

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

ANDRÉ CYPRIANO MONTEIRO COSTA

**MODELAGEM DO DOMÍNIO DO
PROCESSO DE GERENCIAMENTO DE
NÍVEL DE SERVIÇO DO PADRÃO ITIL**

**UMA ABORDAGEM USANDO ONTOLOGIAS DE FUNDAMENTAÇÃO
E SUA APLICAÇÃO NA PLATAFORMA INFRAWARE**

**VITÓRIA
2008**

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

ANDRÉ CYPRIANO MONTEIRO COSTA

**MODELAGEM DO DOMÍNIO DO
PROCESSO DE GERENCIAMENTO DE
NÍVEL DE SERVIÇO DO PADRÃO ITIL**

**UMA ABORDAGEM USANDO ONTOLOGIAS DE FUNDAMENTAÇÃO
E SUA APLICAÇÃO NA PLATAFORMA INFRAWARE**

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para obtenção do grau de Mestre em Informática.

Orientador: Prof. Dr. José Gonçalves Pereira Filho

**VITÓRIA
2008**

**MODELAGEM DO DO PROCESSO DE
GERENCIAMENTO DE NÍVEL DE SERVIÇO DO
PADRÃO ITIL
UMA ABORDAGEM USANDO ONTOLOGIAS DE FUNDAMENTAÇÃO E SUA
APLICAÇÃO NA PLATAFORMA INFRAWARE**

André Cypriano Monteiro Costa

Dissertação submetida ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo como requisito parcial para obtenção do grau de Mestre em Informática.

Prof. José Gonçalves Pereira Filho (D.Sc.) - DI/UFES
Orientador

Prof^a. Patricia Dockhorn Costa (D.Sc.) - DI/UFES

Prof. Luiz Fernando Gomes Soares (D.Sc.)
DI/PUC – Rio de Janeiro

UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
Vitória, Agosto de 2008.

Aos meus pais Carlos Alberto (*in memoriam*) e Maria Angélica
A meus irmãos Arthur e Estevão

AGRADECIMENTOS

Inicialmente eu gostaria de agradecer imensamente ao meu orientador e amigo, o Prof. José Gonçalves Pereira Filho que me proporcionou os primeiros contatos com essas tecnologias e áreas de pesquisas desde a graduação e que motivaram este estudo agora no mestrado. José Gonçalves muito obrigado por me motivar, por me apoiar nas minhas dificuldades e tomadas de decisão, por auxiliar nas minhas pesquisas, no meu desenvolvimento, e principalmente na reta final deste trabalho pelos seus comentários e importantes observações. Valeu!

Gostaria muito também de agradecer ao Prof. Giancarlo Guizzardi pelas conversas que tivemos e por me ajudar muito a compreender melhor essas tal ontologias. Gian foi uma pessoa que sempre estive disposta a me ajudar (apesar da agenda lotada!) e foi de fundamental importância para a realização deste trabalho. Muito obrigado Gian!

Não poderia deixar de agradecer àquela que com muito carinho e boa vontade (características marcantes dela) foi importantíssima para me fazer reencontrar os caminhos dentro da lógica. Prof^a. Rosane Caruso é de você que estou falando. Muito obrigado pelo apoio, pela preocupação, e por acreditar que este trabalho sairia a tempo!

Meus agradecimentos também à Fundação de Apoio à Ciência e Tecnologia do Espírito Santo (FAPES) por ter financiado este meu trabalho e assim, proporcionado a realização de pesquisas que resultaram neste trabalho.

Agora é a vez de agradecer à galera do LPRM, meus companheiros diários (manhã, tarde e noite) de pesquisas e trabalhos no laboratório. Bernardo, Carlos, Diego, Luiz Rodrigo, Ramon, Wilian e tantos outros com quem convivi ao longo de vários anos. Muito obrigado pelo convívio e amizade, pelo apoio de vocês, pelas trocas de experiências e pelos momentos de aprendizagens que tivemos juntos.

