->intersection(self.previous) =o clEmpty(Process))
```

Expressão 11 - Um dos componentes do serviço não pode ser ele mesmo.

```
context CharacteristicOfComponents  
  
inv Services_CharacteristicOfComponents_inv :  
    not self.components->includes(self)
```

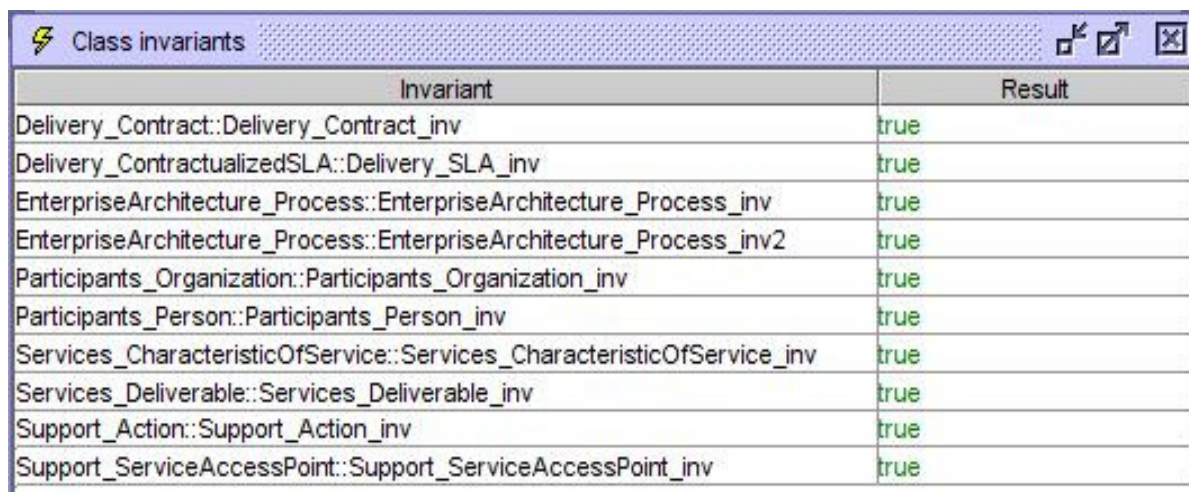
Nota: A **Expressão 12** encontra-se no pacote de Serviços, secção 3.3.2.

3.4 Validação da Conformidade dos SLA

Nesta secção vamos instanciar as entidades da ontologia pertencentes a três dos pacotes descritos nas secções anteriores, utilizando diagramas de objectos UML.

Sob os termos do contracto, o cliente e o fornecedor têm uma lista de obrigações que devem cumprir. O contracto tem uma descrição detalhada dos serviços de suporte e responsabilidades atribuídas ao fornecedor, ou seja, SLA de informação, serviços de rede, etc. Cada um dos serviços tem características. Por exemplo, serviços de rede, tem características como horas de operação ou disponibilidade dos serviços.

As expressões OCL que descrevemos na secção 3.3 são usadas para verificar se a instanciação da ontologia não viola as suas regras de consistência. Para o efeito utilizamos a ferramenta USE que com base nas expressões OCL, no diagrama de classes e corresponde modelo de objectos verifica a conformidade das expressões, como ilustramos com um *snapshot* na **Figura 24**.



Invariant	Result
Delivery_Contract::Delivery_Contract_inv	true
Delivery_ContractualizedSLA::Delivery_SLA_inv	true
EnterpriseArchitecture_Process::EnterpriseArchitecture_Process_inv	true
EnterpriseArchitecture_Process::EnterpriseArchitecture_Process_inv2	true
Participants_Organization::Participants_Organization_inv	true
Participants_Person::Participants_Person_inv	true
Services_CharacteristicOfService::Services_CharacteristicOfService_inv	true
Services_Deliverable::Services_Deliverable_inv	true
Support_Action::Support_Action_inv	true
Support_ServiceAccessPoint::Support_ServiceAccessPoint_inv	true

Figura 24 - Modelo de conformidade lógica OCL

Há um sub-modelo de qualidade, composto de indicadores e valores observados, (que são introduzidos no modelo de forma manual ou automática) e os valores permitidos, tal como representado na **Figura 25**.

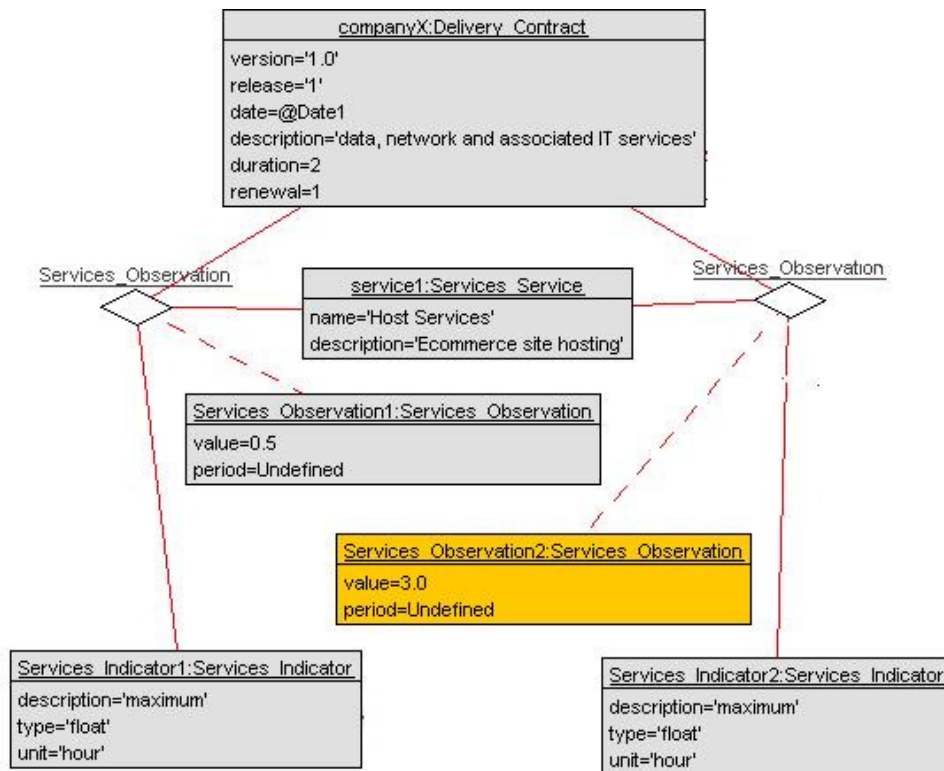


Figura 25 - Valores Observados para o SLA

Os valores observados para este exemplo são exibidos na **Tabela 16**.

	Direção	Permitido	Observado
Tempo de reposição de Serviço	Máximo	2	0.5

Tabela 16 – Valores observados e permitidos no SLA para o exemplo “em conformidade”

Para verificar se os valores observados estão no nível de serviço contratado é observada a expressão 12 (ver secção 3.3.2). O resultado é verdadeiro para o exemplo dado, que é um tempo observado de reposição de serviço de 0,5, como se mostra na **Figura 26**.

Class invariants	
Invariant	Result
Services_Observation::Services_Observation_inv	true

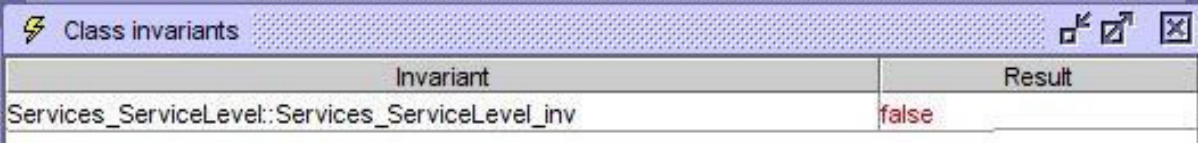
Figura 26 - Invariante OCL 12, resultado da validação OK

No caso de uma violação de SLA, os valores observados para o mesmo exemplo são mostrados na **Tabela 17**;

	Direcção	Permitido	Observado
Tempo de reposição de Serviço	Máximo	2	3.0

Tabela 17 - Valores observados e permitidos no SLA para o exemplo “não conforme”

Para um valor observado de 3.0 na expressão 12 os valores observados não são compatíveis com o nível de serviço contratado, como se verifica no *snapshot* ilustrado na **Figura 27**.



Invariant	Result
Services_ServiceLevel::Services_ServiceLevel_inv	false

Figura 27 - Invariante OCL 12, resultado falhou

3.5 Discussão dos Resultados Ontologia ITSM

Efectuamos uma proposta de ontologia ITSM, que:

- I. Tem a intenção de cobrir a grande maioria dos domínios de TI relevantes, tais como seus participantes (por exemplo, clientes e fornecedores), definição de SLA, prestação de serviços de apoio ao serviço ou a integração da arquitectura de empresa;
- II. Define os serviços e SLA de forma precisa com a ajuda da linguagem de restrições OCL;
- III. Foi feita uma validação preliminar, com a instanciação de um exemplo que esclarece a semântica da ontologia;
- IV. Abrange a maior parte do ciclo de vida de serviços, desde a sua criação à sua implementação;
- V. Permite a definição formal de verificação da conformidade do serviço, utilizando OCL.

[Esta página foi intencionalmente deixada em branco]

Representação da Dinâmica

4.	REPRESENTAÇÃO DA DINÂMICA.....	82
4.1	INTRODUÇÃO	82
4.2	DESCRIÇÃO DOS PACOTES.....	84
4.2.1	<i>Pacote Modelo.....</i>	85
4.2.2	<i>Pacote Elementos Estruturais.....</i>	88
4.2.3	<i>Pacote Elementos de Suporte.....</i>	90
4.2.4	<i>Pacote Conectores.....</i>	91
4.2.5	<i>Pacote Objectos de Fluxo.....</i>	95
4.2.6	<i>Pacote Actividade.....</i>	96
4.2.7	<i>Pacote Gateway.....</i>	98
4.2.8	<i>Pacote Eventos.....</i>	100
4.2.9	<i>Pacote Artefactos.....</i>	101
4.2.10	<i>Pacote Orquestração.....</i>	104
4.2.11	<i>Pacote Simulação.....</i>	107
4.2.12	<i>Pacote Conexão Interna.....</i>	108
4.2.13	<i>Pacote Casos de Utilização.....</i>	111
4.3	RESTRICÇÕES DO METAMODELO.....	112
4.3.1	<i>Expressões do Pacote Modelo.....</i>	112
4.3.2	<i>Expressões Pacote Elementos Estruturais.....</i>	113
4.3.3	<i>Expressões do Pacote Connectores.....</i>	113
4.3.4	<i>Expressões do Pacote Actividade.....</i>	114
4.3.5	<i>Expressões do Pacote Gateway.....</i>	114
4.3.6	<i>Expressões do Pacote Event.....</i>	115
4.3.7	<i>Expressões Pacote Use Case.....</i>	117
4.4	DISCUSSÃO DOS RESULTADOS METAMODELO BPMN.....	117
4.5	VALIDAÇÃO E CONTEXTO METAMODELO BPMN.....	118
4.6	DISCUSSÃO DOS RESULTADOS RELATIVOS AO METAMODELO DE BPMN.....	121

Este capítulo propõe um metamodelo de BPMN capaz de representar componentes da representação dinâmica incluindo as seguintes componentes; diagrama, processo, elementos de BPMN, orquestração, simulação e ligação aos casos de utilização.

4. Representação da Dinâmica

4.1 Introdução

O objectivo da linha de investigação em que se enquadra o trabalho desenvolvido nesta dissertação é o de suportar a modelação de processos de ITSM tais como processos de suporte de serviços e entrega de serviços, com especial ênfase na gestão de níveis de serviço. Em particular, pretendemos modelar tudo o que possa ser definido como utilizando SLA ou que precise de ser definido para implementação do SLA. A representação estática dessa informação foi já conseguida com a proposta da ontologia apresentada no capítulo precedente. Neste capítulo propomos um metamodelo para a linguagem de modelação de processos BPMN (*Business Process Modeling Notation*). Esta linguagem é usada para a modelação da dinâmica dos serviços de ITSM e, como tal, também pode ser usada para a modelação da dinâmica dos SLAs. A necessidade de criar um metamodelo prende-se com o facto de este ser necessário para a prossecução de duas finalidades: (i) a instanciação da dinâmica dos processos de ITSM para verificação da conformidade de SLAs e (ii) a avaliação da complexidade de processos de ITSM.

A primeira finalidade é inicialmente explorada nesta dissertação e será continuada no âmbito de um trabalho de doutoramento de Anacleto Correia, em curso no grupo QUASAR [Correia and Abreu, 2010]. Pretende-se verificar a conformidade do serviço com SLA através da implementação de uma monitorização dos processos. Será efectuada a verificação da conformidade entre os níveis de serviço esperados (especificados num SLA) e os serviços efectivamente prestados. A segunda finalidade foi já atingida no âmbito do trabalho de mestrado de Raquel Porciúncula [Porciúncula, 2010], que se ancorou no metamodelo que seguidamente será apresentado. Nesse sentido, o trabalho aqui desenvolvido é mais um meio do que um fim em si próprio.