Dois agradecimentos muito especiais. Primeiro ao meu amigo Camilo Calvi pelas discussões sobre ITIL e gerenciamento de serviços. À minha mais nova amiga Veruska Zamborlini pela imensa ajuda que deu na implementação deste trabalho, principalmente nas construções das regras e na execução das inferências. Nunca esquecerei a boa vontade de vocês, ao doarem

seu tempo para tirar minhas dúvidas e me auxiliar com os primeiros passos. Agradeço imensamente por tudo que fizeram.

Não posso me esquecer dos meus amigos de infância, escola, graduação, espíritas, de trekking, etc. Obrigado pela paciência de vocês, por compreender as minhas ausências, pelo carinho que têm por mim, e por me incentivarem e torcerem pela minha vitória.

Ao meu pai Carlos Alberto (in memorian) e à minha mãe Maria Angélica por investirem em minha educação me proporcionando tornar a pessoa que hoje sou. Aos meus irmãos Arthur e Estevão pela amizade, incentivo e ajuda. Enfim, um agradecimento mais que especial à minha Família, por todo incentivo, paciência e carinho transmitidos durante todo esse trabalho. Amo muito TODOS vocês!

Enfim, um agradecimento mais do que especial à minha namorada e companheira, que sempre esteve ao meu lado, que compartilhou comigo de todas as alegrias e sofrimentos, e que foi um dos meus pilares nesses anos de pesquisa. Saiba que eu não tenho palavras para agradecer por toda a paciência (e que paciência!) que você teve para suportar minhas ausências. Muito obrigado também por todo amor, carinho e alegrias que você me deu durante o caminho. À minha querida Juliana o meu eterno obrigado!

Por fim, agradeço a Deus e às entidades espirituais por todo o auxílio, intuição e iluminação dada desde o início deste trabalho, tanto nos momentos mais difíceis quanto nos mais felizes.

RESUMO

Este trabalho apresenta um modelo conceitual do domínio do processo de Gerenciamento de Nível de Serviço da biblioteca ITIL utilizando ontologias de fundamentação, e a aplicação dos conceitos definidos neste modelo no módulo provedor de serviço da plataforma de serviços sensíveis ao contexto Infraware. O estudo do processo de gerenciamento de serviços ITIL e a modelagem conceitual deste domínio visam dar os primeiros passos no que se refere à formalização dos conceitos deste domínio e a sua utilização em plataformas de serviços sensíveis ao contexto, em particular. Um estudo de caso é desenvolvido como forma de validar o modelo e de comprovar sua aplicabilidade na geração de informações de gerenciamento e de controle para o módulo provedor de serviço da plataforma e para seus clientes.

Palavras-Chave: Gerenciamento de Serviços de TI; ITIL; Ontologias; Modelagem Conceitual; Computação Sensível ao Contexto.

ABSTRACT

This work presents a domain conceptual model of ITIL Service Level Management by using foundational ontologies, and also presents the application of the defined concepts in this model in the service provider module of Infraware context-aware service platform. The study of the ITIL service management process and conceptual modeling of this domain is an attempt to give the first steps in what refers to the formalization of this domain concepts and its use in context-aware service platforms, in particular. A case study is presented to validate the model and to verify its applicability in the generation of management information and of control to the service provider module of the platform and its customers.

Keywords: IT Service Management, ITIL, Ontologies, Conceptual Modeling, Context-aware Computing.