Escolhemos a linguagem BPMN, devido à sua riqueza semântica e precisão formal, que a torna apropriada como técnica de modelação dinâmica de serviços. Por outro lado, como a linguagem BPMN foi definida como uma extensão (*profile*) à linguagem de modelação UML, que usámos na ontologia de ITSM, somos capazes de retratar processos de SLA com rigor em todo o seu ciclo de vida, e mostrar como eles interagem com entidades da ontologia de ITSM.

Dada a extensão do metamodelo, o mesmo foi organizado em vários pacotes para facilitar a percepção, seguindo critérios de agregação semântica, ou seja, tentando agrupar conceitos (meta-classes) mais inter-relacionados. Por cada pacote produziu-se um ou mais diagramas de meta-classes. Os pacotes criados foram os seguintes:

1. Modelo (**Model**) – Contém uma visão de topo onde se define que projecto estamos a modelar e que versões existem;
2. Elementos Estruturais (**Structural Elements**) – Demonstra como os elementos de BPMN, que constituem um processo, se relacionam entre si;
3. Elementos de Suporte (**Supporting Elements**) – Modela conceitos que não são utilizados directamente no desenho do processo, mas que são necessários à sua implementação. São exemplos os recursos utilizados pela tarefa ou as portas das *gateways*. Nenhum deles tem tradução gráfica em BPMN;
4. Conectores (**Connectors**) – São os elementos que interligam os restantes elementos de BPMN; são compostos por associações, fluxos de mensagens e fluxos de sequência;
5. Objectos de Fluxo (**Flow Objects**) – É o maior grupo de elementos de BPMN, constituído por actividades, *gateways*, eventos;
6. Artefactos (**Artifacts**) – Contém um conjunto de elementos de BPMN com características diferentes, que são os grupos, objectos de dados (*data objects*) e anotações (*text annotation*);
7. Orquestração (**Orchestration**) – Desenha a forma como os elementos de BPMN comunicam entre diferentes *pools*;
8. Simulação (**Simulation**) – Permite a definição de recursos e a sua atribuição a tarefas;
9. Conexão Interna (**Internal Connection**) – Contém o desenho de como os diferentes elementos de BPMN se conectam entre si;
10. Casos de Utilização (**PackageUseCase**) – Relaciona os casos de utilização com o desenho de um processo BPMN;

Para apresentar cada um dos pacotes iremos usar as seguintes secções:

1. **Propósito**
2. **Diagrama** (de meta-classes)
3. **Meta-Classes** (descrição das mesmas)
4. **Instanciação** (diagrama de objectos correspondente ao nível M1)

Com o objectivo de instanciar o metamodelo e assim efectuar uma sua pré-validação, criámos o exemplo representado na **Figura 28**.

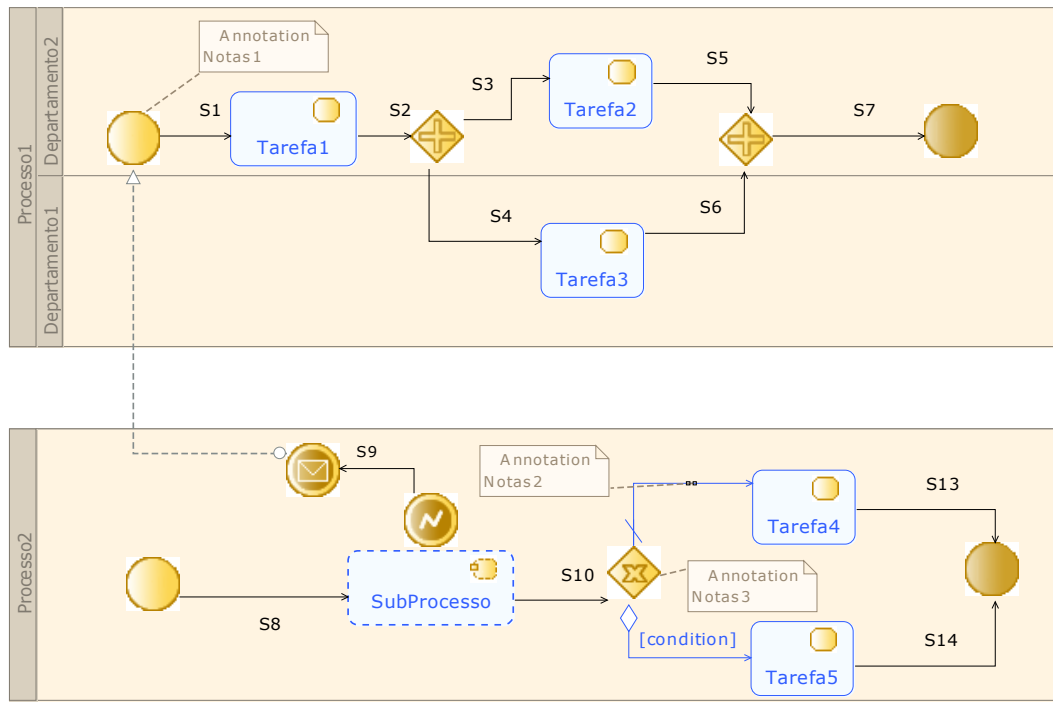


Figura 28 - Exemplo genérico de processo BPMN

4.2 Descrição dos Pacotes

Tal como já referido, o metamodelo proposto é composto por vários pacotes, como representado na **Figura 29**, onde são mostradas as dependências.

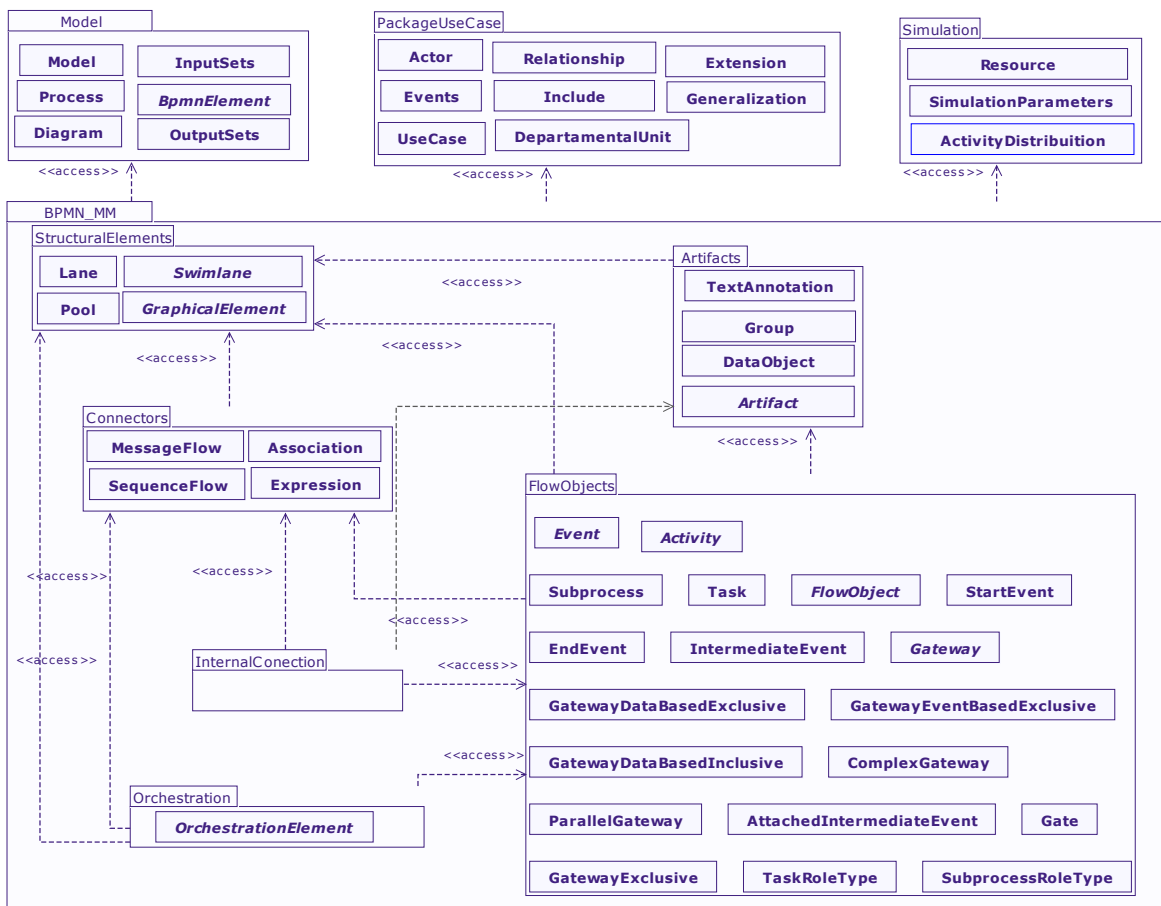


Figura 29 - Pacotes do metamodelo de BPMN

4.2.1 Pacote Modelo

Propósito:

Constitui uma visão dos elementos auxiliares que descrevem o que se está a modelar, qual o nome do processo, quem é o responsável pelo processo e quais os diagramas envolvidos. Na **Figura 30** apresentamos o diagrama de classes do pacote *Modelo*.

Diagrama:

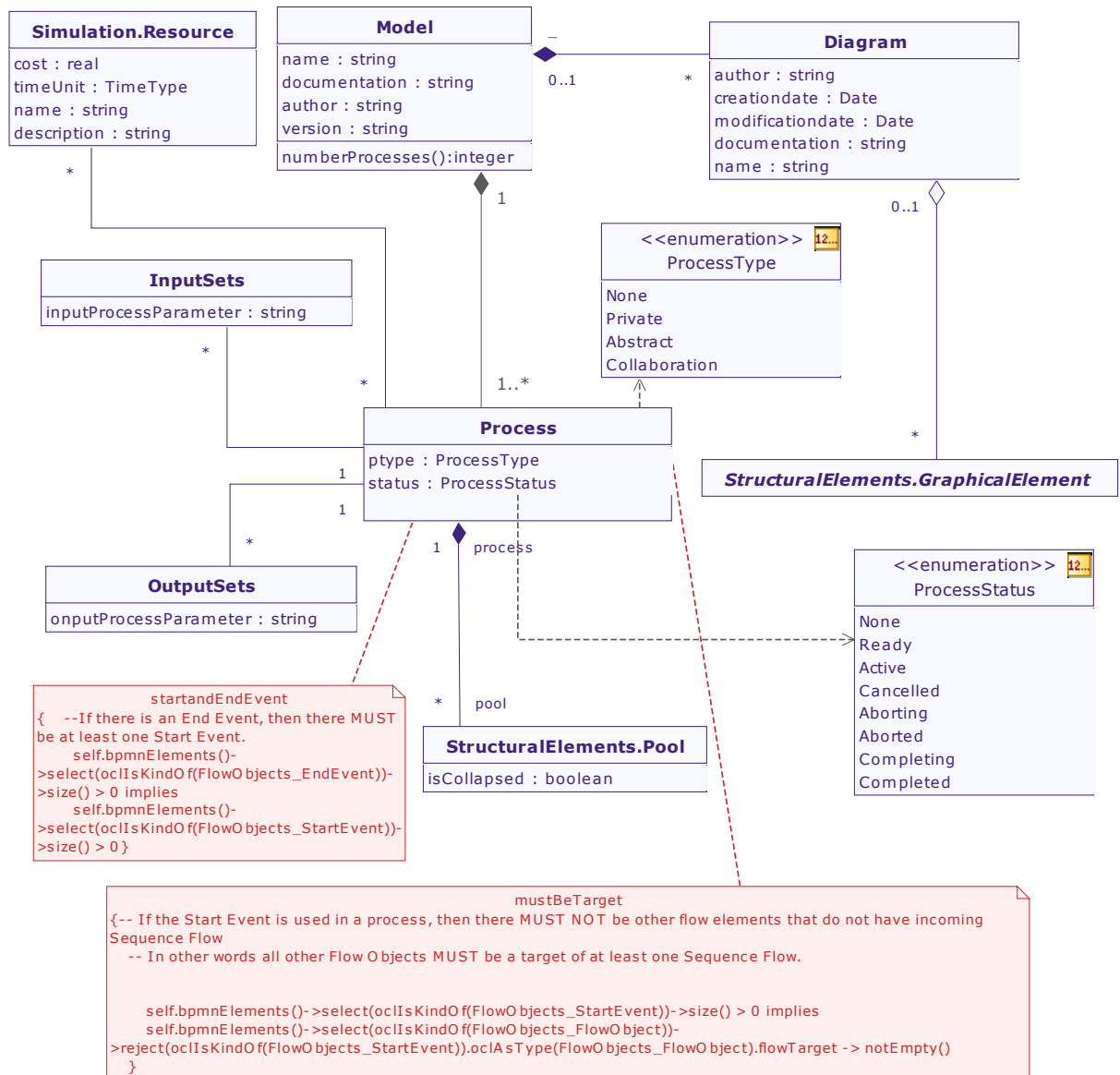


Figura 30 - Diagrama de classes do pacote *Modelo*

Meta-Classes:

A. Diagrama (*Diagram*) - Corresponde aos vários desenhos físicos utilizados para implementar o processo;

- B. Modelo (*Model*) - O modelo corresponde à informação geral do que pretendemos modelar, entendido como um processo ou conjunto de processos;
- C. Processo (*Process*) - Corresponde ao processo que estamos a modelar, que se pode desenvolver numa ou várias *pools*;
- D. Parâmetros de entrada (*Input Set*) - Permite definir os elementos necessários para o processo começar. Podem ser definidos zero ou mais parâmetros de entrada;
- E. Parâmetro de saída (*Output Set*) - O processo ao terminar pode criar parâmetros de output.

Nota: Estes parâmetros de entrada e saída existem nas especificações OMG 1.2 e 2.0. A sua existência está relacionada com o mapeamento com a linguagem de execução de processos *BPEL* (*Business Process Execution Language*).

Os tipos enumerados utilizados são os seguintes:

Tipo de Processo (*ProcessType*) – Providencia informação sobre o nível em que o processo irá ser descrito e pode assumir os seguintes valores:

- Abstracto (*Abstract*) - São interações de um processo privado com processos externos (fora da organização); ocorrem dentro da organização, porém interagem com processos de outra organização, normalmente envolve funcionários, fornecedores ou clientes; muitas vezes os processos incluem actividades que são realizadas fora da empresa, por exemplo em fornecedores; utilizamos um modelo abstracto para representar uma entidade independente com processos que não podemos representar (abstractos) porque não conhecemos ou não nos interessa modelar;
- Privado (*Private*) – São inerentes à organização em si; ocorrem dentro da organização e *normalmente* envolvem os seus colaboradores; o fluxo de processo ocorre dentro da *pool* e não ultrapassa os seus limites;
- Colaboração (Collaboration) – Um processo de colaboração mostra as interações entre duas ou mais entidades de negócio e pode ser visto como uma colaboração ou comunicação entre dois processos abstractos; essas interações são definidas como uma sequência de actividades que representam os padrões de troca de mensagens (pergunta e resposta) entre as entidades envolvidas;
- Nenhum (*None*) – Indefinido (*Default*).

Estado do Processo (*ProcessStatus*) - O estado de um processo é determinado quando está a ser executado por um motor de processos; tem a ver com competição e com cooperação

(competição na utilização de recursos escassos e cooperação, por exemplo, quando um processo tem de esperar por outro para se completar. Pode assumir os seguintes valores:

- 1.1. *Pronto (Ready)* – Processo pronto para ser executado;
- 1.2. *Activo (Active)* – Processo a ser executado;
- 1.3. *Cancelado (Cancelled)* – Processo cancelado por exemplo porque o processo expirou ou foi cancelado por um utilizador;
- 1.4. *Abortado (Aborted)* - Quando o processo interrompe inesperadamente, os processos restantes não recebem nenhuma razão para o encerramento do processo (evento inexplicável);
- 1.5. *Aborting* – Processo a ser abortado;
- 1.6. *Completing* - Processo a ser completado;
- 1.7. *Completo (Completed)* – Processo que terminou normalmente.

Os recursos (modelados em *Resource*) são aqueles que são utilizados no processo (homens, materiais e máquinas) e *input* e *output* correspondem a parâmetros de entrada e saída do processo. Um possível parâmetro de *input* seria o número de vezes que um processo vai ser executado.

Instanciação:

Doravante iremos representar nesta subsecção o fragmento da instanciação (nível M1) da parte do metamodelo, neste caso o diagrama de classes do pacote **Modelo** que se pode consultar na **Figura 30**, a que esta secção se refere para o exemplo proposto na **Figura 28** o exemplo genérico de processo BPMN. Como nem sempre o que é apresentado no metamodelo tem representação em diagramas BPMN, temos de complementar este exercício de instanciação com uma descrição textual, como faremos seguidamente.

Assim, no caso da **Figura 28** temos um modelo que designamos “Exemplo”, que é representado por um diagrama de processos que é constituído por dois processos (Processo1 e Processo2), tal como representado com o modelo de objectos na **Figura 31**.

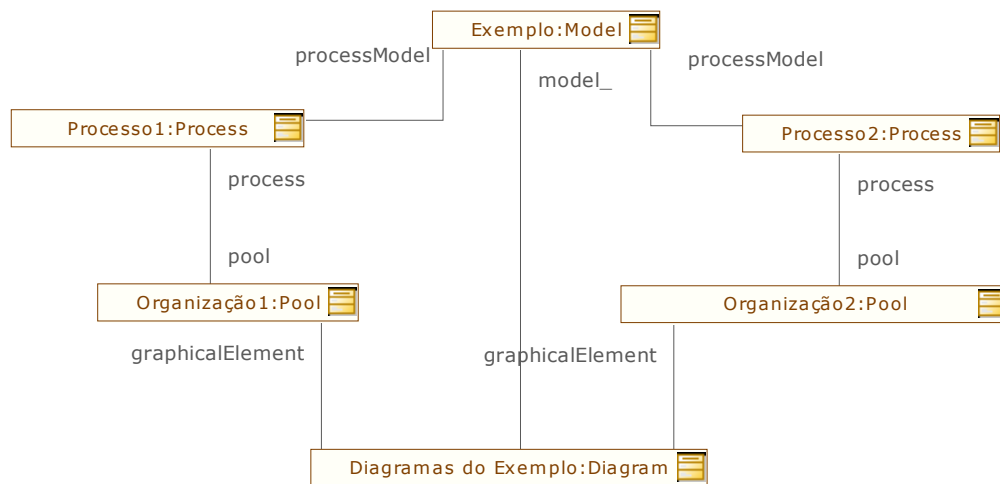


Figura 31 - Diagrama de objectos do modelo “Exemplo”

4.2.2 Pacote Elementos Estruturais

Propósito:

Este pacote propõe a forma como os elementos que constituem o BPMN se relacionam e estruturam, o seu diagrama de classes pode ser visto na **Figura 32**. O elemento central é a *swimlane*, que pode ser uma *lane* ou *pool*. Uma *pool* representa um processo executado por um actor humano, computador ou organização. Actua como um contentor gráfico para dividir um conjunto de actividades de outras *pools*, geralmente no contexto de situações de *B2B (Business to Business)*. Uma *lane* é uma subdivisão de *pool* e pode, por exemplo, representar conceitos como um departamento de uma empresa.

Diagrama:

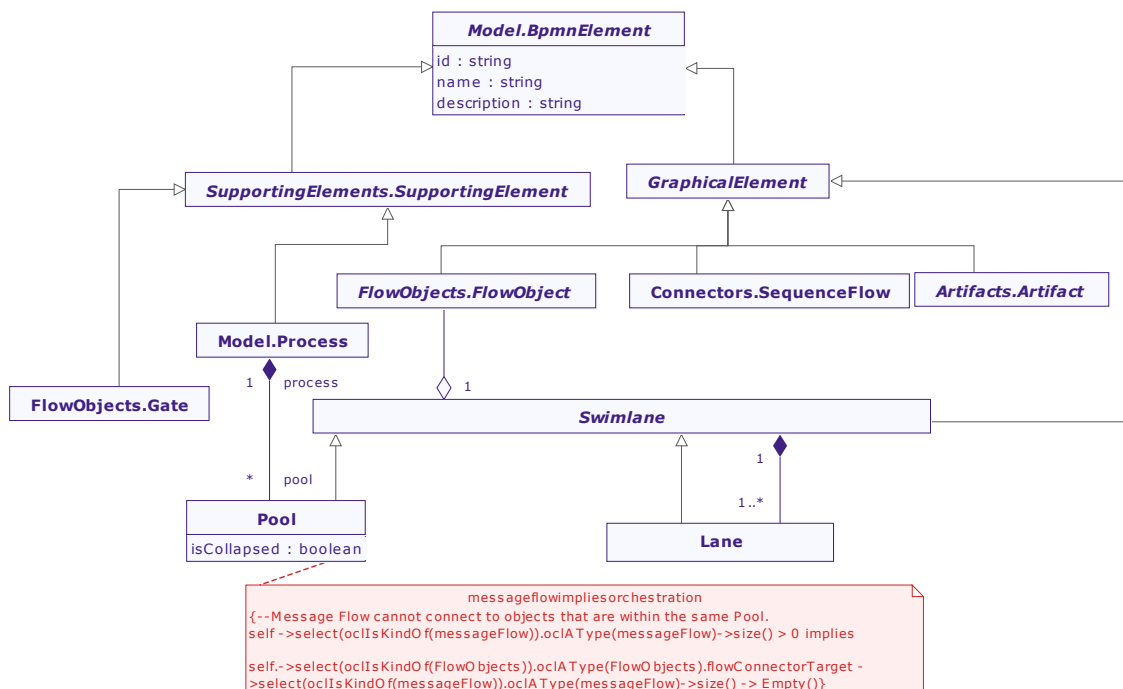


Figura 32 - Pacote de elementos estruturais BPMN

Meta-Classes:

- A. *Swimlane* – Classe abstracta que é concretizada em *lanes* e *pools*. Uma *swimlane* (*pool* ou *lane*) pode conter várias *lanes*, o que permite definir estruturas hierárquicas com tantos níveis, quantos os requeridos. É uma das quatro categorias básicas de elementos (Objectos de Fluxo, Objectos de Conexão, *Swimlane*, Artefactos);
- B. Piscina (*Pool*) - Representa o rectângulo onde se desenha o processo; pode-se subdividir em pistas;
- C. Pista (*Lane*) - Podem existir ou não nas *pools*; podem-se criar *lanes* dentro de *lanes*;
- D. Elemento Gráfico (*Graphical Element*) – Classe abstracta que contém todos os elementos representados graficamente no diagrama.

O significado e representação dos objectos em BPMN são descritos na **Tabela 18 - Pools e Lanes (BPMN)**:



Objecto	Descrição	Figura
Pool	Uma <i>pool</i> representa um papel num processo. Ela actua como um <i>container</i> gráfico para dividir um conjunto de actividades de outras <i>pools</i> .	
Lane	Uma <i>lane</i> é uma subdivisão dentro de uma <i>pool</i> (ou de outra <i>lane</i>) e é usada para organizar e classificar as actividades.	

Tabela 18 - Pools e Lanes (BPMN) extraído de [OMG, 2009b]

Dentro das *pools* ou *lanes* existem objectos do fluxo, conectores, artefactos, decisões, etc., ou seja, todos os elementos que constituem o desenho do processo.

Instanciação:

Apresentamos de seguida na **Figura 33** um extracto retirado do exemplo genérico de processo BPMN da **Figura 28**, que corresponde a duas *pools* ligadas a dois processos:

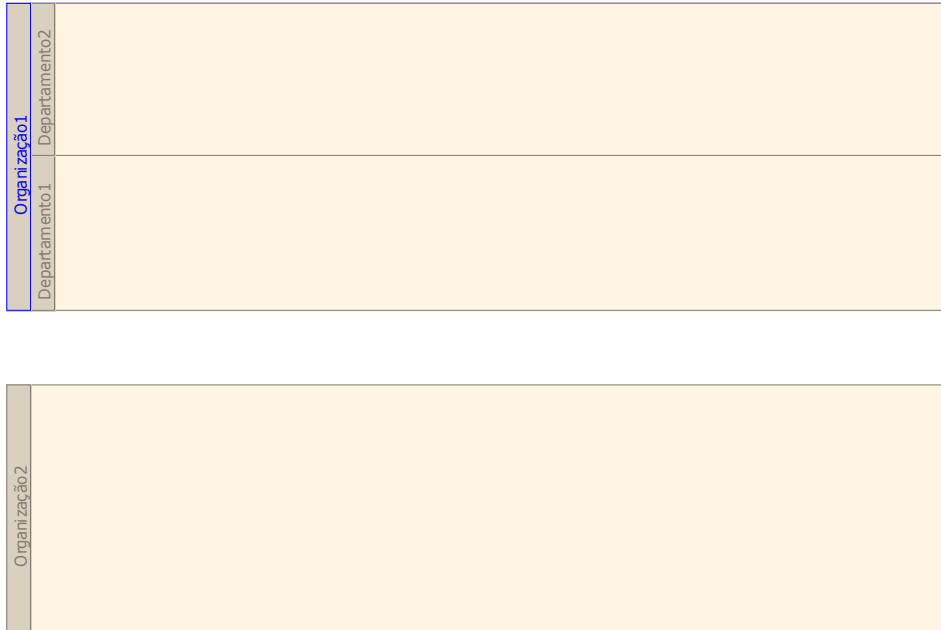


Figura 33 - Exemplo de Lanes e Pools

Apresentamos o correspondente modelo de objectos na **Figura 34**.

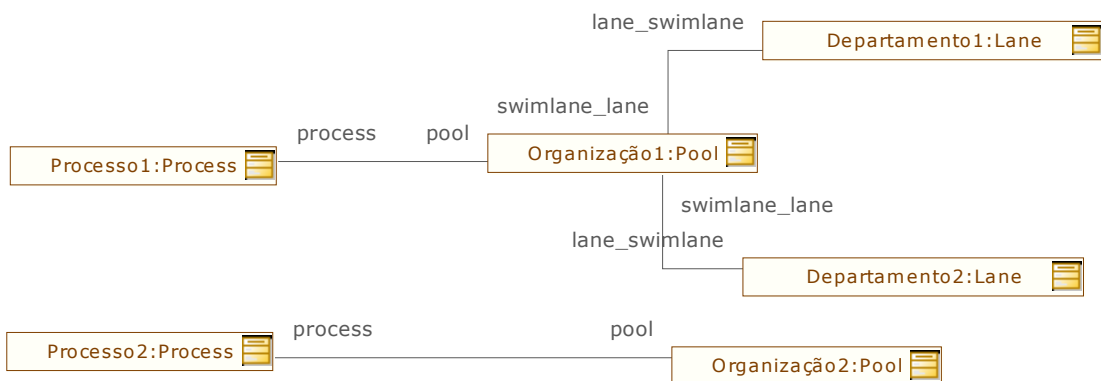


Figura 34 - Diagrama de objectos com exemplo de Lanes e Pools

4.2.3 Pacote Elementos de Suporte

Propósito:

Este pacote agrupa elementos que não têm representação gráfica nos diagramas BPMN, o seu diagrama de classes é mostrado na **Figura 35**.

Diagrama:

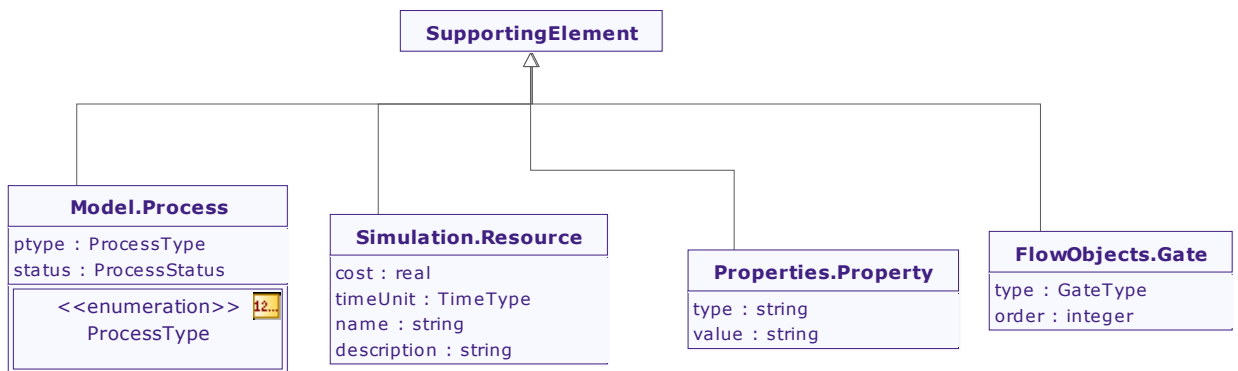


Figura 35 - Elementos de suporte BPMN

Meta-Classes:

Este diagrama não tem classes próprias. Foi criado para que as classes aqui designadas possam herdar características de elementos de suporte, com base em estrutura semelhante existente no documento de especificações BPMN da OMG.

4.2.4 Pacote Conectores

Propósito:

São os elementos que interligam os objectos no fluxo, em termos de elementos BPMN são Associações, Mensagens e Sequências, são elementos descritos na Tabela 19 - Conectores (BPMN).

Diagrama:

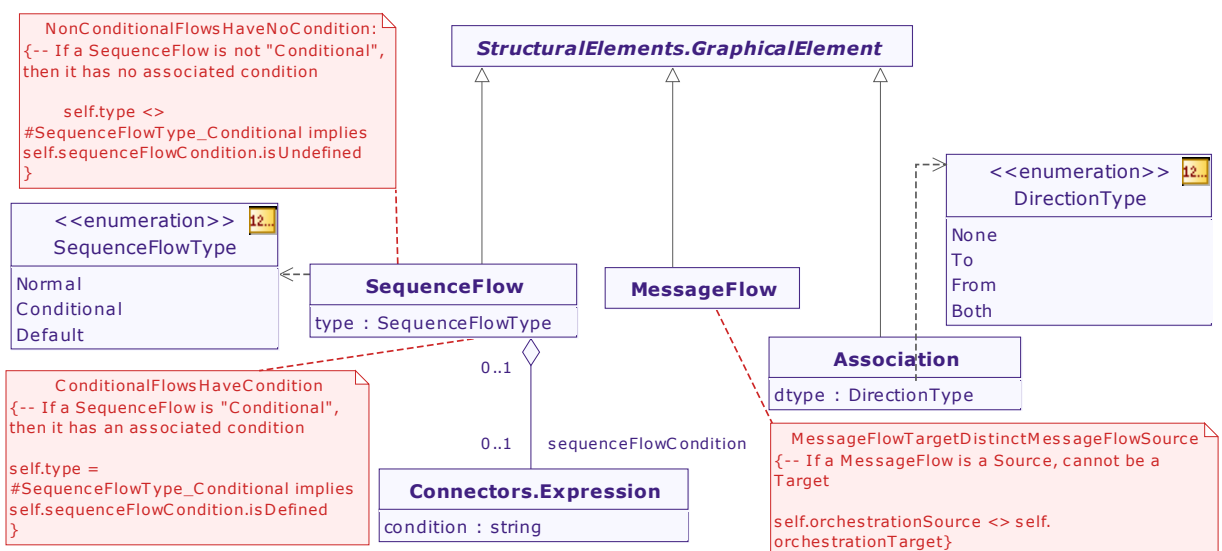


Figura 36 - Pacote de Conectores BPMN

Meta-Classes:

- A. Associação (*Association*) - Para além do seu significado, (ver **Tabela 19**) como elemento de BPMN, este conector é extremamente flexível, dado que se pode ligar a praticamente tudo. Esta flexibilidade implica propriedades diferentes de outros conectores. Apesar de ter representado este objecto com um sentido (seta) poderá não ter ou ter ambos os sentidos (*Direction Type*);
- B. Fluxo de mensagem (*Message Flow*) - É uma mensagem, com emissor, receptor e conteúdo;
- C. Fluxo de sequência (*Sequence Flow*) - É utilizado para ligar essencialmente, eventos, actividades e decisões, podem ter condições. As condições são utilizadas para condicionar o sentido do fluxo e podem ser expressas em *gateways* ou nos próprios fluxos de sequência;
- D. Fluxo de sequência condicional (*Sequence Flow Expression*) - Classe que vai conter a condição do fluxo de sequência condicional.

Descrevemos de seguida os objectos na **Tabela 19**;


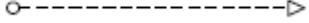

Objecto	Descrição	Figura
Fluxo de sequência	É usado para mostrar a ordem (sequência) com que as actividades, decisões ou eventos são executados num processo.	
Fluxo de mensagem	É usado para mostrar o fluxo das mensagens entre dois participantes diferentes (pistas ou piscinas) que os emitem e recebem.	
Associação	É usada para associar dados, texto, e outros artefactos com os objectos de fluxo. As associações são usadas para mostrar as entradas e as saídas das actividades.	

Tabela 19 - Conectores (BPMN) extraído de [OMG, 2009b]

Os tipos de enumerados utilizados são os seguintes:

Tipo de Direcção (*DirectionType*) – Caracteriza o sentido (seta) do fluxo; pode assumir os seguintes valores:

- 1.8. *None* – Nenhum sentido;
- 1.9. *To* – O sentido expresso é o *target* (ver pacote *Internal connection*);
- 1.10. *From* – O sentido expresso é o do *source* (ver pacote *Internal connection*);
- 1.11. *Both* – Ambos os sentidos (duas setas).

Tipo de Fluxo de Sequência (*SequenceFlowType*) – O *flow* pode ter uma condição ou não e os estados que podem assumir são:

- 1.12. *Normal* – Sem condição;
- 1.13. *Conditional* – Com uma Condição;
- 1.14. *Default* – Não é definido, fluxo por defeito.

Instanciação:

As classes *Association*, *Message Flow*, *Sequence Flow*, *Sequence Flow Condition* são instanciadas, sem nenhuma ordem específica entre si, dado que os conectores servem para ligar os outros elementos de BPMN, não fazendo sentido apresentar um diagrama de objectos só com conectores.

Assim, para ilustrar criamos o Exemplo no “Processo 1” com conectores BPMN que apresentamos na **Figura 37** que é um extracto do modelo que serve de exemplo genérico de processo BPMN apresentado anteriormente na **Figura 28**. Depois o modelo de objectos que corresponde ao nível M1 do processo 1 e respectiva organização 1, é apresentado na **Figura 38**.

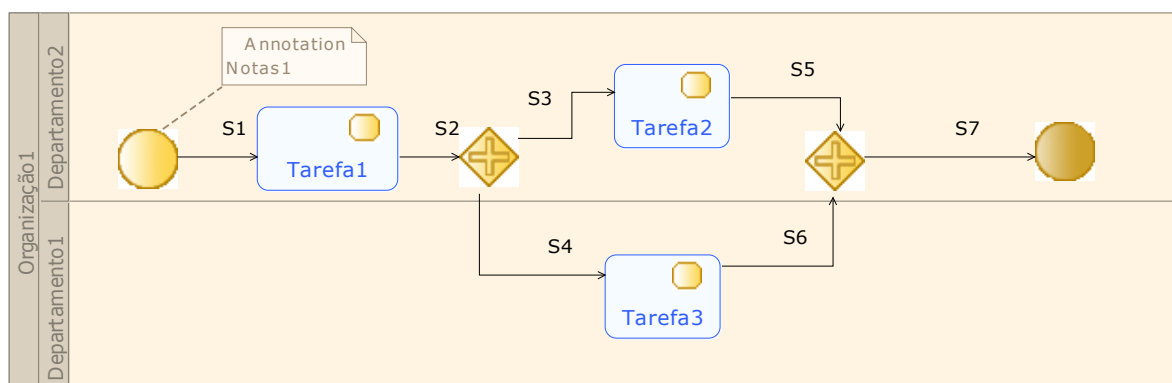


Figura 37 - Exemplo com “Processo 1” com conectores BPMN (Nível M1)

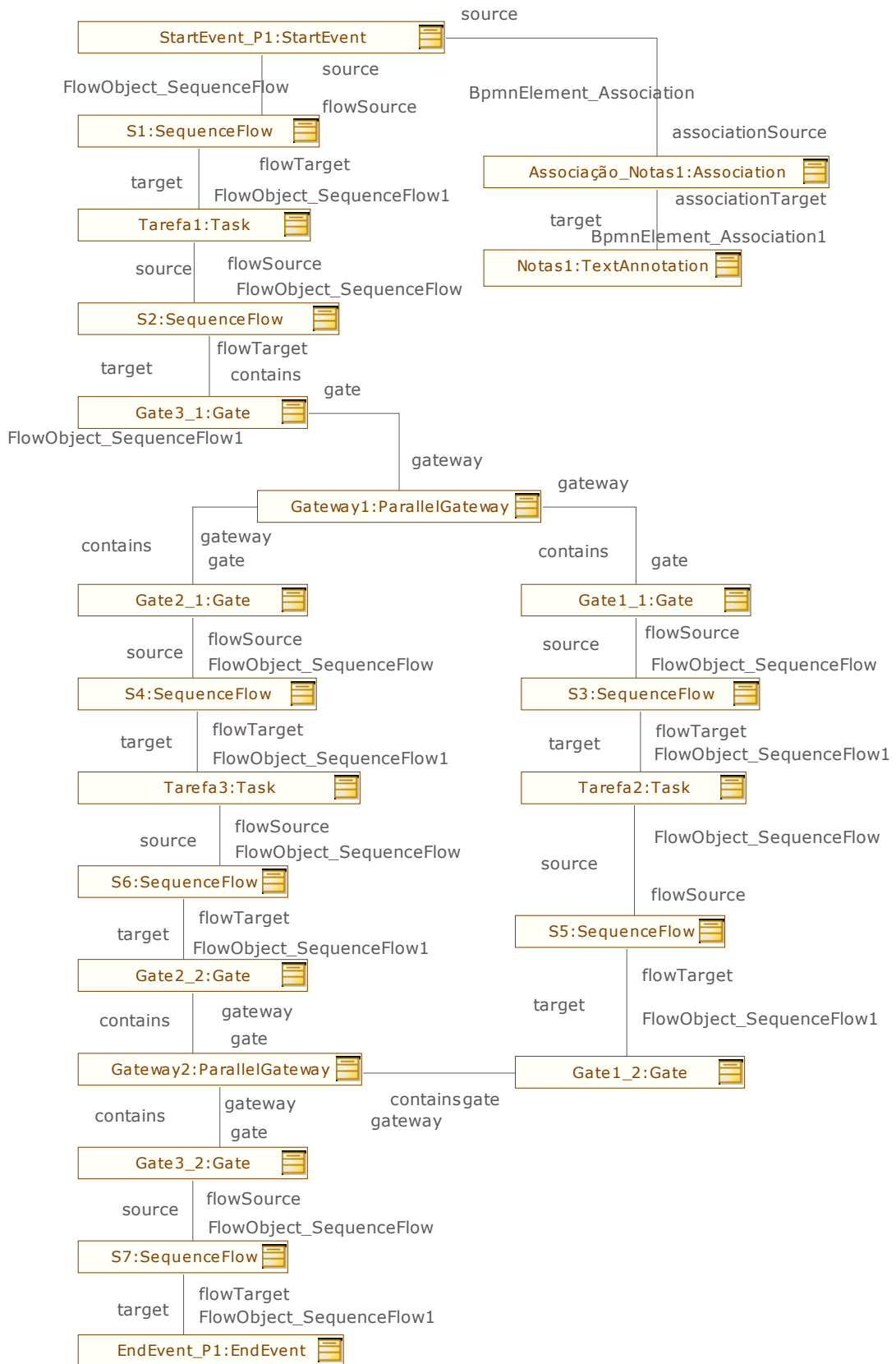


Figura 38 - Diagrama de objectos do "Processo 1"

4.2.5 Pacote Objectos de Fluxo

Propósito:

Os objectos de fluxo são os principais elementos gráficos que definem o comportamento dos processos de negócio. Estes são divididos entre três tipos de objectos: eventos, actividades e decisões (*gateways*), como são descritos na **Tabela 20**.




Objecto	Descrição	Figura
Evento	É algo que acontece durante um processo de negócio. Estes eventos afectam o fluxo do processo e têm geralmente uma causa (são despoletados, por um <i>trigger</i>) ou um impacto (<i>result</i>). Há três tipos de eventos: <i>Start</i> , <i>Intermediate</i> e <i>End</i> .	
Actividade	É um termo genérico para um trabalho executado. Os tipos de actividades são: tarefas e sub-processos.	
Gateway	É usado para controlar a divergência e a convergência da sequência de um fluxo. Assim, determinará decisões como juntar ou dividir caminhos. Ele irá determinar ramificação, bifurcações, fusão ou junção dos caminhos. Marcadores internos irão indicar o tipo de comportamento.	

Tabela 20 - Objectos de Fluxo extraído de [OMG, 2009b]

Diagrama:

De seguida, na **Figura 39**, apresentamos o diagrama de classes de topo com estes três grupos de objectos. Posteriormente entraremos em detalhes com relação à subdivisão destes grupos de elementos.