Lista de Figuras

Figura 2-1 - Evolução da função da TI em uma organização [Sallé, 2004]	17
Figura 2-2 - Modelo simples de melhoria de processos [Bon et al, 2006]	20
Figura 2-3 - Framework de publicações do ITIL [Macfarlane & Rudd, 2005].....	22
Figura 2-4 - Processos de suporte a serviço do ITIL [Macfarlane & Rudd, 2005]	23
Figura 2-5 - Processos de entrega de serviço do ITIL [Macfarlane & Rudd, 2005] ...	26
Figura 2-6 - Processo de Gerenciamento de Nível de Serviço [Bon et al, 2006].....	31
Figura 2-7 - Fase de Especificação [Bon et al, 2006].....	33
Figura 3-1 - Abordagem do SABiO para o desenvolvimento de ontologias [Falbo:1998]	43
Figura 3-2 - Triângulo de Ullmann: relacionamento entre uma linguagem, uma conceituação e uma porção da realidade.....	47
Figura 3-3 - Relacionamento entre conceituação, abstração, linguagem de modelagem e especificação [Guizzardi, 2005]	48
Figura 3-4 - Conseqüência de uma Linguagem de Modelagem imprecisa para a representação da conceituação de um domínio [Guizzardi, 2005].....	48
Figura 3-5 - Relacionamento entre ontologia de domínio e modelos de implementação [Guizzardi, 2005]	51
Figura 3-6 - Um fragmento da UFO-A [Guizzardi, 2008]	53
Figura 3-7 - Um fragmento da UFO-B [Guizzardi et al, 2008]	60
Figura 3-8 - Um fragmento da UFO-C [Guizzardi et al, 2008]	61
Figura 3-9 - Distinção entre as relações de dependência, delegação e aquisição [Guizzardi & Guizzardi, 2008].....	64
Figura 3-10 - Desenvolvimento e implementação do domínio de Gerenciamento de Nível de Serviço usando ontologias	66
Figura 4-1 - Base da ontologia do domínio de Gerenciamento de Nível de Serviço	68
Figura 4-2 - Fragmento principal da ontologia de Gerenciamento de Nível de Serviço	70
Figura 4-3- Fragmento de requisição de serviço da ontologia de Gerenciamento de Nível de Serviço	76
Figura 4-4 - Exemplo de powertype	78
Figura 4-5 - Fragmento de modelagem de powertype da ontologia	79

Figura 4-6 - Fragmento de nível de serviço da ontologia de Gerenciamento de Nível de Serviço	81
Figura 5-1 - Exemplo de aplicação sensível ao contexto [Calvi, 2007].....	86
Figura 5-2 - Plataforma Infraware.....	87
Figura 5-3 - Camadas funcionais da plataforma Infraware [Calvi, 2007].....	89
Figura 5-4 - Módulo de Serviço da Infraware	95
Figura 6-1 - Modelo do Gerenciamento de Serviço em OWL construído no Protégé	105
Figura 6-2 - Interação entre os atores do sistema TeleCardio [Andreão et al, 2006]	106
Figura 6-3 - Interface da Aplicação TeleCardio	108
Figura 6-4 - Serviços que podem ser requisitados por um cliente.....	113
Figura 6-5 - Inferência do HITSP, do SLA existente, da ITSE e ITSD de um serviço requisitado.....	114
Figura 6-6 - Inferência dos IITSPs e EITSPs , dos OLAs e UCs e ITDSs associadas	115

Lista de Tabelas

Tabela 6.1 - Modelos de Implementação em OWL.....	103
---	-----

LISTA DE ACRÔNIMOS

CCTA	<i>Central Computer and Telecommunications Agency</i>
CobiT	<i>Control Objectives for Information and related Technology</i>
DAML	<i>DARPA Agent Markup Language</i>
DI	Departamento de Informática
HP	Hewlett-Packard
IA	Inteligência Artificial
IC	Item de Configuração
ITIL	<i>Information Technology Infrastructure Library</i>
ITIM	<i>IT Infrastructure Management</i>
ITSM	<i>IT Service Management</i>
itSMF	<i>IT Service Management Forum</i>
MOF	<i>Microsoft Operations Framework</i>
LPRM	Laboratório de Pesquisa em Redes e Multimídia
OGC	<i>Office of Government Commerce</i>
OIL	<i>Ontology Inference Layer</i>
OLA	<i>Operational Level Agreements</i>
OWL	<i>Web Ontology Language</i>
PDA	<i>Personal Digital Assistant</i>
PEP	Prontuário Eletrônico do Paciente
RDF	<i>Resource Description Framework</i>
RDFS	<i>Resource Description Framework Schema</i>
SABiO	<i>Systematic Approach for Building Ontologies</i>
SAMU	Serviço de Atendimento Móvel de Urgência
SLA	<i>Service Level Agreements</i>
SLM	<i>Service Level Management</i>
SLR	<i>Service Level Requirement</i>
SQP	<i>Service Quality Plan</i>
SWRL	<i>Semantic Web Rule Language</i>
TI	Tecnologia da Informação
TIC	Tecnologia de Informação e Comunicação
UC	<i>Underpinning Contracts</i>