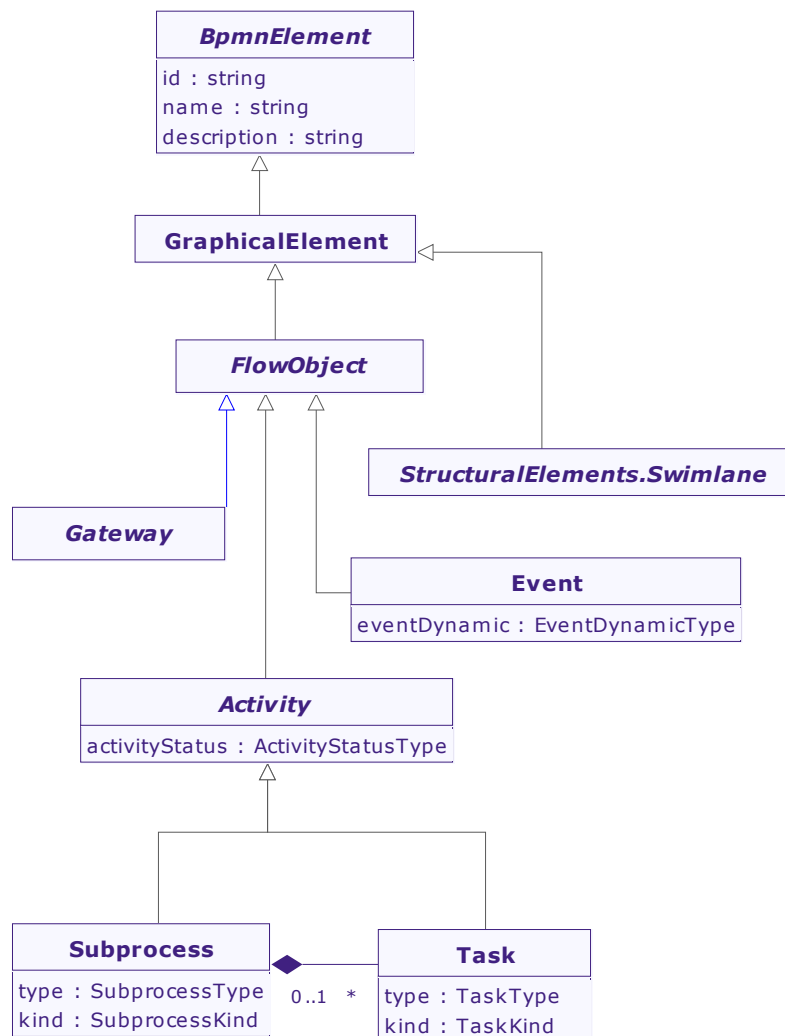


Figura 39 - Pacote de *Flow Objects* de BPMN

Meta-Classes:

Neste diagrama todas as classes são abstractas, excepto *Subprocess* e *Task*. De seguida, apresentamos diagramas de classe detalhados para todos os grupos de elementos, incluindo *Task* e *Subprocess* nas secções seguintes.

4.2.6 Pacote Actividade

Propósito:

Actividade é um termo genérico para um trabalho desempenhado por uma organização. Uma actividade pode ser atómica ou não atómica (composta). Os tipos de actividades que são parte de um modelo de processo são: Processo (*Process*), Sub-Processo (*Sub-Process*) e

Tarefa (*Task*). *Task* e *Sub-Process* são representados por rectângulos ligeiramente arredondados nos cantos, como representado na **Tabela 20**.

De seguida apresentamos o diagrama de classes na **Figura 40** para este pacote;

Diagrama:

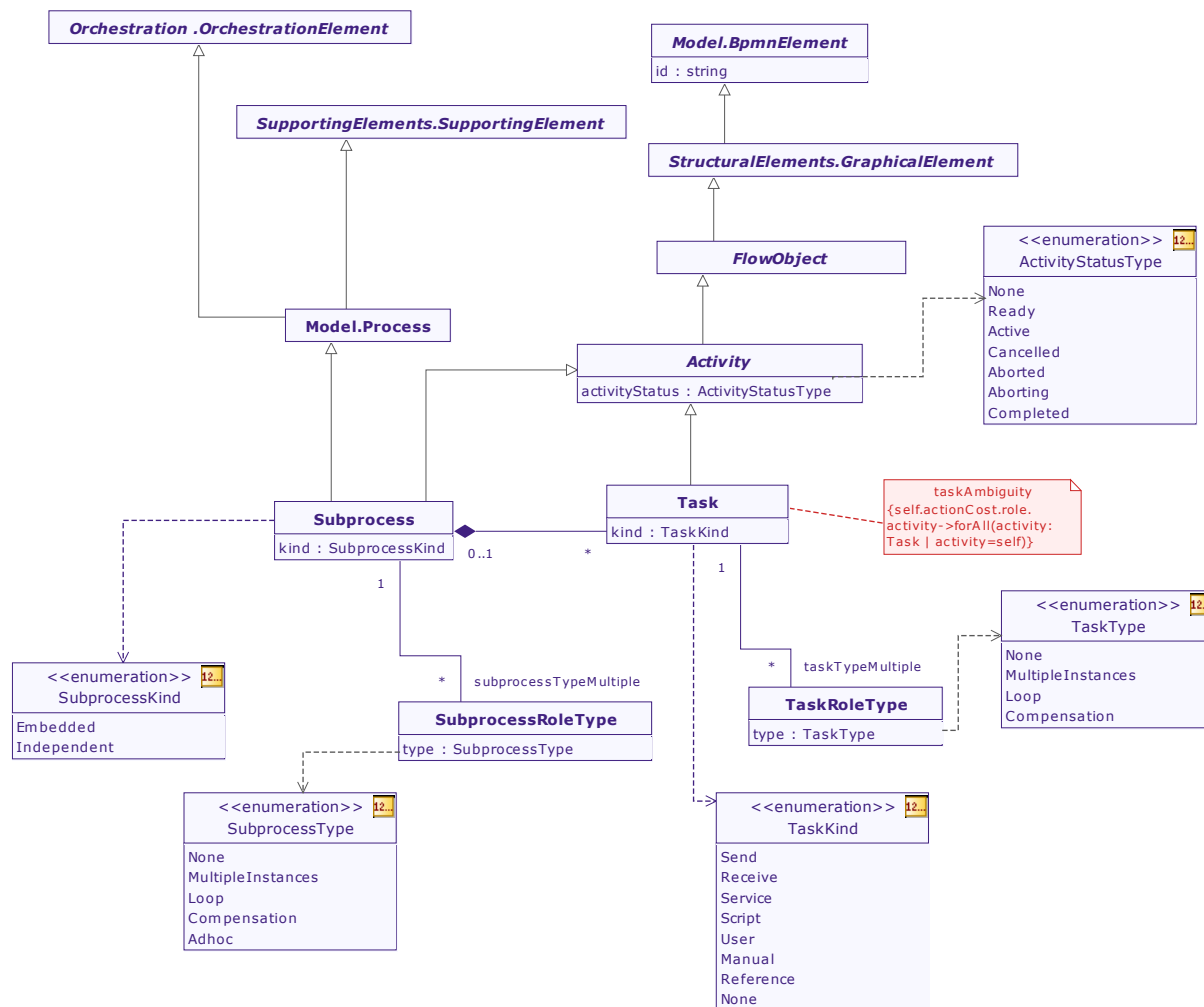


Figura 40 - Pacote Actividade

Meta-Classes:

- A. Actividade (*Activity*) - Uma actividade representa um trabalho que é realizado num processo de negócio, no entanto, pode ser decomposta ou não, ou seja, pode ser uma tarefa bem identificada ou um conjunto de tarefas, um sub-processo;
- B. Tarefa (*Task*) - Representa a menor unidade de trabalho; não pode ser dividida noutras actividades;
- C. *Sub-processo (Subprocess)* - Pode ser subdividido noutras actividades, constituindo processos.

Os tipos enumerados utilizados são os seguintes:

Estado da Actividade (*ActivityStatusType*) – O estado de uma actividade é determinado quando está a ser executado por um motor de processos. Tem a ver com competição e com cooperação, competição na utilização de recursos escassos, e cooperação, por exemplo, quando uma actividade tem de esperar por outra para se completar. Os valores que podem assumir são os mesmos de Estado do Processo (*Process Status*) anteriormente apresentados, pelo que não iremos aqui repetir a sua descrição.

Tipo de Tarefa (*TaskType*) – Uma actividade pode ter um ou dois destes marcadores; os valores que podem assumir são os seguintes:

1.15. Indefinido (*None*) - *Default*;

1.16. Múltiplas Instâncias (*Multiple Instances*) – A tarefa pode ser lançada várias vezes sem que esteja terminada e existirem múltiplas *threads* de execução;

1.17. Ciclo (*Loop*) – Uma tarefa pode ter um *loop* e é executada repetidamente enquanto a expressão avaliada for verdadeira (*while*) ou até que a expressão avaliada for verdadeira (*repeat until*);

1.18. Compensação (*Compensation*) - Tarefa executada para repor um estado anterior coerente (*rollback*).

Instanciação:

Apresentamos na **Figura 45** um exemplo que engloba actividades, *gateways*, eventos e artefactos, este é um extracto do exemplo apresentado anteriormente na **Figura 28** (Exemplo genérico de processo BPMN). Em seguida, na **Figura 46** mostramos o correspondente modelo de objectos (nível M1).

4.2.7 Pacote Gateway

Propósito:

Um *gateway* é usado para controlar a divergência ou convergência de fluxos de sequência. Dessa forma, ele irá determinar ramificação, bifurcação, ligação e junção de caminhos. Os símbolos internos à notação gráfica irão indicar o tipo de controlo. Pode consultar os tipos de gateways na **Figura 41**.

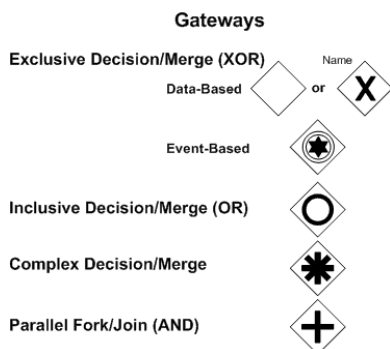


Figura 41 - Tipos de Gateways, extraído de [OMG, 2009a]

Apresentamos na **Figura 42** o diagrama de classes para o Pacote Gateway;

Diagrama:

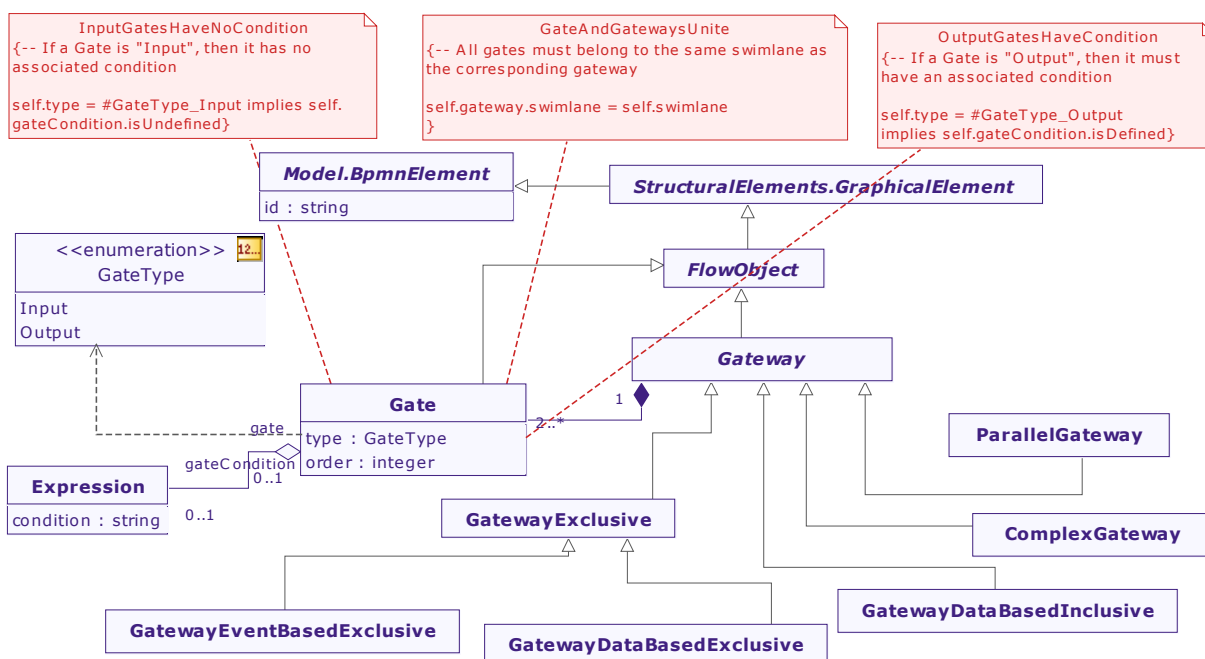


Figura 42 - Pacote Gateways de BPMN

Meta-Classes:

- A. *Gateway* - Classe abstracta, generalização de todos os *gateways*;
- B. *Gate* - Portas dos gateways, trata-se de um objecto não representado graficamente;
- C. *ConditionGate* - Permite expressar condições nos fluxos de entrada e saída;
- D. *GatewayExclusive* (XOR) - O fluxo de saída pode, somente, optar um dos caminhos. Pode ser percorrido para uma dada instanciação do processo; inclui dois casos:
 - a. *GatewayEventBasedExclusive* - O fluxo de saída que é seleccionado é baseado num evento;
 - b. *GatewayDataBasedExclusive* - O fluxo de saída que é seleccionado baseia-se na expressão booleana existente na *gateway*;

- E. *GatewayDataBasedInclusive* (OR) - O fluxo pode tomar dois ou mais caminhos alternativos, neste caso, a avaliação de uma condição verdadeira não exclui a avaliação de expressões nas condições das outras saídas e todas saídas cuja avaliação for verdadeira, serão atravessadas por um *token*:
- F. *ComplexGateway* - É um *gateway* complexo para lidar com situações que não são facilmente manipuladas através de outros tipos de *gateways*; Permite um conjunto de fluxos de entrada e de saída. É o *gateway* cujo comportamento está menos bem definido e pode ser usado de várias formas, por exemplo, uma expressão pode avaliar os dados do processo e, em seguida, seleccionar diferentes conjuntos de saída;
- G. *ParallelGateway* (AND) - Permite paralelismo e sincronização.

Instanciação:

Apresentamos na **Figura 45** um exemplo com o “Processo 2” para objectos de fluxo e artefactos que engloba actividades, *gateways*, eventos e artefactos, que é um extracto do exemplo apresentado anteriormente na **Figura 38** designada Exemplo genérico de processo BPMN. Em seguida, na **Figura 46** apresentamos um exemplo com um modelo de objectos do “Processo 2” para objectos de fluxo e artefactos.

4.2.8 Pacote Eventos

Propósito:

Um evento é algo que acontece durante a execução do processo. Ele afecta a execução do processo e, geralmente, possui uma causa (*trigger*) e um impacto (*result*).

Eventos podem ser de três tipos, com base no momento em que afectam o processo: Início (*Start*), Intermediário (*Intermediate*) e Fim (*End*).

Os eventos do tipo início e intermediário também podem ser divididos quanto ao seu gatilho (*trigger*) em: *Message*, *Timer*, *Error*, *Cancel*, *Compensation*, *Rule*, *Link*, *Multiple*, *Terminate*.

Mostramos de seguida na **Figura 43** o diagrama de classes para o Pacote Eventos;

Diagrama:

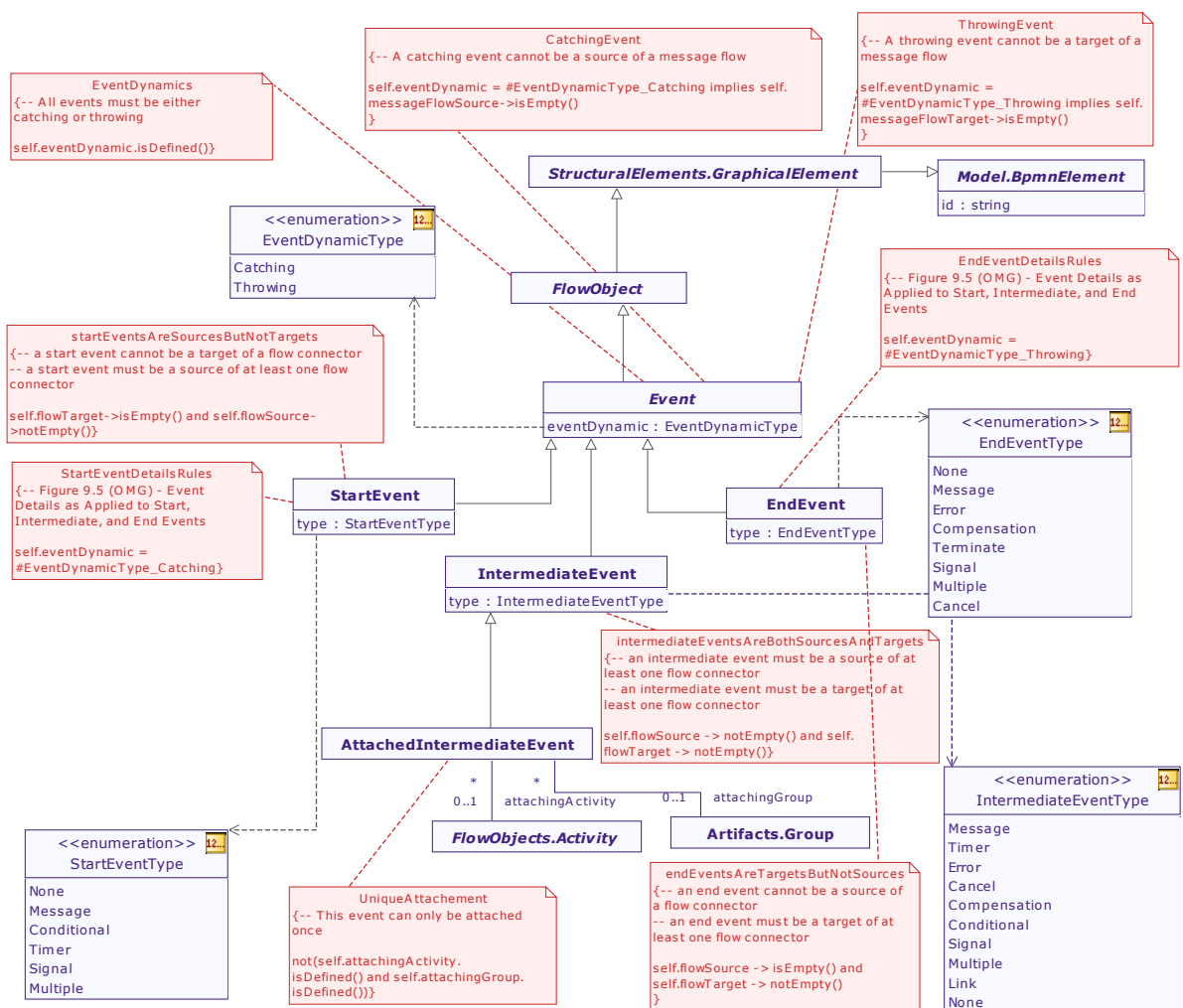


Figura 43 - Pacote de Eventos de BPMN

Instanciação:

Apresentamos na **Figura 45** um exemplo que engloba actividades, *gateways*, eventos e artefactos, que é um extracto do exemplo geral apresentado anteriormente na **Figura 28**. Em seguida, na **Figura 46** mostramos o correspondente modelo de objectos (nível M1).

4.2.9 Pacote Artefactos

Propósito:

São objectos com características especiais, como, por exemplo, poderem ser ligados a mais do que um outro objecto ou poderem existir em mais do que uma pista. São constituídos por Grupos, Anotações e Objectos de dados (*Data Objects*).

Objecto	Descrição	Figura
Objectos de dados (Data Objects)	O objecto de dado é um mecanismo para mostrar como os dados são requeridos ou produzidos por actividades. São conectados às actividades usando associações.	
Grupo (Group)	Um grupo é representado por um rectângulo e pode ser usado com finalidade de documentação ou de análise.	
Anotações (Text Annotation)	As anotações são mecanismos para fornecer informações adicionais para o leitor de um diagrama BPMN.	

Tabela 21 - Artefactos (BPMN) extraído de [OMG, 2009b]

Apresentamos de seguida na **Figura 44** o correspondente diagrama de classes;

Diagrama:

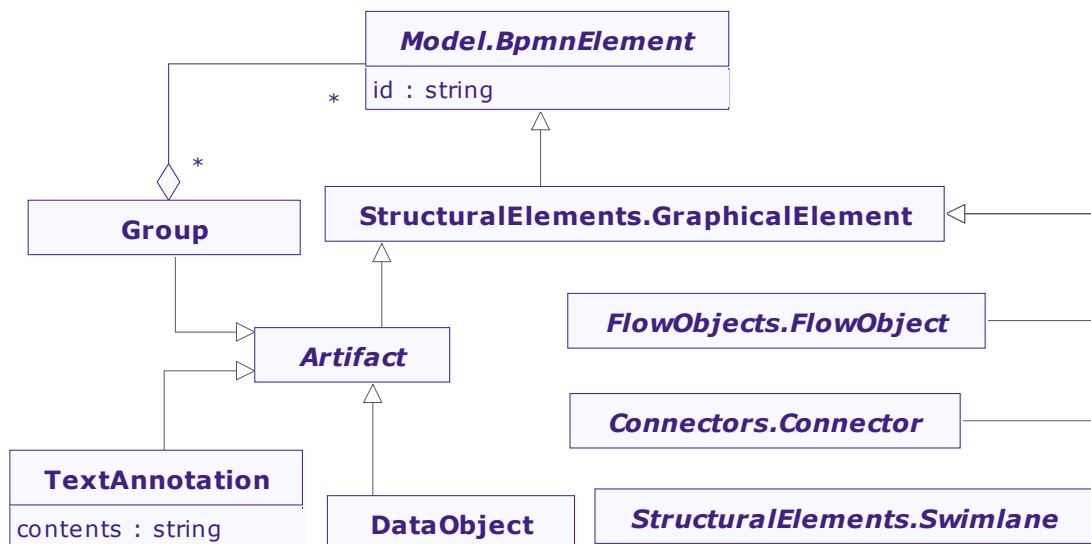


Figura 44- Pacote de artefactos BPMN

Meta-Classes:

- A. Objectos de dados (*Data Object*) - São considerados artefactos, porque não têm um efeito directo no fluxo de sequência ou no fluxo de mensagens do processo. Contudo

4. Dinâmica

forneem informação acerca do que as actividades precisam para serem executadas e o que elas produzem;

- B. Grupo (*Group*) - É um grupo de actividades que não afecta o fluxo de sequência. O agrupamento pode ser realizado para fins de análise ou documentação. Grupos também podem ser usados para identificar actividades de uma transacção distribuída que é mostrada entre *pools*;
- C. Anotações (*Text Annotation*) - Anotações de texto são mecanismos utilizados pelos analistas para prover informação adicional aos leitores do diagrama de processo de negócio.

Instanciação:

Apresentamos, na **Figura 45**, um extracto do exemplo genérico apresentado anteriormente na **Figura 28**. Segue-se a apresentação do modelo de objectos que corresponde ao nível M1 (**Figura 46**), neste caso um exemplo para englobar *Flow Objects* e *Artifacts*:

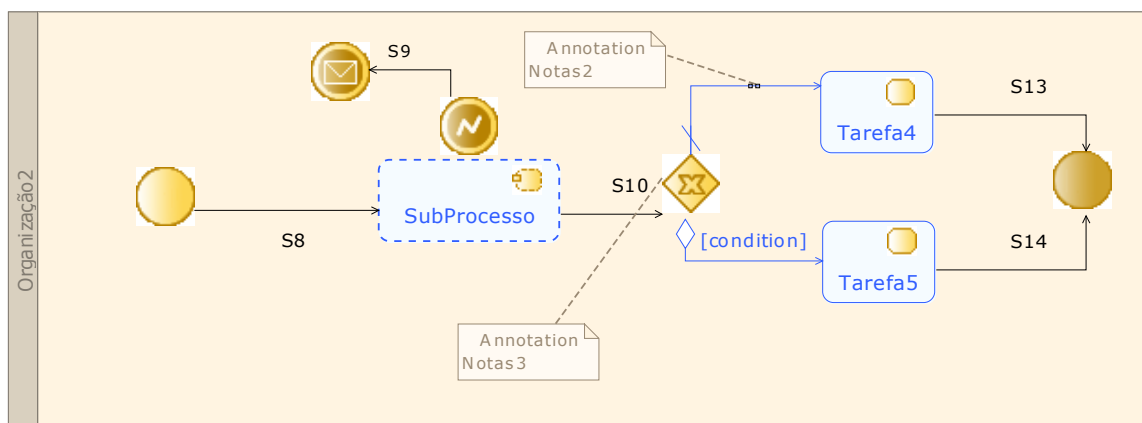


Figura 45 - Exemplo de “Processo 2” para objectos de fluxo e artefactos

4. Dinâmica

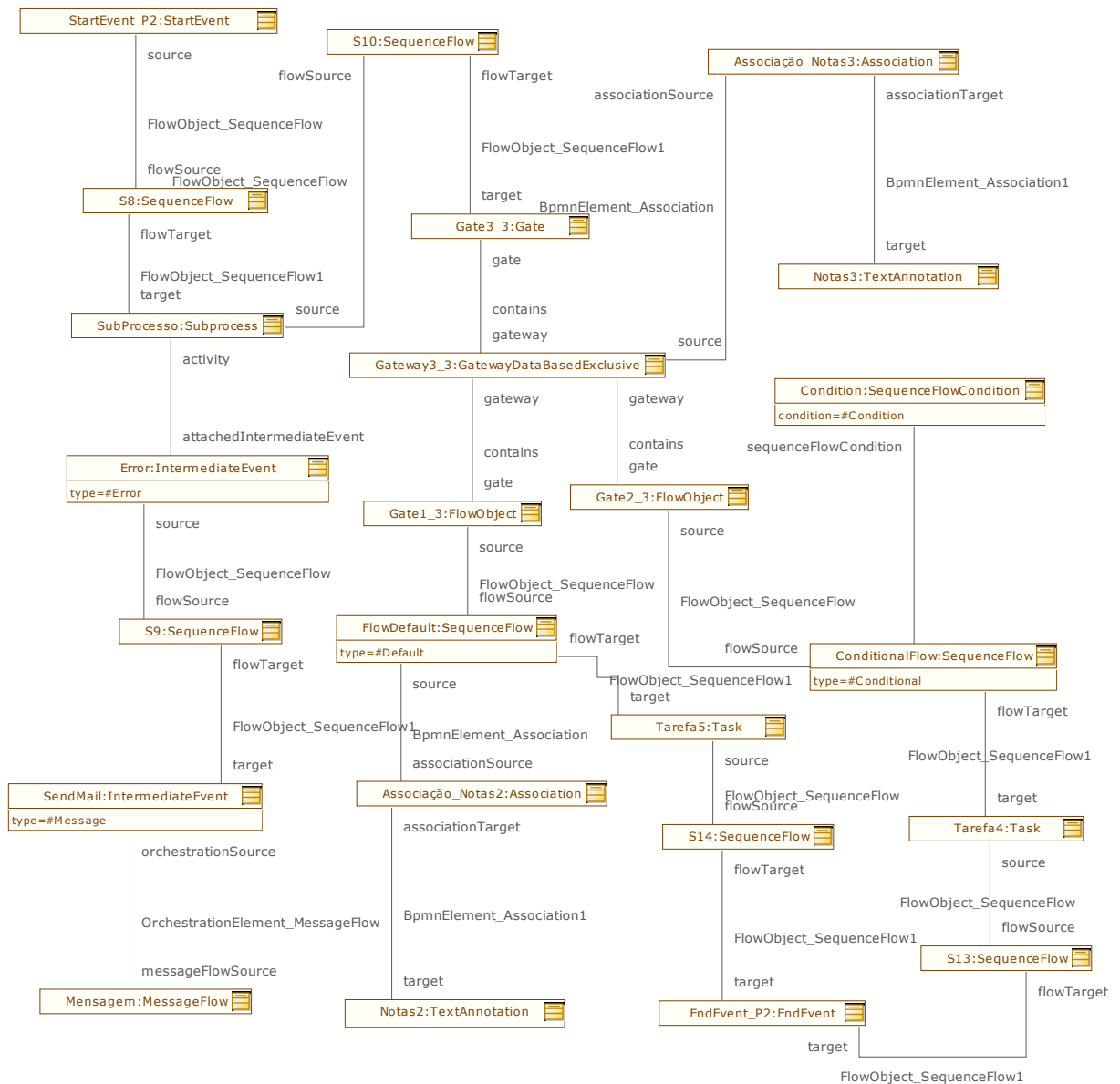


Figura 46 - Exemplo de modelo de objectos do “Processo 2” para objectos de fluxo e artefactos

4.2.10 Pacote Orquestração

Propósito:

A orquestração não se destina a representar elementos de BPMN mas sim a permitir que duas *pools* distintas comuniquem entre si ou a permitir a comunicação entre duas entidades distintas com algumas regras. A comunicação é realizada por *Message Flow* e os elementos de BPMN que participam podem ser *pools*, tarefas, subprocessos ou eventos, ou seja objectos BPMN que podem ser emissores ou receptores de *message flows*.

Vamos de seguida apresentar na **Figura 47** o correspondente diagrama de classes;

Diagrama:

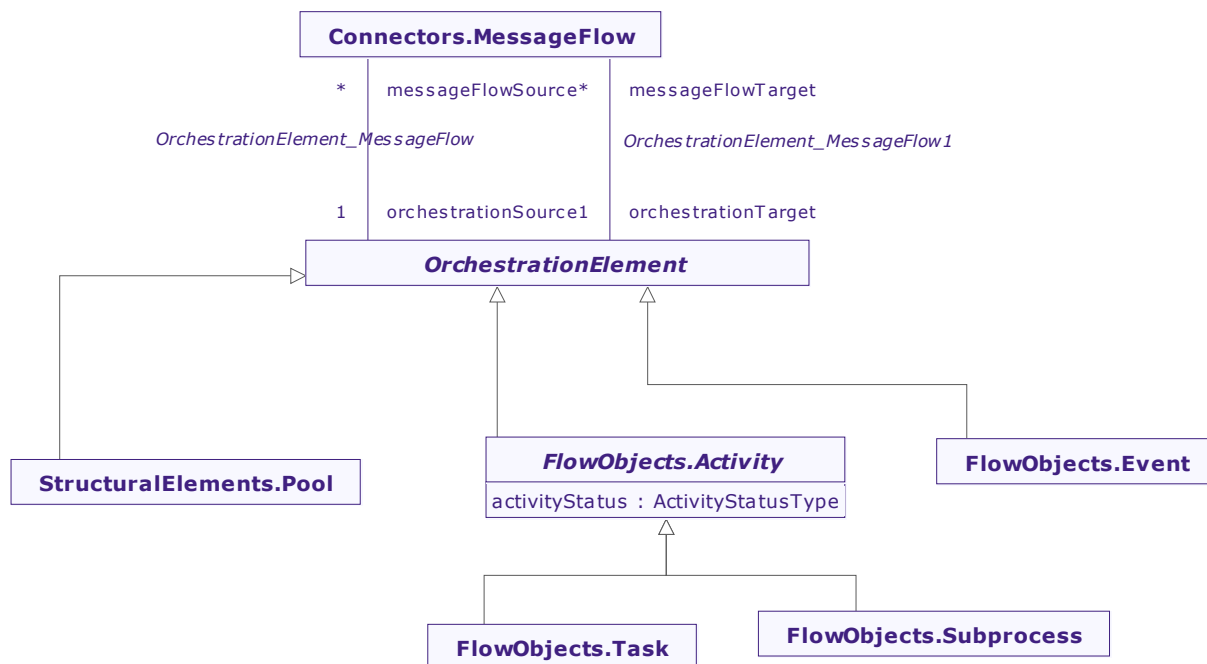


Figura 47 - Pacote de orquestração BPMN

Meta-Classes:

Este pacote só tem uma classe própria, a *OrchestrationElement* que serve para organizar as classes que podem intervir na orquestração. Esta é realizada pelas associações *target* e *source* com a classe *MessageFlow*.

Instanciação:

Vamos de seguida apresentar, na **Figura 48** um extracto do exemplo completo apresentado anteriormente na **Figura 28**. De seguida apresentamos o modelo de objectos que corresponde ao nível M1, na **Figura 49**.

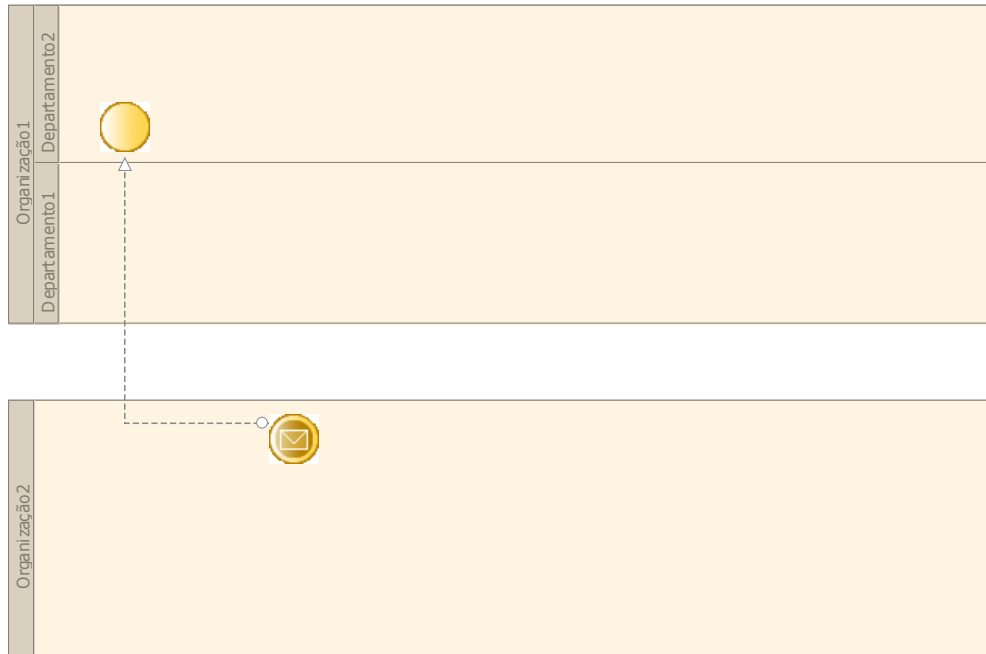


Figura 48 - Extracto do processo exemplo relativo à orquestração

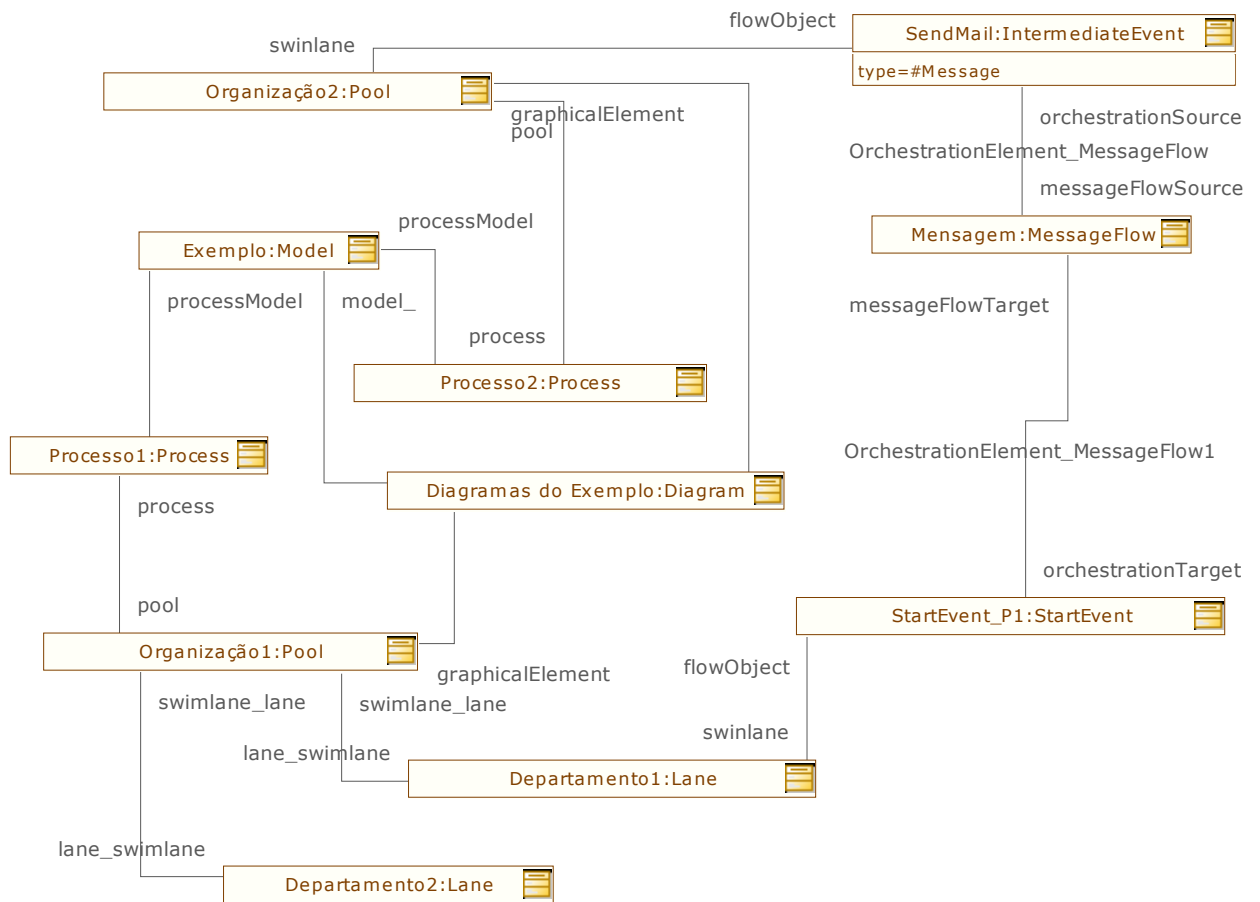


Figura 49 - Modelo de objectos do exemplo de orquestração

4.2.11 Pacote Simulação

Propósito:

O pacote simulação destina-se a possibilitar a simulação de cenários de execução após a definição do modelo, prevenindo a execução de processos com custos demasiado elevados, ou processos sem viabilidade devido, por exemplo, à existência de *bootlenecks* ou *loops*. Este pacote permite a associação de custos, recursos e distribuição de rotas dos *tokens* criados em tempo de execução.

Diagrama:

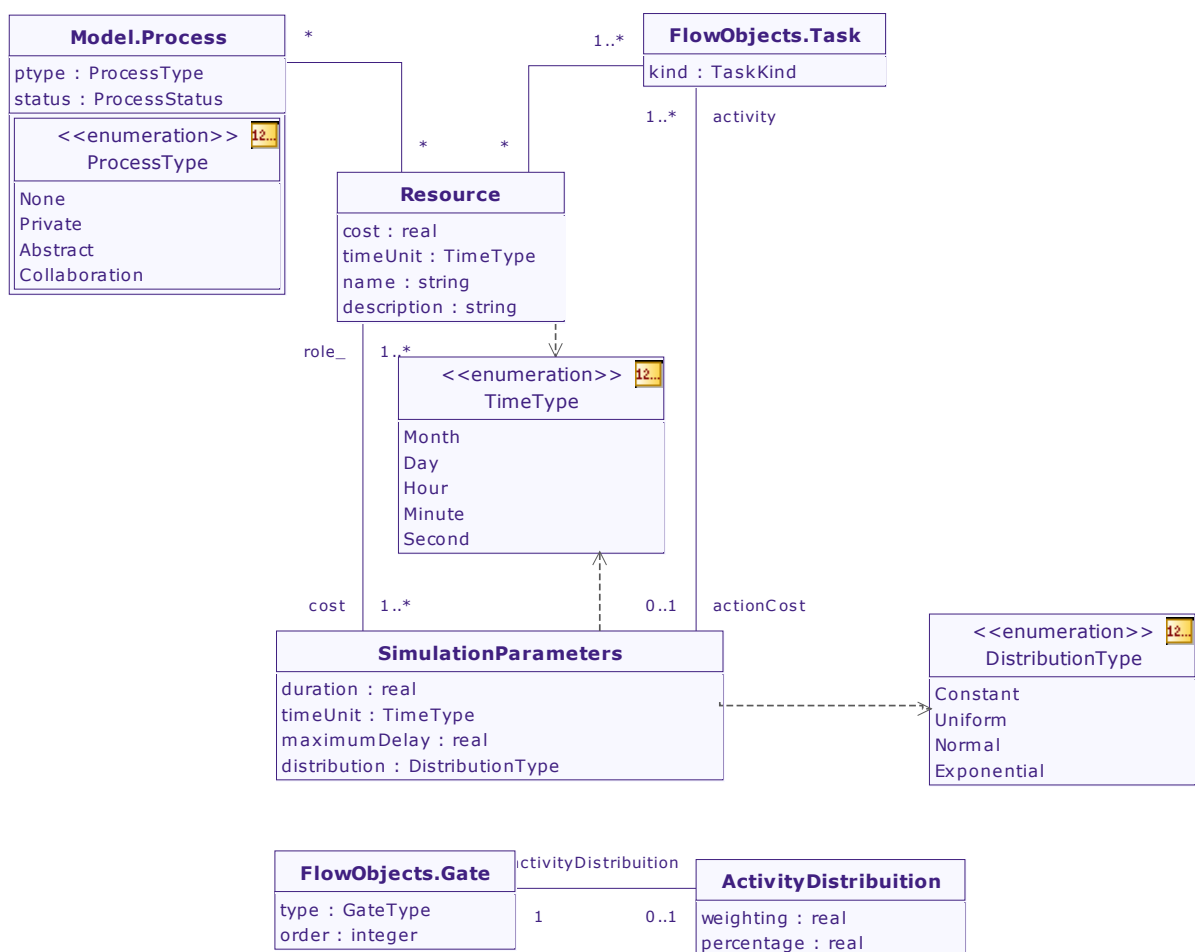


Figura 50 - Pacote de simulação BPMN

Meta-Classes:

- A. *Simulation Parameters* – Permite parametrizar o custo de recurso ou a duração da tarefa, podendo inclusivamente qualquer um destes parâmetros não ser constante ao longo do tempo, mas seguir uma função Uniforme, Normal ou Exponencial;
- B. *Activity Distribution* – Atribui a cada *gate* um peso ou percentagem que permite uma distribuição de fluxo;

C. *Resource* – Recurso e respectivo custo utilizado na realização da actividade.

4.2.12 Pacote Conexão Interna

Propósito:

O pacote de conexão interna tem como objectivo ligar os elementos de BPMN existentes na mesma *pool*. Também é seu objectivo ligar um caso especial de elemento BPMN, a *text annotation* que podem ser ligadas a qualquer elemento BPMN incluindo as próprias *pools*.

Temos assim 2 casos distintos neste pacote:

1. Ligações de elementos BPMN na *pool*;
2. Ligações das *text annotations*.

Diagrama:

Apresentamos de seguida os diagramas dos dois casos, na **Figura 51 - Ligações de elementos BPMN na *pool*** e na **Figura 52 - Ligações das *text annotations***;

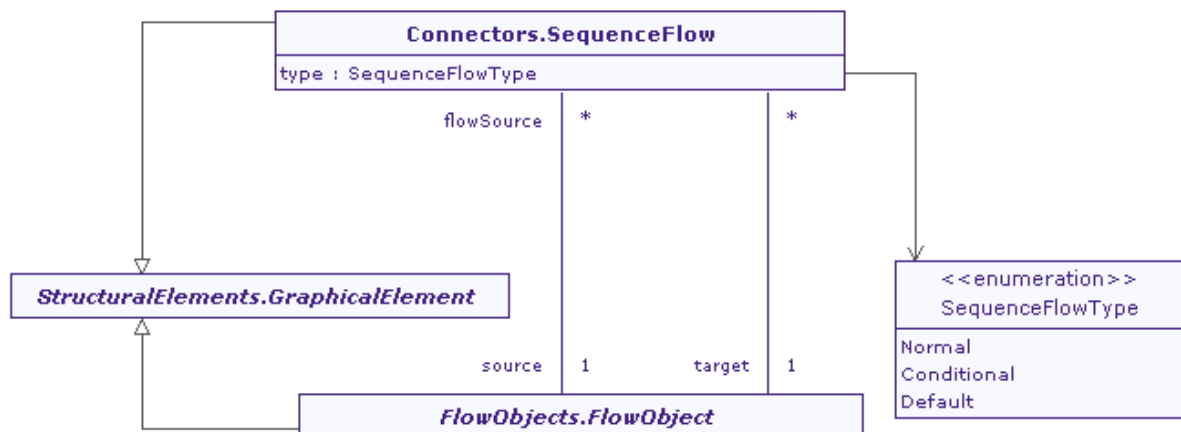


Figura 51 - Ligações de elementos BPMN na *pool*

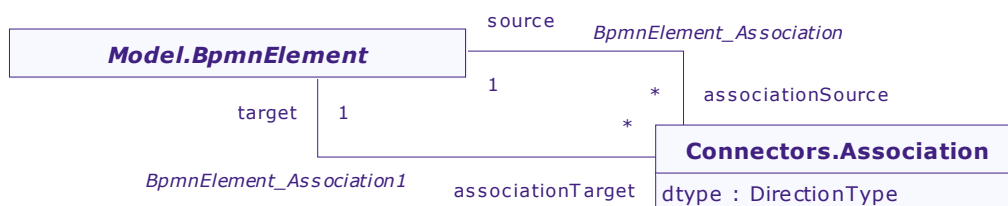


Figura 52 - Ligações das *text annotations*

Meta-Classes:

Os elementos BPMN são ligados dentro da *pool* com *sequence flows* e as *text annotation* são sempre ligadas aos elementos BPMN com um tipo de conector, designado Associação.

Instanciação:

Apresentamos de seguida um exemplo de modelação com BPMN na **Figura 53 - Exemplo de Ligação Interna com Sequence Flow S1** e na **Figura 54 - Exemplo de Ligação Interna Com Associações** que são dois extractos do exemplo apresentado anteriormente na Figura 28 - **Exemplo genérico de processo BPMN**. Estes extractos correspondem a um *sequence flow* designado por S1 no 1º exemplo e duas *text annotation* designadas notas 2 e notas 3.



Figura 53 - Exemplo de Ligação Interna com Sequence Flow S1

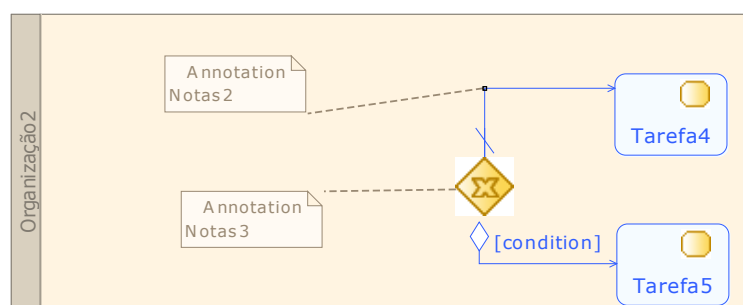


Figura 54 - Exemplo de Ligação Interna Com Associações

Não existem classes para obter as ligações entre os elementos BPMN, estas ligações são obtidas por meio de associações designadas por *target* e *source*. Estas associações estabelecem a ligação do *sequence flow* com a origem e com o destino. Mostramos na **Figura 55** e na **Figura 56** modelos de objectos para estes extractos, respectivamente:

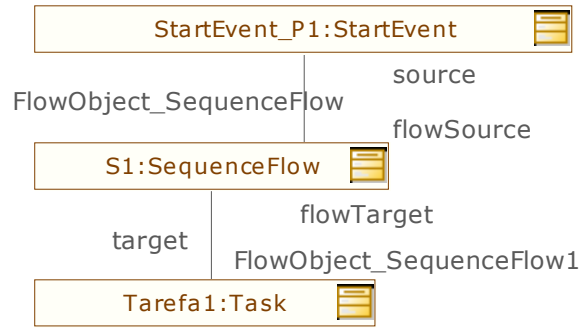


Figura 55 - Modelo de objectos exemplo de ligação interna

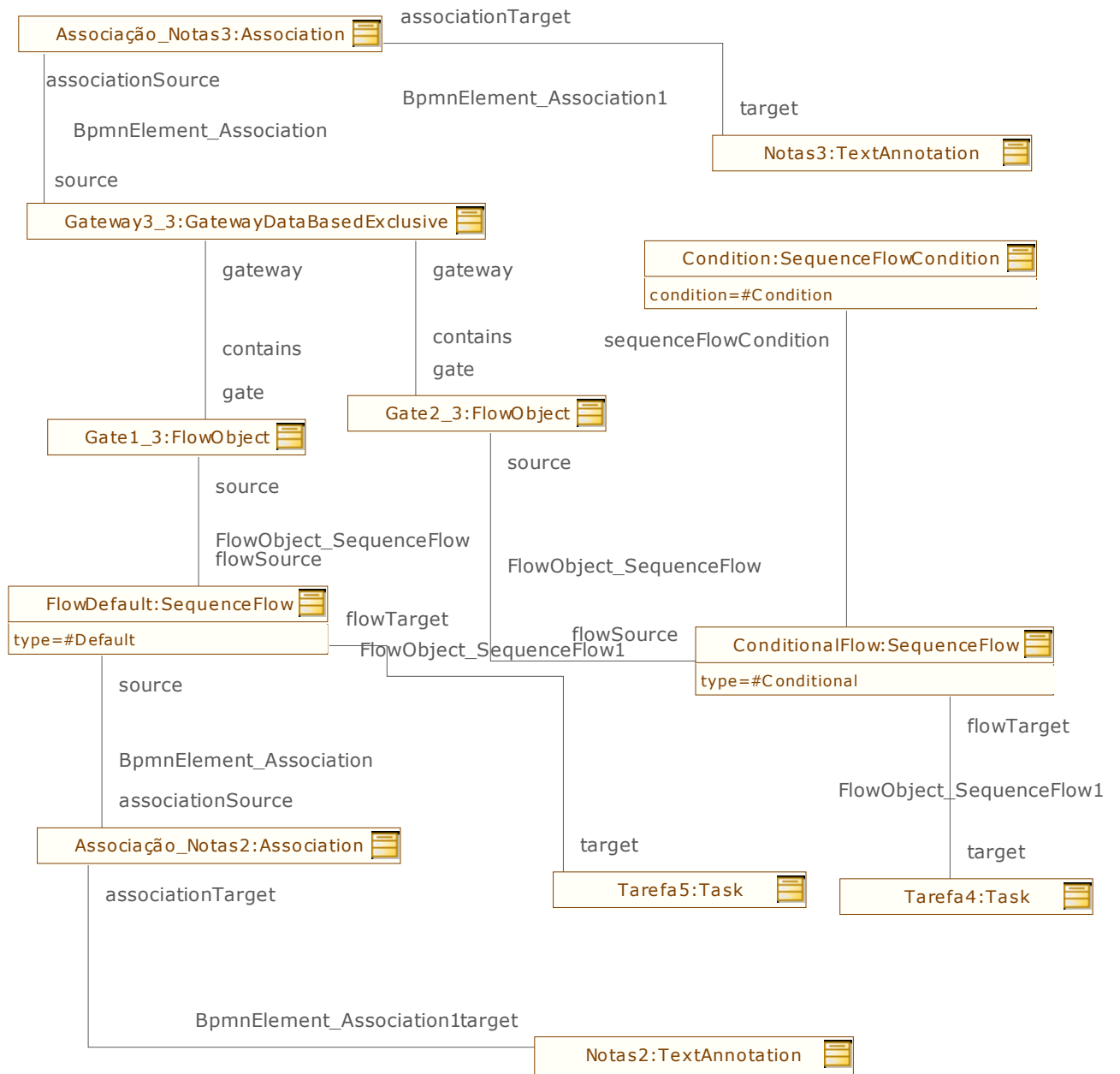


Figura 56 - Modelo de objectos com exemplos de associações

4.2.13 Pacote Casos de Utilização

Propósito:

Neste pacote estabelecemos uma relação entre os casos de utilização e o BPMN. Um caso de uso representa uma interação entre um actor (humano ou máquina) e um sistema. Por exemplo: a **Figura 3** É um exemplo de caso de uso. Cada caso de uso tem uma descrição, para especificar a funcionalidade que irá ser construída por sequências de mensagens que são trocadas entre os sistemas e um ou mais actores. Um caso de uso pode "incluir" outra funcionalidade de caso de uso ou "estender" outro caso de uso com o seu próprio comportamento, e generalização ou especialização entre actores.

Apresentamos em seguida na **Figura 57** o diagrama de classes correspondente;

Diagrama:

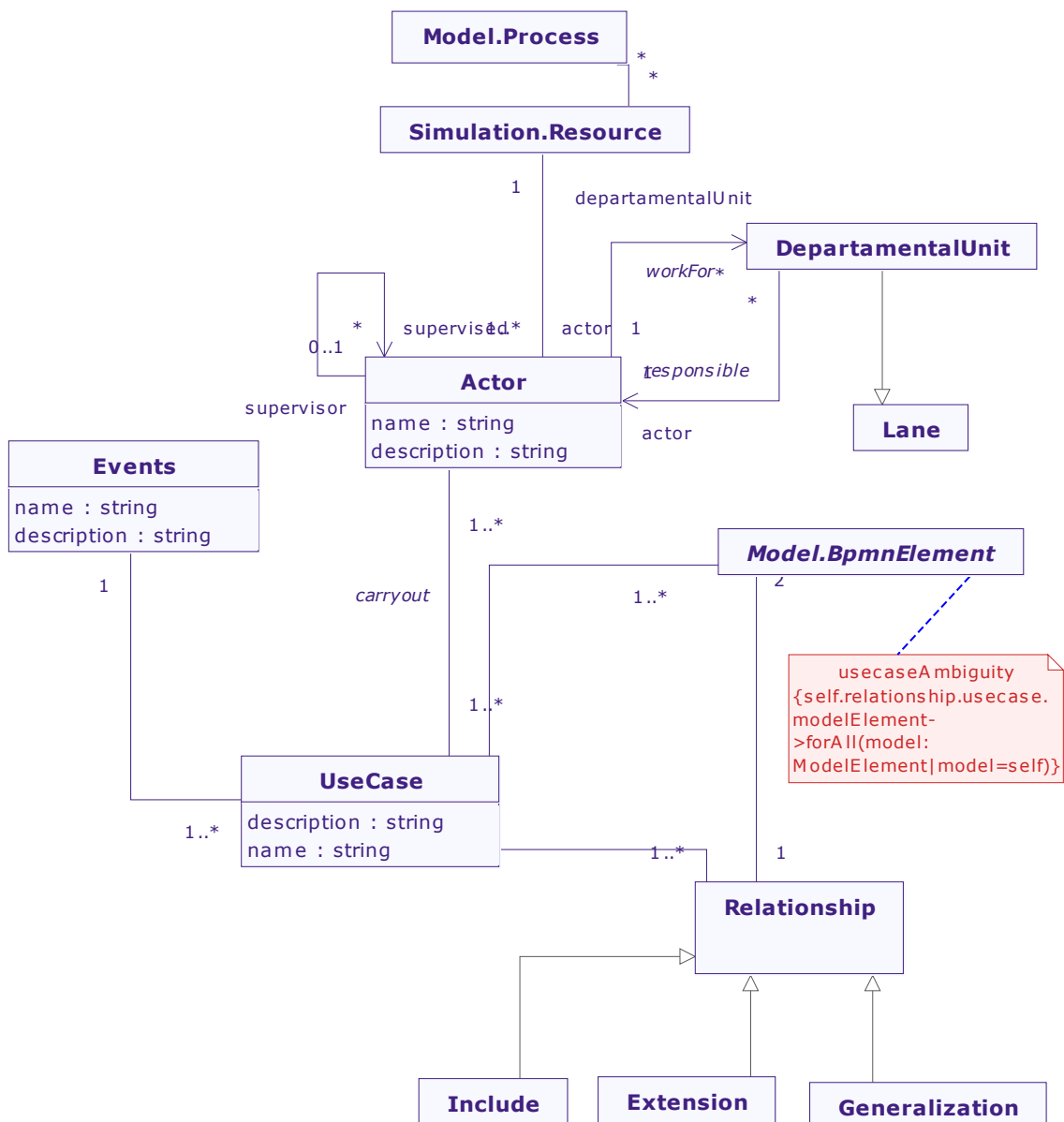


Figura 57 - Diagrama do pacote Casos de utilização

Meta-Classes:

- A. *Use Case* - Corresponde a um diagrama de caso de uso em UML. Um caso de é uma descrição do comportamento de um sistema, e que responde a um pedido que se origina de fora desse sistema;
- B. *Actor* - Actor é algo que interage com o sistema, têm um papel externo e são quem iniciam (e quem respondem) aos casos de utilização. Por exemplo: fazem o pedido num restaurante, comem, bebem ou pagam. Tipicamente, um actor representa um papel que um ser humano, um outro processo, um outro sistema, ou até um dispositivo de hardware, desempenha ao interagir com o sistema. Corresponde a um papel específico e por exemplo uma pessoa pode desempenhar vários papéis;
- C. *Departmental Unit* - Departamento ou organização no qual se insere o actor;
- D. *Relation ship* - São os tipos de relações que podem ocorrer entre os elementos do caso de uso.

4.3 Restrições do Metamodelo

Tal como no caso da ontologia, vamos utilizar a linguagem de restrições OCL [OMG1997] para aumentar a precisão, retirando ambiguidade à especificação do metamodelo. É habitual designar estas restrições como regras de boa formação de modelos (*well-formedness rules*). As mesmas serão formuladas através de invariantes, cujo contexto é identificado para cada caso.

4.3.1 Expressões do Pacote Modelo

Expressão 1 - Se houver um evento final, então deve haver pelo menos um evento de início;