UFES	Universidade Federal do Espírito Santo
UFO	<i>Unified Foundational Ontology</i>
UML	<i>Unified Modeling Language</i>
W3C	World Wide Web Consortium
WASP	<i>Web Architectures for Service Platforms</i>

Sumário

1. INTRODUÇÃO.....	10
1.1. MOTIVAÇÃO	10
1.2. ESCOPO DO TRABALHO.....	12
1.2.1. GERENCIAMENTO DE NÍVEL DE SERVIÇO	12
1.2.2. SERVIÇOS SENSÍVEIS AO CONTEXTO	12
1.3. OBJETIVOS	14
1.4. ESTRUTURA DO TRABALHO	15
2. GERENCIAMENTO DE SERVIÇOS E GOVERNANÇA DE TI.....	17
2.1. GERENCIAMENTO E GOVERNANÇA	17
2.1.1. GERENCIAMENTO DE SERVIÇOS DE TI	18
2.1.2. ITSM ORIENTADOS A PROCESSOS	19
2.2. PADRÕES DE GERENCIAMENTO DE TI.....	21
2.2.1. O PADRÃO ITIL.....	21
2.3. DESCRIÇÃO DO DOMÍNIO DE GERENCIAMENTO DE NÍVEL DE SERVIÇO	27
2.3.1. CONCEITOS BÁSICOS	28
2.3.2. OBJETIVOS E BENEFÍCIOS DO PROCESSO DE GERENCIAMENTO DE NÍVEL DE SERVIÇO	29
2.3.3. O PROCESSO DO SLM.....	30
2.3.4. RELACIONAMENTO COM OUTROS PROCESSOS ITIL.....	35
2.3.5. CONTROLE DO PROCESSO	36
2.4. CONSIDERAÇÕES FINAIS DO CAPÍTULO	37
3. ONTOLOGIA E MODELAGEM CONCEITUAL	39
3.1. ONTOLOGIA: DEFINIÇÃO E BREVE HISTÓRICO	39
3.2. METODOLOGIAS DE DESENVOLVIMENTO DE ONTOLOGIAS	42
3.2.1. IDENTIFICAÇÃO DE PROPÓSITO E ESPECIFICAÇÃO DE REQUISITOS.....	43
3.2.2. CAPTURA DA ONTOLOGIA	44
3.2.3. FORMALIZAÇÃO DA ONTOLOGIA.....	44
3.2.4. INTEGRAÇÃO COM ONTOLOGIAS EXISTENTES.....	45
3.2.5. AVALIAÇÃO E DOCUMENTAÇÃO DA ONTOLOGIA	45
3.3. MODELAGEM CONCEITUAL E LINGUAGEM	45
3.4. MODELOS DE ESPECIFICAÇÃO.....	49
3.5. ONTOLOGIAS DE FUNDAMENTAÇÃO	51
3.5.1. UFO-A: UMA ONTOLOGIA DE ENDURANTES.....	52
3.5.2. UFO-B: UMA ONTOLOGIA DE PERDURANTES	58
3.5.3. UFO-C: UMA ONTOLOGIA DE ENTIDADES SOCIAIS	59
3.6. CONSIDERAÇÕES FINAIS DO CAPÍTULO	66
4. ONTOLOGIA DO GERENCIAMENTO DE NÍVEL DE SERVIÇO DO PADRÃO ITIL.....	67

4.1.	ONTOLOGIA DO DOMÍNIO DE GERENCIAMENTO DE NÍVEL DE SERVIÇO.....	67
4.1.1.	QUESTÕES DE COMPETÊNCIA E BASE DA ONTOLOGIA.....	67
4.1.2.	ONTOLOGIA DE GERENCIAMENTO DE NÍVEL DE SERVIÇO	69
4.1.3.	ONTOLOGIA DE REQUISIÇÃO DE SERVIÇO.....	75
4.1.4.	ONTOLOGIA DE NÍVEL DE SERVIÇO	78
4.2.	CONSIDERAÇÕES FINAIS DO CAPÍTULO	83
5.	GERENCIAMENTO DE SERVIÇOS DA PLATAFORMA INFRAWARE.....	85
5.1.	APLICAÇÕES SENSÍVEIS AO CONTEXTO.....	85
5.2.	A PLATAFORMA DE SERVIÇOS INFRAWARE.....	86
5.3.	APLICAÇÃO DAS BOAS PRÁTICAS ITIL NA PLATAFORMA INFRAWARE.....	89
5.4.	REQUISITOS DO GERENTE DE SERVIÇOS DA INFRAWARE	91
5.5.	PROPOSTA DE UMA NOVA ARQUITETURA PARA O GERENTE DE SERVIÇOS	94
5.5.1.	IMPLEMENTAÇÃO DO ITIL NO MÓDULO DE SERVIÇO.....	97
5.6.	CONSIDERAÇÕES FINAIS	99
6.	ESTUDO DE CASO: APLICAÇÃO DO PROCESSO DE GERENCIAMENTO DE NÍVEL DE SERVIÇO NA PLATAFORMA INFRAWARE	101
6.1.	APRESENTAÇÃO	101
6.2.	LINGUAGENS DE IMPLEMENTAÇÃO DO MODELO	101
6.3.	CENÁRIO DE APLICAÇÃO.....	105
6.3.1.	DESCRIÇÃO DO CENÁRIO DE MONITORAMENTO DOMILICIAR	108
6.4.	EXECUÇÃO DAS ATIVIDADES DE GERENCIAMENTO DE SERVIÇO.....	110
6.4.1.	SUB-ATIVIDADE DE IDENTIFICAÇÃO	110
6.5.	IMPLEMENTAÇÃO E INFERÊNCIA DE INFORMAÇÕES GERENCIAIS.....	111
6.6.	AVALIAÇÃO	116
7.	CONCLUSÕES.....	118
	REFERÊNCIAS	121
	ANEXO A: REGRAS IMPLEMENTADAS EM SWRL E EXECUTADAS PELO JESS	127

1. Introdução

1.1. Motivação

Nas décadas de 1990 e 2000, o desenvolvimento das Tecnologias da Informação (TI) gerou, e ainda tem gerado, grandes impactos nos processos de negócio das organizações e empresas. Com o surgimento e disseminação dos computadores pessoais (*desktops* e *laptops*), das redes de computadores, das tecnologias de comunicação e da Internet, as organizações puderam divulgar no mercado seus produtos e serviços de forma mais rápida. Todo esse desenvolvimento transformou a “Era Industrial” na “Era da Informação”, levando empresas, governos e demais organizações a exigirem e fornecerem serviços mais rápidos e dinâmicos, capazes de suportar as necessidades dos negócios, tanto suas quanto dos clientes e parceiros.