```
context Process
```

```
    inv startandEndEvent:
```

```
        self.bpmnElements()->select(oclIsKindOf(FlowObjects_EndEvent))->size() > 0 implies  
            self.bpmnElements()->select(oclIsKindOf(FlowObjects_StartEvent))->size() > 0
```

Expressão 2 - Todos os outros *Flow objects* que não *start event* devem ser alvo (destino) de pelo menos um fluxo de sequência.

```
context Process
```

```
inv mustBeTarget:
```

```
self.bpmnElements()->select(oclIsKindOf(FlowObjects_StartEvent))->size() > 0 implies  
self.bpmnElements()->select(oclIsKindOf(FlowObjects_FlowObject))->  
reject(oclIsKindOf(FlowObjects_StartEvent)).oclAsType(FlowObjects_FlowObject).flowTarget  
-> notEmpty()
```

4.3.2 Expressões Pacote Elementos Estruturais

Expression 3 – Com um *MessageFlow* não é possível ligar para os objectos que estão dentro do mesmo *pool*.

```
context Pool
```

```
inv messageflowimpliesorchestration:
```

```
self -> select(oclIsKindOf(messageFlow)).oclAsType(messageFlow)->size() > 0 implies  
self -> select(oclIsKindOf(FlowObjects)).oclAsType(FlowObjects).flowConnectorTarget ->  
select(oclIsKindOf(messageFlow)).oclAsType(messageFlow)->size() -> Empty()
```

4.3.3 Expressões do Pacote Connectores

Expressão 4 - Se um *SequenceFlow* não é "condicional", então não tem condições associadas.

```
context SequenceFlow
```

```
inv NonConditionalFlowsHaveNoCondition:
```

```
self.type <> #SequenceFlowType_Conditional implies  
self.sequenceFlowCondition.isUndefined
```

Expressão 5 - Se um *SequenceFlow* é "condicional", então ele tem uma condição associada.

```
context SequenceFlow
```

```
inv ConditionalFlowsHaveCondition:
```

```
self.type = #SequenceFlowType_Conditional implies self.sequenceFlowCondition.isDefined
```

Expressão 6 - Se um *MessageFlow* é uma fonte, não pode ser um alvo em simultâneo.

```
context MessageFlow
    inv MessageFlowTargetDistinctMessageFlowSource:
        self.orchestrationSource <> self.orchestrationTarget
```

4.3.4 Expressões do Pacote *Actividade*

Expressão 7 - A tarefa seguinte não pode ser ela mesmo.

```
context Task
    inv taskAmbiguity:
        self.actionCost.role.activity->forAll(activity:Task | activity=self)
```

4.3.5 Expressões do Pacote *Gateway*

Expressão 7 - Se o Gate é de input, então ele não tem condições associadas

```
context Gate
    inv InputGatesHaveNoCondition:
        self.type = #GateType_Input implies self.gateCondition.isUndefined
```

Expressão 8 - Todas as *gates* devem pertencer à mesma *pool* que o *gateway*.

```
context Gate
    inv GateAndGatewaysUnite:
        self.gateway.swimlane = self.swimlane
```

Expressão 9 - Se o *Gate* é *Output* então deve ter uma condição associada.

```
context Gate
    inv OutputGatesHaveCondition:
        self.type = #GateType_Output implies self.gateCondition.isDefined
```

4.3.6 Expressões do Pacote Event

Expressão 10 - Todos os eventos devem ser *catching* or *throwing*.

```
context Event
    inv EventDynamics:
        self.eventDynamic.isDefined()
```

Expressão 11 - Um evento de *catching* não pode ser uma fonte de um *message flow*.

```
context Event
    inv CatchingEvent:
        self.eventDynamic = #EventDynamicType_Catching implies
            self.messageFlowSource->isEmpty()
```

Expressão 12 - Um evento de *throwing* não pode ser alvo de um *message flow*.

```
context Event
    inv ThrowingEvent:
        self.eventDynamic = #EventDynamicType_Throwing implies
            self.messageFlowTarget->isEmpty()
```

Expressão 13 - Um *end event* não pode ser uma fonte de *flow connector* e deve ser uma *target* de pelo menos um *flow connector*.

```
context EndEvent
    inv endEventsAreTargetsButNotSources:
        self.flowConnectorSource -> isEmpty() and not self.
```

Expressão 14 – Um *end Event* deve ser do tipo *Throwing*.

```
context EndEvent  
  
    inv EndEventDetailsRules:  
  
        self.eventDynamic = #EventDynamicType_Throwing
```

Expressão 15 -Um *end event* não pode ser uma *source* de um conector de fluxo e deve ser um *target* de pelo menos um fluxo de conector.

```
context EndEvent  
  
    inv endEventsAreTargetsButNotSources:  
  
        self.flowSource -> isEmpty() and self.flowTarget -> notEmpty()
```

Expressão 16 -Um evento intermediário deve ser uma fonte de pelo menos um conector de fluxo e deve ser *target* de pelo menos um fluxo de conector.

```
context IntermediateEvent  
  
    inv intermediateEventsAreBothSourcesAndTargets:  
  
        self.flowSource -> notEmpty() and self.flowTarget -> notEmpty()
```

Expressão 17 – Um evento intermédio só pode ser ligada (*attached*) uma vez.

```
context AttachedIntermediateEvent  
  
    inv UniqueAttachement:  
  
        not(self.attachingActivity.isDefined() and self.attachingGroup.isDefined())
```

Expressão 18 – Um *start event* não pode ser um *target* de um *flow connector* e tem de ser a *source* de pelo menos um *flow connector*.

```
context StartEvent  
  
    inv startEventsAreSourcesButNotTargets  
  
        self.flowTarget->isEmpty() and self.flowSource->notEmpty()
```

Expressão 19 – Os *Start Events* são sempre do tipo *catching*.

```
context StartEvent
```

```
    inv StartEventDetailsRules:  
        self.eventDynamic = #EventDynamicType_Catching
```

4.3.7 Expressões Pacote Use Case

Expressão 20 – O elemento não pode ser ele mesmo.

```
context BpmnElement
```

```
    inv usecaseAmbiguity:  
        self.relationship.usecase.modelElement->forAll(model:ModelElement|model=self)
```

4.4 Discussão dos Resultados Metamodelo BPMN

A proposta de metamodelo BPMN apresentada neste capítulo:

- I. Tem a intenção de oferecer um compromisso entre as especificações completas de BPMN OMG e a capacidade tática;
- II. Define os elementos de BPMN com a ajuda da linguagem de restrições OCL;
- III. Foi previamente validada pela instanciação de um exemplo realista, que reforça as regras do metamodelo;
- IV. Abrange a notação BPMN, Simulação e Casos de utilização.

4.5 Validação e Contexto Metamodelo BPMN

A presente dissertação foi desenvolvida no âmbito do grupo de investigação QUASAR do CITI do DI/FCT/UNL, em conjunto com mais duas outras dissertações com o mesmo orientador [Costa, 2010, Porciúncula, 2010]. Na **Figura 58** apresentamos a colaboração entre estas 3 dissertações;

Modelação (Objecteeting – UML/USE) – No âmbito da dissertação aqui elaborada foi criado o metamodelo, instanciado com pequenos exemplos e desenvolveu-se um trabalho de colaboração com outras duas dissertações com o objectivo de validar e desenvolver os metamodelos propostos. Como se pode ver na **Figura 58**, existe um *feed-back* a partir do qual se corrigiu e ampliou o metamodelo, fazemos de seguida a descrição da Figura;

Transformação (Objecteeting – UML/OCL) – Em [Porciúncula, 2010] trabalhou-se sobre o metamodelo de BPMN e definiram-se métricas de complexidade de processos de ITSM e instanciou-se o metamodelo com processos *As-Is* e *To-be* de toda uma organização, realizando estudos de complexidade.

Instanciação (Objecteeting – Visio+Excell+VBA) – Em [Costa, 2010] também se instanciou o metamodelo de BPMN, mas neste caso com modelos de grande dimensão (milhares de meta-objects e meta-ligações) no domínio da gestão de projectos informáticos.

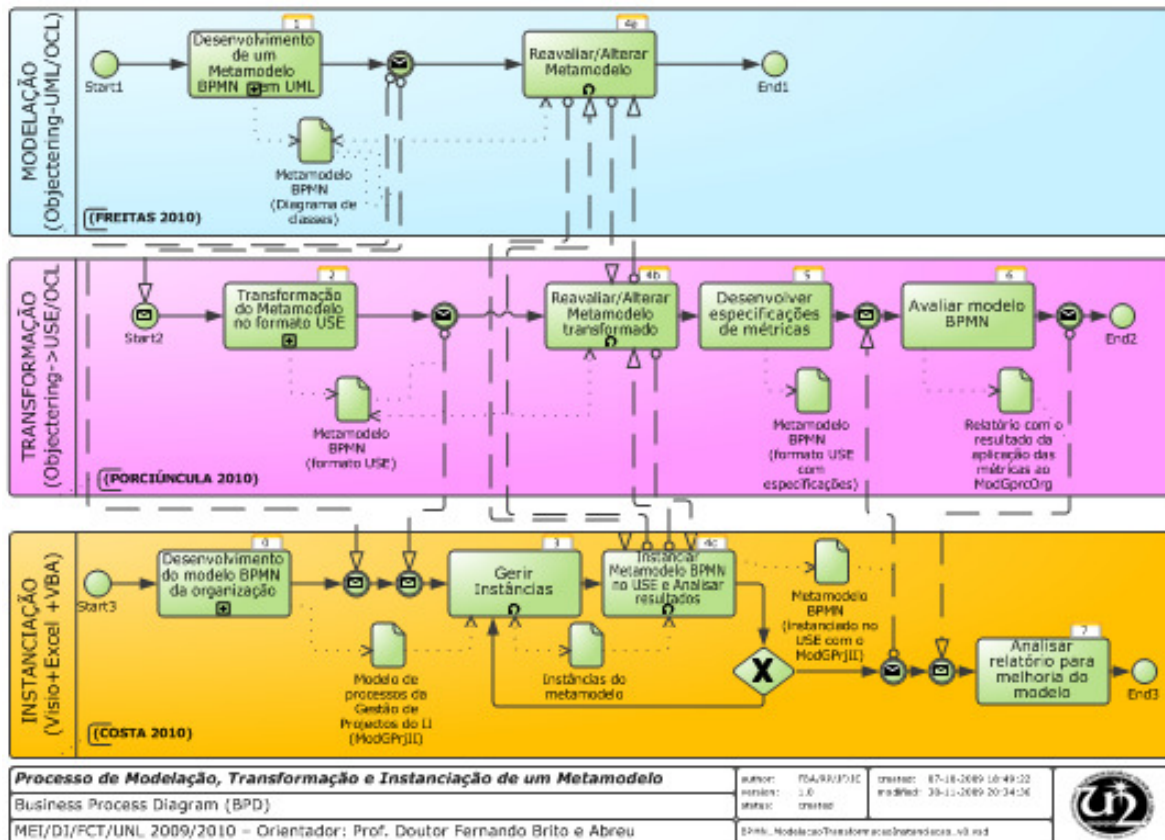


Figura 58 - Processo de Modelação, Transformação e Instanciação do Metamodelo [Costa, 2010]

4. Dinâmica

A nossa dissertação é de metamodelação e cria a base do projecto, a dissertação de Costa denominada “MGPSI – Metodologia de Gestão de Projectos Aplicada ao Desenvolvimento de Sistemas de Informação”, vai instanciar o metamodelo com um caso real de modelação com 14 processos reais, que apresentamos na **Figura 59** um dos processos como exemplo.

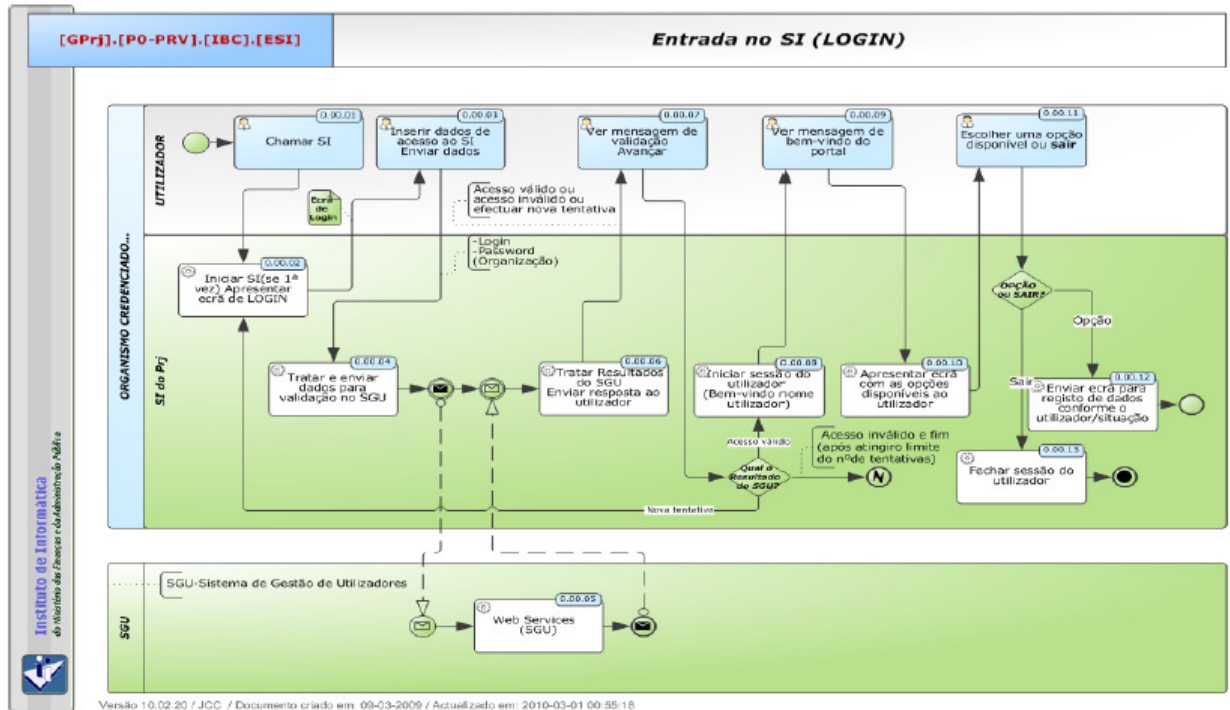


Figura Ap.A4 – Subprocesso “Entrada no SI” (ESI)

Figura 59 - Exemplo de processo MGPSI [Costa, 2010]

A partir do exemplo do MGPSI, Costa inseriu no metamodelo 2.335 instanciações sobre as classes e 3.281 associações, num total de quase 6000 objectos. A lista de meta-objectos está representada na **Figura 60**.