O *Gerenciamento de Serviços de TI* surgiu com o intuito de dar suporte a todo o processo de fornecimento de serviços de tecnologia da informação com qualidade, eficiência, eficácia e baixo custo, alinhados com as necessidades do negócio dos clientes e da organização fornecedora. Ao longo dos anos, a TI tornou-se imprescindível ao negócio e o seu gerenciamento e o gerenciamento dos serviços por ela fornecidos permitiram que as organizações evoluíssem, passando do estágio de simples fornecedores de TI para verdadeiros parceiros estratégicos. Ultimamente, tem-se advogado que para o negócio de uma organização ser um sucesso, ela precisa estar envolvida e comprometida com o que a TI é capaz de fazer e os seus processos de TI precisam estar completamente integrados com os processos de negócio. Para entregar os serviços que uma organização precisa, a TI necessita ser gerenciada *pelo negócio como um negócio*. Esse é o principal objetivo do que vem sendo denominado de *Governança em TI* [Kordel, 2004].

Grande parte desta evolução se deu com o movimento realizado pela indústria de TI em direção à padronização das boas práticas de gerenciamento de serviços em sistemas de informação, apoiado no surgimento de *frameworks* como o *Control Objectives for Information and related Technology* (CobiT) [CobiT, 2008], de bibliotecas de boas práticas como o *Information Technology Infrastructure Library* (ITIL) [Rudd & Hodgkiss, 2004], e de padrões como o ISO 20000 [ISO 20000, 2008], que facilitam e permitem promover o conceito de governança nos sistemas de informação, garantindo aos usuários e aplicações

benefícios como maior qualidade, disponibilidade e confiabilidade dos diversos serviços oferecidos.

O ITIL, em particular, objeto de estudo deste trabalho, define um conjunto de processos, procedimentos e atividades que podem ser aplicados a qualquer organização. O objetivo do ITIL é alinhar os processos da organização com seus objetivos estratégicos, focando sempre na excelência dos serviços oferecidos.

A biblioteca ITIL vem tornando-se um padrão “*de facto*” para o gerenciamento de serviços de TI. Ela vem sendo introduzida gradativamente nas organizações, que utilizam as suas boas práticas como referência para a implantação de serviços de gerência de TI, em diversos cenários de uso. Entretanto, um dos problemas encontrados na implantação de sistemas baseados em ITIL nas organizações é que, pelo fato do padrão não fornecer uma especificação formal dos conceitos do universo de discurso de gerenciamento de serviços de TI (a especificação é feita em linguagem natural), interpretações dos conceitos ITIL, feitas de maneira *ad hoc*, podem resultar na implementação de sistemas ambíguos e com baixo grau de interoperabilidade. Existe, portanto, um campo promissor para a investigação de técnicas e linguagens apropriadas para a modelagem e formalização do domínio de gerenciamento de serviços de TI. Essas técnicas e linguagens devem permitir a geração de modelos conceituais de alta expressividade, que consigam capturar a semântica dos conceitos envolvidos em um domínio tão complexo como o apresentado neste trabalho: gerenciamento de serviços de TI.

Alinhado com isso, há alguns anos vem crescendo o uso de **ontologias** [Mealy, 1967], [Smith & Welty, 2001] aplicadas à Ciência da Computação. Uma ontologia consiste em uma descrição formal dos conceitos e dos relacionamentos existentes em um determinado domínio [Guarino, 1998]. Sua aplicabilidade tem sido reconhecida em diversas áreas, como Inteligência Artificial (IA), Engenharia de Software e *Web Semântica*, possibilitando a criação de modelos conceituais claros, concisos e não ambíguos. Esses modelos permitem representar as características mais relevantes do domínio e impedir que sejam modeladas situações impossíveis. Por exemplo, uma ontologia do domínio de genealogia não deve permitir que seja modelada uma situação em que um indivíduo é pai do seu pai ou de si próprio.

A aplicação de modelos formais usando ontologias para descrever os conceitos do domínio de gerenciamento de serviços, segundo a abordagem ITIL, por exemplo, pode resultar numa uma série de benefícios. Primeiramente, permite fornecer uma descrição clara, precisa e explícita

dos conceitos ITIL, o que evita ambigüidades de interpretações e inconsistências nas descrições dos termos e conceitos definidos no padrão, além de possibilitar o compartilhamento de conhecimento sobre processos, atividades e conceitos desse domínio entre os atores envolvidos (clientes, usuários, provedores de serviço, etc.). Isso facilita a criação de sistemas computacionais que se comunicam utilizando uma sintaxe e semântica única a respeito dos conceitos e processos ligados ao gerenciamento do serviço. Permite, ainda, a utilização de mecanismos automatizados de execução de procedimentos para capacitar tais sistemas a realizarem processamentos e inferências automáticas, analisando sobre as informações existentes e gerando novas informações úteis sobre o domínio. Tal capacidade de geração de informações é demonstrada neste trabalho através de um caso de uso.