```

use.bat
use> info state
State: state#1
class                                     : #Objects + #Objects in subclasses
-----
<Artifacts_Artifact>                       : 0 130
Artifacts_DataObject                       : 82 82
Artifacts_Group                             : 11 11
Artifacts_TextAnnotation                   : 37 37
Connectors_Association                     : 143 143
Connectors_MessageFlow                     : 175 175
Connectors_SequenceFlow                    : 742 742
Connectors_SequenceFlowCondition           : 0 0
Date                                         : 0 0
<FlowObjects_Activity>                     : 0 321
FlowObjects_AttachedIntermediateEvent      : 0 0
FlowObjects_ComplexGateway                 : 0 0
FlowObjects_ConditionGate                  : 0 0
FlowObjects_EndEvent                       : 114 114
FlowObjects_Event                          : 0 390
<FlowObjects_FlowObject>                   : 0 1.050
FlowObjects_Gate                            : 271 271
<FlowObjects_Gateway>                     : 0 68
FlowObjects_GatewayDataBasedExclusive     : 29 29
FlowObjects_GatewayDataBasedInclusive     : 4 4
FlowObjects_GatewayEventBasedExclusive    : 16 16
FlowObjects_GatewayExclusive              : 0 45
FlowObjects_IntermediateEvent              : 207 207
FlowObjects_ParallelGateway                : 19 19
FlowObjects_StartEvent                     : 69 69
FlowObjects_SubProcess                      : 94 94
FlowObjects_Task                            : 227 227
<Model_BpmnElement>                       : 0 2.320
Model_Diagram                              : 14 14
Model_InputSets                            : 0 0
Model_Model                                 : 1 1
Model_OutputSets                           : 0 0
Model_Process                              : 1 95
Object                                       : 0 2.335
<Orchestration_OrchestrationElement>      : 0 773
PackageUseCase_actor                       : 0 0
PackageUseCase_Events                      : 0 0
PackageUseCase_Extension                   : 0 0
PackageUseCase_Generalization              : 0 0
PackageUseCase_Include                     : 0 0
PackageUseCase_Intervient                  : 0 0
PackageUseCase_Relationship                : 0 0
PackageUseCase_UseCase                     : 0 0
Simulation_ActivityDistribution             : 0 0
Simulation_Resource                        : 0 0
Simulation_SimulationParameters            : 0 0
StructuralElements_GraphicalElement        : 0 2.319
StructuralElements_Lane                    : 18 18
StructuralElements_Pool                    : 61 61
<StructuralElements_Swimlane>              : 0 79
SupportingElements_Properties_Property     : 0 0
SupportingElements_SupportingElement       : 0 366
-----
total                                       : 2.335
  
```

Figura 60 - Meta-objetos carregados MGPSI

A dissertação realizada pela Raquel Porciúncula denominada “Governança e Modelação de Processos de TI” propôs métricas para medir os processos “As-Is” e “To-Be” sobre o metamodelo apresentado nesta dissertação e instanciou mais dois processos, que se podem ver em D: Métricas da Gestão de Incidentes e Gestão de Problemas. Ambos os trabalhos colaboraram na validação desta dissertação. Apresentamos na **Figura 61** o esquema de colaboração das dissertações.

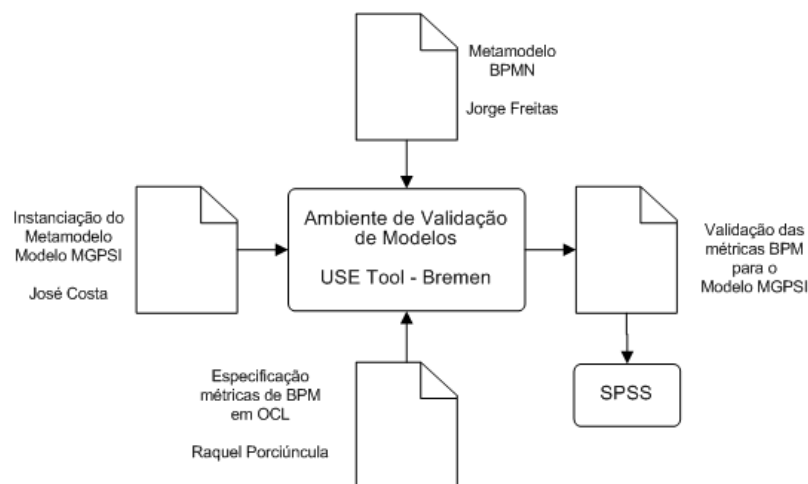


Figura 61 - Colaboração dissertações [Porciúncula, 2010]

4.6 Discussão dos resultados relativos ao metamodelo de BPMN

Esta proposta de metamodelo:

- I. Tem a intenção de oferecer um compromisso entre as especificações completas de BPMN OMG e o STP_Eclipse;
- II. Define os elementos de BPMN com a ajuda da linguagem de restrições OCL;
- III. Foi validada pela instanciação preliminar de um exemplo realista, que reforça as regras do metamodelo;
- IV. Abrange a notação BPMN, simulação e casos de utilização.

[Esta página foi intencionalmente deixada em branco]

Conclusões e Trabalho Futuro

5. CONCLUSÕES E TRABALHO FUTURO	124
5.1 CONTEXTO.....	124
5.2 COMPARAÇÃO DOS OBJECTIVOS PROPOSTOS E ALCANÇADOS	124
5.3 REVISÃO DAS CONTRIBUIÇÕES	125
5.4 PUBLICAÇÕES CIENTIFICAS	126
5.4.1 AN ONTOLOGY FOR IT SERVICES	126
5.4.2 DEFINITION AND VALIDATION OF COMPLEXITY METRICS FOR ITSM PROCESS MODELS	127
5.4.3 IDENTIFICAR OS NÍVEIS DE APTIDÃO DA GESTÃO DE SERVIÇOS DE TI USANDO AS MELHORES PRÁTICAS	127
5.4.4 MGPSI – METODOLOGIA DE GESTÃO DE PROJECTOS APLICADA AO DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO	
128	
5.5 TRABALHO FUTURO	128

Este capítulo apresenta as conclusões da dissertação e trabalho futuro. Um ponto importante é o contexto do desenvolvimento da dissertação que decorreu em colaboração com outras dissertações em curso na mesma equipa de investigação.

5. Conclusões e trabalho Futuro

5.1 Contexto

Esta dissertação foi desenvolvida no âmbito da equipa de investigação QUASAR (*QUantitative Approaches in Software engineering And Reengineering*), onde em 2008 se iniciou uma linha de investigação denominada *Quantitative IT Service Management* da qual já resultaram várias publicações e dissertações, que podem ser encontradas em <http://ctp.di.fct.unl.pt/QUASAR>.

Tal como já foi referido no capítulo anterior (metamodelo de BPMN), os resultados desta dissertação são instrumentais para atingir objectivos de investigação em outras dissertações, muito em particular o caso de [Costa, 2010, Porciúncula, 2010], entretanto já concluída, onde o metamodelo aqui proposto foi extensivamente utilizado. Por outro lado, o trabalho desenvolvido em [Costa, 2010, Porciúncula, 2010] foi também instrumental para a validação do mesmo metamodelo, ao fornecer um exemplo muito complexo (2.335 meta-objects e 3.281 meta-ligações entre os mesmos) de modelação em BPMN. O metamodelo foi assim instanciado com esse exemplo, o que permitiu detectar e corrigir muitas das suas limitações e inconsistências.

5.2 Comparação dos Objectivos Propostos e Alcançados

Nesta secção por intermédio da **Tabela 22** apresentamos uma comparação entre os objectivos propostos e a sua prossecução.

Objectivos Propostos	Objectivos Alcançados
Definir e representar formalmente conceitos de ITSM;	Na Ontologia ITSM apresentada são descritos os conceitos de: <ul style="list-style-type: none"> • Catálogo, características do serviço ou produto, indicador, nível de serviço, observação, prioridade, <i>help-desk</i>, acção, cliente, fornecedor.
Representar todos os conceitos abrangidos pelos SLA;	Além das características descritas na Ontologia ITSM características próprias de SLA foram adicionadas e descritas: <ul style="list-style-type: none"> • Contracto, participantes no SLA, responsabilidade, Nível de escalonamento do problema, serviço ou produto contratualizado e respectivo

	indicador contratualizado (para comparação com o observado).
Representar a dinâmica dos conceitos;	<p>Criamos um metamodelo que permite o desenho de processos em BPMN, e que representa conceitos como;</p> <ul style="list-style-type: none"> • Pool, Lane, Conectores, Eventos, Actividades, Gateways, dados, grupos, anotações. <p>Para além de representar os conceitos de desenho de processos, apresenta outros conceitos como simulação que permite atribuir recursos a cada tarefa (por exemplo).</p>
Compatibilizar conceitos como produtos e serviços não só de um ponto de vista estático, mas também dinâmico.	O Metamodelo de BPMN permite definir e representar os processos de fornecedores, clientes e a sua interacção (entre processos) e permite representar conceitos como SLA, Serviços e Produtos essencialmente pelos objectos de dados.

Tabela 22 – Objectivos propostos e sua prossecução

5.3 Revisão das Contribuições

É possível identificar as seguintes contribuições científicas nesta dissertação:

- **Taxonomia ITSM e sua aplicação** – Criámos uma taxonomia para avaliar estudos ontológicos no domínio da gestão de serviços de tecnologias da informação (ITSM). Com base nessa taxonomia efectuámos o levantamento dos trabalhos relacionados com estudos ontológicos no de ITSM, o que nos permitiu identificar algumas limitações sobre o actual estado-da-arte.
- **Ontologia de ITSM** - Como já referido, apesar dos esforços em curso e já efectuados sobre a criação de um corpo de conhecimentos sobre ITSM, esses esforços ainda não fornecem uma base para a avaliação objectiva de trabalhos científicos nesta área e também não apresentam o formalismo nas suas definições que permita a

prossecação de estudos empíricos de natureza quantitativa, baseados no método científico. Para atenuar as lacunas detectadas propusemos uma ontologia para o domínio de ITSM. Para conceder a precisão necessária na formalização desta ontologia utilizámos diagramas de (meta)classes em UML, complementados com regras de boa formação expressas através de invariantes em linguagem OCL (*Object Constraint Language*).

- **Revisão do estado da arte em meta-modelação de BPMN** – Efectuámos um estudo comparativo das únicas duas propostas conhecidas neste área. O metamodelo oficial da OMG é excessivamente complexo para aplicação prática no contexto da nossa linha de investigação (a justificação detalhada deste aspecto cai fora do âmbito desta dissertação) e, na altura em que decorreu o trabalho correspondente nesta dissertação, ainda não estava concluído. Por outro lado, a outra proposta conhecida, proveniente de uma iniciativa a decorrer no seio da iniciativa Eclipse denominada *STP BPMN Modeler* (www.eclipse.org/bpmn), é demasiado incompleta e carece de documentação.
- **Metamodelo de BPMN** – Pelas razões supracitadas, criámos uma versão independente, embora inspirada na versão da OMG, de um metamodelo para BPMN. Este metamodelo foi preliminarmente validado através da criação de um pequeno exemplo, apresentado nesta dissertação e, posteriormente, com um modelo BPMN de grandes dimensões produzido no âmbito de outra dissertação [Costa, 2010, Porciúncula, 2010]. A validação foi realizada recorrendo à ferramenta USE (*UML Specification Environment*) da Universidade de Bremen.
- **Publicações científicas** – O trabalho desenvolvido no âmbito desta dissertação contribuiu, pelo menos parcialmente, para dar corpo a vários trabalhos científicos (dois artigos em eventos internacionais com revisão por pares e duas dissertações de mestrado), que se descrevem abreviadamente na secção seguinte.

5.4 Publicações Científicas

5.4.1 An Ontology for IT Services

[Freitas, et al., 2008]

Artigo apresentado na conferência JISBD'2008 (<http://www.sistedes.es/jisbd2008>)

Resumo:

O levantamento de trabalhos relacionados com ontologias no domínio dos serviços de TI, nos permitiu identificar alguns sérias limitações no actual estado-da-arte através de um conjunto de seis dimensões de maturidade. Para minimizar estas lacunas este trabalho propõe uma ontologia formal de TI serviços. Para conceder a esta ontologia precisão temos expressas regras de boa formação, com a restrição OCL linguagem. A ontologia proposta é instanciada para ilustrar a sua validade com exemplos realistas.

5.4.2 Definition and Validation of Complexity Metrics for ITSM Process Models

[Abreu, et al., 2010]

Artigo apresentado na conferência QUATIC'2010 (<http://www.fe.up.pt/quatic2010>)

Resumo:

O Business Process Modeling (BPM) é usado para especificar processos IT Service Management (ITSM). A linguagem da OMG designada BPMN é cada vez mais adoptada para definir cenários "As-Is" e "To-Be" em ITSM processos. Para determinar a melhoria entre os dois cenários, é preciso quantificar cada um dos modelos correspondentes.

Este artigo propõe um conjunto de métricas para os modelos BPMN. Essas métricas foram definidas usando a abordagem *Metamodel-Driven Measurement* (M2DM) que fornece precisão e objectividade. De acordo com essa abordagem, as métricas foram especificados com a linguagem de restrições OCL, em cima de um metamodelo BPMN leve. Esse metamodelo foi instanciado com vários estudos de caso, variando a partir de exemplos de processos para um grande modelo BPMN abrangendo cerca de seis mil objectos. Para cada instanciação nós executamos a definição de cada métrica OCL para recolher o valor correspondente. A colinearidade das métricas é analisada e fomos capazes de identificar um pequeno conjunto de métricas, que serão úteis para realizar trabalhos de investigação sobre a complexidade dos processos de ITSM.

5.4.3 Identificar os níveis de aptidão da gestão de serviços de TI usando as melhores práticas

[Porciúncula, 2010]

Dissertação de Mestrado em Engenharia Informática, FCT/UNL

Resumo:

Dissertação que propôs um método para avaliar o grau de maturidade da governança de TI dentro de uma organização. Foram recolhidas métricas da complexidade dos processos de TI. Para isso formalizou-se um conjunto de indicadores quantitativos que permitem exprimir a complexidade de um processo, com recurso à linguagem OCL (*Object Constraint Language*), sobre um metamodelo BPMN (*Business Process Modeling*

Notation). Estas métricas foram correlacionadas com os níveis de maturidade. Baseado nestas correlações, foi criado um modelo que interliga a governança de TI com o nível de complexidade dos processos de TI.

5.4.4 MGPSI – Metodologia de Gestão de Projectos Aplicada ao Desenvolvimento de Sistemas de Informação

[Costa, 2010]

Dissertação Mestrado em Engenharia Informática na área de Engenharia de Software

Resumo:

Dissertação que propôs a formalização de uma metodologia de gestão de projectos, usando a modelação, simulação de processos. Faz uma análise e um diagnóstico do estado actual da gestão de processos e projectos no seio da Administração Pública em Portugal;

5.5 Trabalho Futuro

Identificam-se seguidamente as próximas etapas que o trabalho desenvolvido até ao momento permite elencar:

- Novas publicações; está já prevista a colaboração com os outros elementos da equipa de ITSM do QUASAR na submissão de artigos a duas conferências nacionais a realizar brevemente (INFORUM'2011 e CAPSI'2011);
- Validação da ontologia de ITSM com exemplos mais complexos; nesta dissertação apenas se realizou a prova de conceito com exemplos triviais;
- Integração da representação estática e dinâmica de serviços de gestão de TI; com efeito, embora o metamodelo de BPMN tenha sido extensivamente validado com um exemplo real de grande dimensão, este exemplo não tirou partido da ontologia ITSM definida; por outro lado o referido exemplo era em gestão de projectos de TI e por isso mais genérico que a desejada gestão de serviços de TI e, mais precisamente a gestão de níveis de serviço de TI que se pretende;
- Avaliação da conformidade de SLAs em ambiente real; este é o pano de fundo desta dissertação e, como já foi anteriormente referido, constituirá uma das contribuições esperadas no âmbito da dissertação de doutoramento em curso de Anacleto Correia, a decorrer igualmente no grupo QUASAR; a presente dissertação permitiu ganhar alguma massa crítica que já deu origem a alguns contactos com a indústria e ajudou a alavancar uma submissão a um projecto QREN em que a FCT/UNL é parceira.

Bibliografia

- [Abe, 2005] Tadahiko Abe, "What is Service Science?" 246, Fujitsu Research Institute, Tokyo, December 2005, 2005.
- [Abreu, et al., 2010] Fernando Brito e Abreu, Raquel de Bragança V. da Porciúncula, José Carlos Costa, and Jorge Manuel Freitas, "Definition and Validation of Complexity Metrics for ITSM Process Models", in proceedings of *The International Conference on the Quality of Information and Communications Technology (QUATIC 2010)*, Porto, 2010.
- [Araújo, 2005] Manuel Jorge Brandão Estêvão de Araújo, "A influência dos SLAs na remuneração e outsourcing", Relatório na disciplina de Complementos de Sistemas de Informação, Departamento de Engenharia Informática, Universidade de Coimbra, Coimbra, Portugal, 2005.
- [Arsanjani, 2004] Ali Arsanjani, "Service-oriented modeling and architecture", 09 Nov 2004. Available: <https://www.ibm.com/developerworks/library/ws-soa-design1/>. [Accessed: 2009].
- [Arun Rai and Sambamurthy, 2006] Arun Rai and Vallabh Sambamurthy, "Editorial Notes—The Growth of Interest in Services Management: Opportunities for Information Systems Scholars," *Information Systems Research*, vol. Vol. 17, pp. 327–331, INFORMS, 2006.
- [BEA - Bureau Of Economics Analysis, 2009] U.S. - Department of Commerce BEA - Bureau Of Economics Analysis, "U.S. International Transactions, 1960-present", 15-12-2009. Available: <http://www.bea.gov/international/#trade>. [Accessed: 18-12-2009 2009].
- [Bitner, et al., 2008] Mary Jo Bitner, Stephen W. Brown, Michael Goul, and Susan Urban, "Services Science Journey: Foundations, Progress, Challenges", in *Service Science, Management and Engineering Education for the 21st Century*, B. Hefley and W. Murphy, Eds.: Springer US, 2008, pp. 227-233.
- [Booch, et al., 1998] Grady Booch, James Rumbaugh, and Ivar Jacobson, *The Unified Modeling Language User Guide*: Addison Wesley Longman Inc., 1998, ISBN 0-201-57168-4.
- [BPMNI, 2004] Business Process Management Initiative BPMNI, "Business Process Modeling Notation (BPMN) v1.0": Object Management Group (OMG), 2004.
- [Campos and Gomes, 2008] Maria Luiza de Almeida Campos and Hagar Espanha Gomes, "Taxonomia e Classificação: o princípio de categorização," *DataGramaZero - Revista de Ciência da Informação*, vol. v.9 n.4 ago/08, 2008.
- [Carter-Stell, et al., 2006] Aileen Carter-Stell, Mark Toleman, and Wui-Gee Tan, "Transforming IT Service Management - the ITIL Impact", in proceedings of *17Th Australasian Conference on Information Systems*, Adelaide, 2006.
- [Cavalcanti, et al., 2007] André Marques Cavalcanti, André Marques Cavalcanti Filho, and Telma Lúcia de Andrade Lima, "Modelo de gerenciamento de contratos de outsourcing baseados em SLA e prestação de serviços", in *XXVII Encontro Nacional de Engenharia De Produção U.-U. F. d. Pernambuco*, Ed. Foz do Iguçu, Brasil, 2007.
- [Cibran, 2008] María Agustina Cibran, "Translating BPMN Models into UML Activities", in proceedings of *1st International Workshop on Model-Driven Engineering For Business Process Management 2008*, 2008.

- [Coelho, 2003] Jorge S. Coelho, "Método LEARN – Um contributo para a definição das necessidades de informação de acordo com a estratégia do negócio", in proceedings of *CAPSI 2003*, Porto, 2003, p. 22.
- [Commerce, 2007] Office of Government Commerce, Ed. "The Introduction to the ITIL Service Lifecycle Book ": The Stationery Office, 2007.
- [Correia and Abreu, 2010] Anacleto Correia and Fernando Brito e Abreu, "Model-Driven Service Level Management", in proceedings of *Proceedings of 4th International Conference on Autonomous Infrastructure, Management and Security (AIMS'2010)*, 2010.
- [Costa, 2010] Carlos Gonçalves Costa, "MGPSI – Metodologia de Gestão de Projectos Aplicada ao Desenvolvimento de Sistemas de Informação". *Tese de Mestrado dissertation*, P. D. F. B. e. Abreu (advisor), Departamento de Informática, Lisboa: Universidade Nova de Lisboa/Faculdade de Ciências e Tecnologia, 2010, p. 248.
- [David and Zahavi, 2009] M. Bridgeland David and Ron Zahavi, *Business Modeling: A Practical Guide to Realizing Business Value* vol. 1: Elsevier, 2009, ISBN 978-0-12-374151-6.
- [Drucker, 2003] Peter Drucker, *Sociedade Pós-Capitalista*, 2003, ISBN 9728152469.
- [Fagundes, 2004] Eduardo Mayer Fagundes, "Gestão de Contratos com SLA", 2008. Available: http://www.efagundes.com/Artigos/Gestao_de_Contratos_com_SLA.htm. [Accessed: 2008].
- [Farren, 2005] Margaret Farren, "How can i create a pedagogy of the unique through a web of betweenness?", Education and Training Management, Bath: University of Bath, 2005.
- [Faurer, et al., 2003] Cliff Faurer, Chris Hartley, Helen Hepburn, John Reilly, Wayne Sigley, John Strassner, Johan Lindbladh, and Andrea Westerinen, "Shared Information/Data (SID) Model ", T. F. TMF, 2003.
- [Forum, 2007] TeleManagement Forum, "Enhanced Telecom Operations Map (eTOM) - The Business Process Framework v7.1". gb921, T. Forum: TeleManagement Forum, 2007.
- [Forum, 2006] TeleManagement Forum, "Enhanced Telecommunications Operations Map - eTOM - Estrutura de Processos de Negócio - Versão 6.0": TeleManagement Forum, 2006.
- [Foundation, 2009a] The Eclipse Foundation, "Eclipse Modeling Framework Project (EMF)". Available: <http://www.eclipse.org/modeling/emf/>. [Accessed: 29, October, 2009]
- [Foundation, 2009b] The Eclipse Foundation, "SOA Tools Platform Project". Available: <http://www.eclipse.org/stp/>. [Accessed: 29-10-2009]
- [Foundation, 2009c] The Eclipse Foundation, "The STP BPMN modeler". Available: <http://www.eclipse.org/bpmn/>. [Accessed: 28-10-2009]
- [Freitas, et al., 2008] Jorge Freitas, Anacleto Correia, and Fernando Brito e Abreu, "An Ontology for IT Services", in proceedings of *13th Conference on Software Engineering and Databases (JISBD 2008)*, Gijón (Spain), 2008, p. 12.
- [Garschhammer , et al., 2001a] Markus Garschhammer , R. Hauck, H.-G. Hegering, B. Kempter, M. Langer, M. Nerb, I. Radisic, H. Roelle, and H Schmidt, "Towards generic service management concepts - A service model based approach", in proceedings of *7th International IFIP/IEEE Symposium on Integrated Management*, 2001a.
- [Garschhammer , et al., 2001b] Markus Garschhammer , Rainer Hauck , Bernhard Kempter , Igor Radisic , Harald Roelle , Holger Schmidt , and H. Schmidt, "The MNM Service Model – Refined Views on Generic Service Management," *Journal of Communications and Networks*, pp. 297-306, 2001b.
- [GIL, 1999] Antonio Carlos GIL, *Métodos e técnicas de pesquisa social*. São Paulo: Atlas, 1999.

- [Group, 2008] University of Bremen Database Systems Group, "USE - A UML-based Specification Environment". Available: <http://www.db.informatik.uni-bremen.de/projects/USE>. [Accessed: 2010].
- [Guizzardi, 2007] Guiancarlo Guizzardi, "On Ontology, Ontologies, Conceptualizations, Modeling Languages, and (Meta)Models", in proceedings of *Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV*, Amsterdão, 2007.
- [Harrington, 1991] H. James Harrington, Ed. "Business process improvement", New York: McGraw Hill, 1991.
- [Hiles, 2002] Andrew Hiles, *E-Business Service Level Agreements - Strategies for Service Providers, E-Commerce and Outsourcing: The Rothstein Catalog on Service Level Books*, 2002, ISBN 0-9641648-9-2.
- [Hill, 2005] T. P. Hill, "On goods and Services", in *Review of Income and Wealth*. vol. 23, 2005, pp. 315-338.
- [IBM, 2004] IBM, "Services Sciences, Management and Engineering". Available: <http://www.research.ibm.com/ssme/services.shtml>. [Accessed: 2010].
- [ISO14598] ISO14598, "Software Product Evaluation", in *Information Technology: ISO/IEC*.
- [Leader, 2009] Knowledge Leader, "Service Level Agreement (SLA) - Sample Template". Available: <http://www.knowledgeleader.com>. [Accessed: 20 October 2009]
- [Lee and Ben-Natan, 2002] John J. Lee and Ron Ben-Natan, *Integrating Service Level Agreements - Optimizing Your OSS for SLA Delivery*: Robert Ipsen, 2002.
- [Loyd, 2007] V. Loyd, Ruud, C., *ITIL Service Design*. London: TSO, 2007.
- [Loyd and Rudd, 2007] Vernon Loyd and Colin Rudd, Eds., "ITIL - Service Design", O.-O. o. G. Commerce, Ed., London: TSO - The Stationery Office, 2007.
- [Martins and Silva, 2004] Paula Ventura Martins and Alberto Rodrigues Silva, "Comparação de Metamodelos de Processos de Desenvolvimento de Software", in proceedings of *5ª Conferência para a Qualidade nas Tecnologias da Informação e Comunicações (QUATIC'2004)*, Porto, 2004.
- [Masche, et al., 2006] Philipp Masche, Paul Mckee, and Bryce Mitchell, "The Increasing Role of Service Level Agreements in B2B Systems", in proceedings of *Second International Conference on Web Information Systems and Technologies*, Setúbal, Portugal, 2006, pp. 123-126.
- [Medeiros, 2007] Alexandre David Medeiros, "SLA em redes de telecomunicações aplicados a multi-serviços em redes IP", Centro de ciências exactas, ambientais e de tecnologias, Campinas: Pontifícia Universidade Católica de Campinas, 2007.
- [Mendes and Mira da Silva, 2009] Carlos Mendes and Miguel Mira da Silva, "Controlo de custos operacionais de informática", in proceedings of *9ª Conferência da Associação Portuguesa de Sistemas de Informação (CAPSI)* Escola Superior de Tecnologia e Gestão de Viseu - Instituto Politécnico de Viseu, 2009.
- [Meyer, 1985] Bertrand Meyer, "On formalism in specifications," *IEEE Software*, vol. 01/85, p. 21, 01-01-1985, 1985.
- [Michael Hammer and Champy, 1994] Michael Hammer and James Champy, Eds., "Reengineering the corporation", HarperBusiness, Ed., New York, 1994.
- [Moura, 1995] Hermano Perrelli de Moura, "O Uso de Especificações Formais na Construção de Software," *Jornal de Brasília*, 1995.
- [Nawrocki, et al., 2006] Jerzy Nawrocki, Tomasz Nędza, Mirosław Ochodek, and Łukasz Olek, "Describing Business Processes with Use Cases", in proceedings of *BIS 2006, Business Information Systems*, Poznań, Poland, 2006.
- [Niessink and Vliet, 2000] Frank Niessink and Hans van Vliet, "Software maintenance from a service perspective," *Journal of Software Maintenance: Research and Practice* vol. 12, pp. 103-120, John Wiley & Sons, Inc, 2000.

- [OGC, 2003] OGC, *IT Infrastructure Library - Service Delivery*. London: Stationery Office, 2003.
- [OMG, 2008] OMG, "Business Process Modeling Notation (BPMN) Version 1.1". formal/2008-01-17: OMG, 2008.
- [OMG, 2009a] OMG, "Business Process Modeling Notation (BPMN) Version 1.2". vol. OMG Document Number: formal/2009-01-03: OMG, 2009a.
- [OMG, 2009b] OMG, "Business Process Modeling Notation (BPMN) Version 1.2". formal/2009-01-03: OMG, 2009b.
- [OMG, 2009c] OMG, "Business Process Modeling Notation (BPMN) Version 2.0 Beta". OMG Document Number: dtc/2009-08-14: OMG, 2009c.
- [OMG, 2002] OMG, "Meta Object Facility (MOF) Specification Version 1.4": Object Management Group, 2002.
- [OMG, 2003] OMG, "UML 2.0 OCL Final Adopted specification" ptc/03-10-14, Object Management Group Inc., October, 2003.
- [OMG, 2006] OMG, "UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms - OMG Available Specification Version 1.0": Object Management Group Inc., 2006.
- [Penker and Eriksson, 2000] Magnus Penker and Hans-Erik Eriksson, "Business Modeling With UML: Business Patterns at Work (Paperback)": Jonh Wiley and Sons, 2000.
- [Petit, 2004] M. Petit, "A Service Level Agreement For Information Technology Services Between the Dean, Faculty of Arts And the Director, ITS", 2004.
- [Philippe, et al., 2006] Jeremy Philippe, Noel De Palma, Sara Bouchenak, Fabienne Boyer, and Daniel Hagimont, "A BlackBox Approach for Web Application SLA", in proceedings of *SAC'06*, Dijon, France, 2006.
- [Porciúncula, 2010] Raquel Helena de Bragança Vasconcelos da Porciúncula, "Governança e Modelação de Processos de TI". *Tese de Mestrado dissertation*, P. D. F. B. e. Abreu (advisor), Departamento de Informática, LISboa: Universidade Nova de Lisboa/Faculdade de Ciências e Tecnologia, 2010, p. 234.
- [Research, 2005] IBM Research, "Services Science: A New Academic Discipline?", IBM, 2005.
- [Ricardo A. Falbo, 2002] Giancarlo Guizzardi Ricardo A. Falbo, Katia C. Duarte, Ana Candida C. Natali, "Developing Software for and with Reuse: An Ontological Approach", in proceedings of *ACIS International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications (CSITeA-02)*, Foz do Iguacu, Brazi, 2002.
- [Rumbaugh, et al., 1991] James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorensen, *Object-Oriented Modelling and Design*. Englewood Cliffs, NJ, EUA: Prentice Hall, 1991.
- [Saeki and Kaiya, 2006] Motoshi Saeki and Haruhiko Kaiya, "On Relationships among Models, Meta Models and Ontologies", in proceedings of *DSM'06@OOPSLA2006 - 6th OOPSLA Workshop on Domain-Specific Modeling*, Portland, Oregon USA, 2006.
- [Silva and Menezes, 2001] Edna Lúcia da Silva and Estera Muszkat Menezes, "Metodologia da Pesquisa e Elaboração de Dissertação", 3ª edição ed Florianópolis: UFSC/PPGEP/LED - Universidade Federal de Santa Catarina / Programa de Pós-Graduação em Engenharia de Produção / Laboratório de Ensino a Distância, 2001.
- [Skene, et al., 2004] J. Skene, D. Lamanna, and W. Emmerich, "Precise Service Level Agreements", in proceedings of *26th International Conference on Software Engineering (ICSE'04)*, 2004.
- [Software, 2008] E-commerce Serviços de Software, "Contrato de Serviço (SLA)", Caso de estudo, 20/04/2008, 2008.

- [Stephen L. Vargo and Lusch, 2004] Stephen L. Vargo and Robert F. Lusch, "Evolving to a New Dominant Logic for Marketing", in *Journal of Marketing*. vol. 68: American Marketing Association, 2004, pp. 1-17.
- [Su, et al., 2007] Xiaomeng Su, Damiano Bolzoni, and Pascal Van Eck, "Understanding and Specifying Information Security Needs to Support the Delivery of High Quality Security Services", in proceedings of *International Conference on Emerging Security Information, Systems and Technologies (SECUREWARE'2007)*, 2007, pp. 107-114.
- [Teyssié, 2006] Cédric Teyssié, "UML-based Specification of QoS Contract Negotiation and Service Level Agreements", in proceedings of *ICNICONSMCL International Conference on Systems and International Conference on Mobile Communications and Learning Technologies*, 2006.
- [Wetzstein, et al., 2008] Branimir Wetzstein, Dimka Karastoyanova, and Frank Leymann, "Towards Management of SLA-Aware Business Processes Based on Key Performance Indicators", in proceedings of *9th Workshop on Business Process Modeling, Development, and Support (BPMDS'08 @ CAiSE'08)*, Montpellier, France, 2008.
- [Wikipedia, 2007] Wikipedia, "Adam Smith", 2007. Available: http://pt.wikipedia.org/wiki/Adam_Smith. [Accessed: 2010].
- [Wikipedia, 2009a] Wikipedia, "Information society". Available: http://en.wikipedia.org/wiki/Information_society. [Accessed: August 15th 2009].
- [Wikipedia, 2009b] Wikipedia, "Metamodeling". Available: <http://en.wikipedia.org/wiki/Metamodeling>. [Accessed: 2009].
- [Wikipedia, 2009c] Wikipedia, "Semantic Web", 2010. Available: http://en.wikipedia.org/wiki/Semantic_Web. [Accessed: 2010].
- [Wikipedia, 2006] Wikipedia, "Taylorismo". Available: <http://pt.wikipedia.org/wiki/Taylorismo>. [Accessed: 2010].
- [Wikipédia, 2010] Wikipédia, "Epistemologia". Available: <http://pt.wikipedia.org/wiki/Epistemologia>. [Accessed: 2010].
- [Wustenhoff, 2002] Edward Wustenhoff, "Service Level Agreement in the Data Center", in *Sun BluePrints™ OnLine* S. P. Services, Ed.: Sun Microsystems, Inc., 2002.

[Esta página foi intencionalmente deixada em branco]

Anexo A

A : Template de SLA

Este anexo reúne breve explicação dos itens que podem constar de um contracto de SLA.

Capa	Service Level Agreement (SLA) Subtítulo com Domínio de Aplicação Versão X.0 (inserir data) (Especificar o uso apropriado e lista de distribuição do documento.)					
Histórico de revisão do documento	Data		Autor		Revisão	Descrição
Aprovações	Data	Nome	Assinatura			
Índices	Desta secção constam os índices remissivos, de figuras, etc.					
1 Âmbito	Informações que descrevem o âmbito de aplicação do documento, ou seja o seu domínio, por exemplo um serviço específico ou departamento,					
1.1 Audiência	Identifica qual o público adequado para o documento.					
1.2 Objectivo	Informações detalhadas que descrevem a finalidade do documento.					
1.3 Pressupostos	Pressupostos associados com o documento.					
1.4 Contactos	Informações descrevendo os contactos relacionados com o documento, tais como departamentos, fornecedores, clientes, etc., sem esquecer os nomes dos gestores principais por parte do cliente e do fornecedor.					
2. Serviço	Esta secção introduz o serviço descrito por este SLA.					
2.1 Detalhes do Serviço	Esta secção contém explicações que sejam importantes para o SLA sobre o serviço que é coberto pelo SLA.					
2.2 Ambiente do Serviço	Esta secção descreve a arquitectura e tecnologias que são utilizadas pelos consumidores do serviço, assim como os seus limites, por exemplo número de utilizadores.					
3. Disponibilidade de serviços no	(Ver pontos seguintes)					

âmbito do SLA							
3.1 Disponibilidade normal de serviço	<p>Cronograma de disponibilidade normal. Exemplo:</p> <p>24 Horas por dia, 365 dia por ano, com as seguintes excepções:</p> <table border="1"> <tr> <td>Disponibilidade do Cliente</td> <td>Domingos a partir das 14h</td> </tr> <tr> <td>Manutenção</td> <td>Domingos a partir das 14h</td> </tr> <tr> <td>Monitorização automática dos serviços</td> <td>24 horas por dia, 365 dia por ano</td> </tr> </table>	Disponibilidade do Cliente	Domingos a partir das 14h	Manutenção	Domingos a partir das 14h	Monitorização automática dos serviços	24 horas por dia, 365 dia por ano
Disponibilidade do Cliente	Domingos a partir das 14h						
Manutenção	Domingos a partir das 14h						
Monitorização automática dos serviços	24 horas por dia, 365 dia por ano						
3.2 Acontecimentos agendados com impacto na disponibilidade	Esta secção descreve os cortes de serviço que são programados.						
3.3 Processo de mudança arquitectura tecnológica	Esta secção descreve o processo completo de como se efectuarmos mudanças, por exemplo novo software ou novos utilizadores, estas são introduzidas no serviço, incluindo o impacto na disponibilidade.						
3.4 Serviços de valor acrescentado a serem fornecidos no âmbito do SLA	<p>Exemplo</p> <table border="1"> <thead> <tr> <th>Referência</th> <th>Descrição do Serviço.</th> </tr> </thead> <tbody> <tr> <td>Req_01</td> <td> <p>O serviço oferecido deve permitir os seguintes registos:</p> <ul style="list-style-type: none"> - Registo de utilizadores; - Histórico de compras; - Gestão de Stocks. </td> </tr> <tr> <td>Req_02</td> <td>O serviço oferecido deve permitir que utilizadores efectuem compras on-line, com suporte ao carrinho de compras e submissão ou cancelamento da compra.</td> </tr> </tbody> </table>	Referência	Descrição do Serviço.	Req_01	<p>O serviço oferecido deve permitir os seguintes registos:</p> <ul style="list-style-type: none"> - Registo de utilizadores; - Histórico de compras; - Gestão de Stocks. 	Req_02	O serviço oferecido deve permitir que utilizadores efectuem compras on-line, com suporte ao carrinho de compras e submissão ou cancelamento da compra.
Referência	Descrição do Serviço.						
Req_01	<p>O serviço oferecido deve permitir os seguintes registos:</p> <ul style="list-style-type: none"> - Registo de utilizadores; - Histórico de compras; - Gestão de Stocks. 						
Req_02	O serviço oferecido deve permitir que utilizadores efectuem compras on-line, com suporte ao carrinho de compras e submissão ou cancelamento da compra.						
3.5 Serviços fora do âmbito deste SLA	Esta secção contém as exclusões explícitas ao SLA.						
3.6 Níveis de	Aos recursos do cliente em conjugação com as prioridades são atribuídos tempos de resposta e escalonados em níveis que podem ter impacto na remuneração ou						

Serviço	penalizações.	
4. Qualidade do Serviço	Esta secção contém uma descrição detalhada de como a disponibilidade de serviço é medida e reportada.	
4.1 SLA KPI (Key Performance Indicators)	Compromisso (Exemplo)	KPI
	O serviço <i>inserir nome</i> , estará disponível X percentagem do tempo de Y horas normais de operação.	Disponibilidade de Serviço.
	X percentagens das operações do serviço K que serão exibidas no máximo em T segundos, sendo este o intervalo entre o enviar uma transacção até receber a informação.	Transacções que demorem mais de T segundos.
	A equipa de suporte técnico do serviço D irá responder a incidentes que afectam os utilizadores no período de tempo de H horas e resolver o problema no máximo em R Horas.	Incidente do serviço D (que afecta utilizadores) tempo de resposta e de resolução.
	A medição da disponibilidade externa.	Relatório Periódico.
4.2 Relatórios		
4.3. Análise da Performance	Envolve a revisão de prioridades, níveis de serviço e o desempenho do sistema. Análise de problemas específicos / Excepções e Prioridades. Tempo de resposta para resolução de problemas. Revisão do SLA e acções correctivas para superar as deficiências encontradas. (Manutenção do contracto de SLA).	
5. Responsabilidades	Informações sobre as responsabilidades de clientes e fornecedores.	
5.1 Obrigações do Cliente	Esta secção pode estar relacionada com o serviço a ser fornecido no âmbito do SLA, detalhando ou atribuindo ao Cliente a sua parte de obrigações para o serviço ser realizado. Exemplos: <ul style="list-style-type: none"> • Programar políticas, processos e procedimentos. • Registo de incidentes e / ou solicitar priorização e / ou em cooperação com o prestador de serviços; • Pagamento de todos os serviços relacionados com a instalação e / ou de configuração relativas à prestação do serviço; • Revisão todas as horas autenticadas pelo fornecedor de serviços para controlo; 	
5.2 Condições de execução do	Exemplos: energia, espaço físico, ar condicionado, etc.	

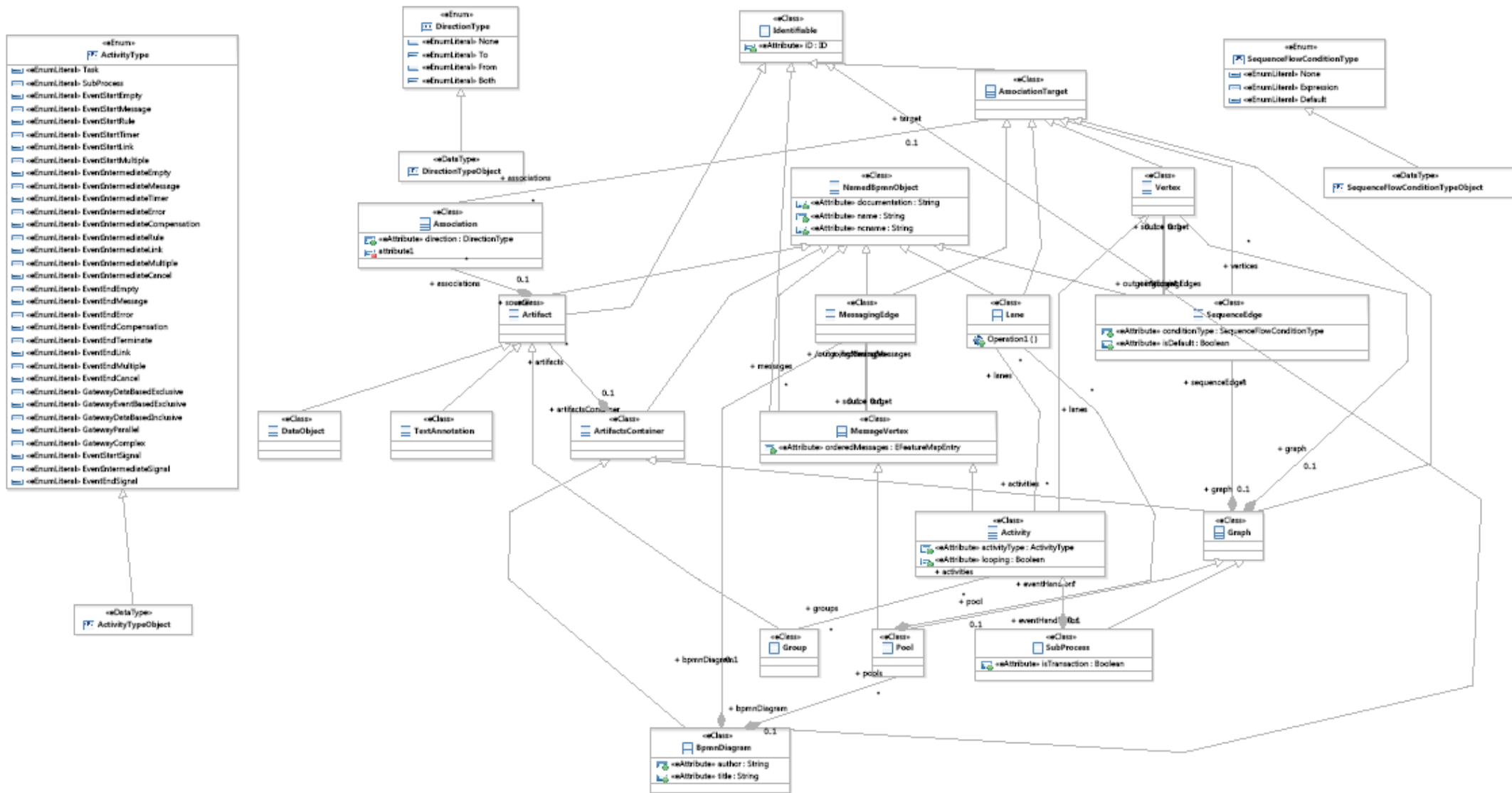
Serviço	
5.3 Acessos do Fornecedor	Sistema de controlo de acesso, autorizações.
5.5 Obrigações do Fornecedor	Esta secção pode estar relacionada com o serviço a ser fornecido no âmbito do SLA, detalhado ou atribuindo ao fornecedor a sua parte de obrigações para o serviço ser realizado. Pode também ser utilizado para especificar outras obrigações do fornecedor tais como questões ambientais, sigilo, seguros, etc.
6. Serviços de <i>Helpdesk</i>	É o serviço de apoio a utilizadores para suporte e resolução de problemas. Este apoio pode ser tanto dentro de uma empresa (profissionais que cuidam da manutenção de equipamentos e instalações dentro da empresa), quanto externamente (prestação de serviços a utilizadores).
6.1 Registo de Problemas	Controlo dos incidentes e respostas.
6.2 Prazos de resposta	Ligação aos serviços operacionais de TI.
6.3 Gestão de Alterações	Gestão de versões e de configurações.
6.4. Horário do Serviço de resolução dos problemas	Informações com a descrição do serviço para resolver problemas.
7 Resolução de Conflitos	Esta secção descreve processo a ser utilizado no caso de um litígio entre o prestador de serviços e o cliente quanto à aplicação do SLA para a resolução de um problema.
7.1 Escalada das Acções	Inicialmente, o operacional contacta o suporte técnico. O suporte técnico atribui uma prioridade e resolve dentro do previsto. Se não se resolver reúnem-se ambas as partes para tentar resolver o diferendo. Se uma solução não é atingida, então o problema será escalado até aos contactos principais.
7.3 Prioridades	São referenciados quais os recursos que existem no ambiente do cliente, a fim de se estabelecerem níveis de prioridades dentro do sistema. Este controle será possível pela implementação de mecanismos de logística a serem implementados ao sistema, o que pode ser feito por meio de políticas de controlos automatizados. [Medeiros, 2007]
7.4 Exclusões	Incidentes não cobertos pelo contracto de SLA.
8. Tempo de resposta para resolução de	O SLM (<i>Service Level Management</i>) verifica se os tempos de resposta para o registo e resolução de incidentes e problemas são os que estão definidos nos SLA. Podem ser definidos vários tempos de resposta, em função das prioridades e dos

problemas	níveis de serviços que se pretende alcançar.	
9. Cessação do Acordo	A denúncia do presente acordo pode ser iniciada por qualquer das partes em caso desacordos.	
10. Período do Acordo	Período de tempo em que o acordo é estabelecido, bem como forma de renovação ou cessação.	
11. Pagamento dos serviços	Valor, forma e data de pagamento.	
12. Penalidades ou Incentivos	Definição de penalizações por quebras no SLA, ou mesmo recompensas/incentivos no caso do nível de qualidade acordado ser excedido.	
	Número de Violações (Exemplo)	Penalidade (Exemplo)
	1 - 5	Inserir penalidade. Geralmente uma redução dos pagamentos.
	5 - 10	Inserir penalidade. Geralmente uma redução das pagamentos mais alguns adicionais de compensação e um plano de acção correctiva.
	➤ 10	Inserir pena. Geralmente uma redução da taxa mais alguns adicionais de compensação e um plano de acção correctiva
13. Manutenção do SLA	“Depois da sua assinatura e entrada em acção, tem de ser feito um acompanhamento/monitorização do acordo, de forma a evitar incumprimentos ou mesmo para garantir que este ainda é necessário – se a estratégia da empresa mudar, o SLA pode já não ser interessante e ser alvo de renegociações, ou mesmo descartado. Mais, a tecnologia avança rapidamente e é preciso que os termos do SLA estejam actualizados no que diz respeito aos standards da indústria. “[Araújo, 2005]	

Anexo B

B : Metamodelo STP_Eclipse

Este anexo contém o metamodelo BPMN utilizado no STP Eclipse. (De notar que é toda a documentação encontrada referente a este metamodelo).



Anexo C

C: Extracto para exemplo das especificações OMG BPMN 1.2

Este anexo contém uma página para efeitos de exemplo das especificações OMG BPMN 1.2, que são constituídas por 316 páginas das quais constam exemplos, especificações escritas, tabelas e diagramas (só com as especializações).

B.8 Swimlanes (Pools and Lanes)

The following figure displays a diagram of the relationship between BPMN Swimlane elements and their attributes (see Figure B.5).

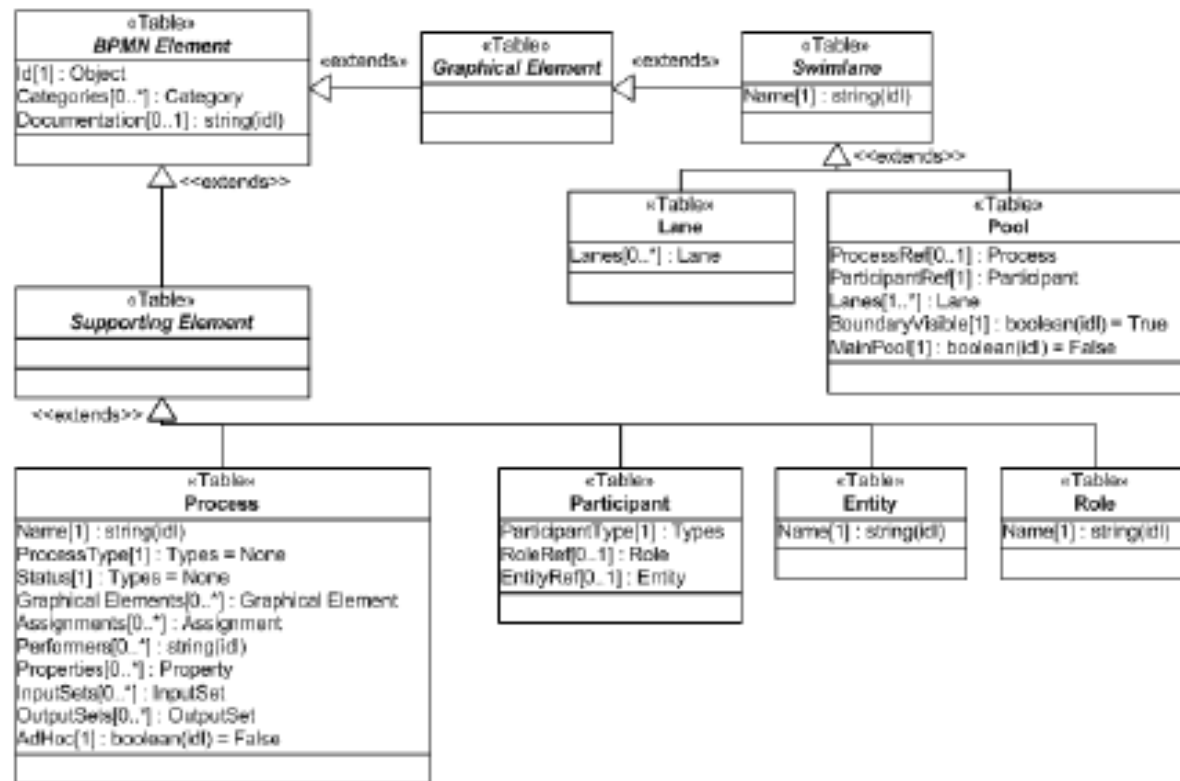


Figure B.5 - BPMN Swimlane Elements and Attributes

B.8.1 Common Swimlane Attributes

The following table displays a set of common attributes for Swimlanes (Pools and Lanes), and which extends the set of common BPMN Element attributes (see Table B.2).

Table B.28 - Common Swimlane Attributes

Attributes	Description
Name : String	Name is an attribute that is text description of the Swimlane.

Anexo D

D: Métricas da Gestão de Incidentes e Gestão de Problemas

Este anexo contém uma relação dos meta-objects inseridos no metamodelo BPMN no âmbito da dissertação “Governança e Modelação de Processos de TI” [Porciúncula, 2010].

	Gestão de Incidentes		Gestão de Problemas	
	<i>As-Is</i>	<i>To-Be</i>	<i>As-Is</i>	<i>To-Be</i>
Nº de Processos	1	1	1	1
Nº de Caminho Possíveis (Start1, End1)	3	10	1	4
Nº de Objectos diferentes em todos os caminhos possíveis	27	44	14	31
Nº de Objectos nos vários caminhos (Start1, End1)	{15,20,21}	{16,16,20,20,24,24,29,29,30,30}	[ISO14598]	{25,25,26,26}
Nº de Objectos no caminho maior (Start1, End1)	21	30	14	26
Nº de Objectos no caminho menor (Start1, End1)	15	16	14	25
Nº de Pools	1	1	1	1
Nº de Lanes	4	3	4	3
Nº de Swimlanes	5	4	5	4
Nº de FlowObjects	17	23	13	19
Nº de Activities	12	15	10	13
Nº de SubProcesses	0	0	0	0
Nº de Tasks	12	15	10	13
Nº de Events	2	2	2	2
Nº de Start Events	1	1	1	1
Nº de End Events	1	1	1	1
Nº de Intermediate Events	0	0	0	0
Nº de Gateways	3	6	1	4
Nº de GatewayDataBasedExclusives	3	6	1	4
Nº de GatewayEventBasedExclusives	0	0	0	0
Nº de GatewayDataBasedInclusives	0	0	0	0
Nº de ComplexGateways	0	0	0	0

N° de ParallelGateways	0	0	0	0
N° de Gates	10	21	3	12
N° de InputGates	5	10	1	6
N° de OutputGates	5	11	2	6
N° de Artifacts	0	0	0	0
N° de DataObjects	0	0	0	0
N° de TextAnnotations	0	0	0	0
N° de Groups	0	0	0	0
N° de Connectors	18	27	13	20
N° de SequenceFlows	18	27	12	20
N° de MessageFlows	0	0	1	0
N° de Associations	0	0	0	0
Métrica CFC	4	10	2	4
Métrica HPC - length	4,8	4,8	2	4,8
Métrica HPC - volume	23,8	33,4	11	27,0
Métrica HPC - difficulty	1,5	1,5	1	1,5
Métrica CNC	1,1	1,2	0,9	1,1
Métrica -NestingDepth (Start1, End1)	3	6	1	4
Métrica - BPM	8	17	2	10
Métrica HKM	12	15	10	13

[Esta página foi intencionalmente deixada em branco]

Anexo E

E: Processo “Monitorização de Incidentes no SLA” modelado com BPMN

Este anexo contém um exemplo de processo modelado com BPMN designado “Monitorização de Incidentes no SLA”.

[Esta página foi intencionalmente deixada em branco]