1.2. Escopo do Trabalho

1.2.1. Gerenciamento de Nível de Serviço

Dentre os processos definidos na biblioteca ITIL, o processo de **Gerenciamento de Nível de Serviço** mostra-se fundamental por definir acordos com os clientes sobre o tipo e a qualidade dos serviços a serem entregues, por garantir a implementação desses acordos, e por tentar fazer com que os serviços requisitados pelos clientes sejam sempre mantidos e melhorados.

O Gerenciamento de Nível de Serviço é o processo de negociação, definição, monitoração, gerência e melhoria da qualidade dos serviços fornecidos, visando encontrar o equilíbrio certo entre a qualidade da oferta e da demanda. Dessa forma, percebe-se que a implementação desse processo de gerenciamento é um dos primeiros passos para viabilizar o fornecimento de serviços de qualidade para os usuários e aplicações.

Em virtude da importância que o Gerenciamento de Nível de Serviço tem para o fornecimento de serviços de qualidade e devido à complexidade de se modelar e implementar todas as boas práticas ITIL, este trabalho se restringe ao escopo da disciplina de Gerenciamento de Nível de Serviço definida pelo ITIL.

1.2.2. Serviços Sensíveis ao Contexto

Um cenário que desponta atualmente é o da mobilidade do usuário e o uso de aplicações adaptativas. Com a expansão e a redução dos custos da tecnologia, muitos usuários estão

trocando as estações de trabalho para dispositivos móveis com razoável poder de computação, como *laptops*, *palmtops*, *Personal Digital Assistant (PDA)* e *smartphones*. Dessa forma, eles não ficam restritos aos ambientes estáticos e relativamente previsíveis dos escritórios, podendo interagir em vários ambientes.

Esta mudança constitui um passo em direção ao paradigma da *computação ubíqua (Ubiquitous Computing)*, ou "terceira onda da computação", previsto por Mark Weiser (Weiser, 1991). Weiser, observando ambientes com recursos computacionais capazes de fornecer serviços e informações em qualquer lugar, a qualquer momento, propõe que haja uma integração contínua entre ambiente e tecnologia para auxiliar os usuários em suas atividades do dia-a-dia.

A computação ubíqua permite criar uma série de novas aplicações, conhecidas como *aplicações (móveis) sensíveis ao contexto*, as quais são capazes de perceber o ambiente, utilizar informações sobre o contexto do usuário e dinamicamente se adaptar, selecionar e executar os serviços que melhor atendam às necessidades do usuário. Diferentemente das aplicações tradicionais, as aplicações sensíveis ao contexto usam adicionalmente informações implícitas vindas do contexto físico e computacional do ambiente e dos seus usuários, e não somente informações explícitas fornecidas por eles.

Aplicações sensíveis ao contexto já são uma realidade [Family Locator, 2007]), [Shanahan, et al, 2004], [iHospital, 2007] e trazem em si características intrínsecas de complexidade que chamam atenção para uma série de novos desafios que os projetistas de *hardware* e *software* precisam considerar. Alguns desses desafios estão ligados à aquisição, modelagem, armazenamento, distribuição e monitoração de informações contextuais e serviços. Estes desafios motivaram a criação de infraestruturas ou plataformas de *middleware* especializadas no suporte ao desenvolvimento e execução de aplicações sensíveis ao contexto [Chen & Kotz, 2002], [Dey, 2001], [Dockhorn, 2003], [Rocha & Endler, 2006]). Essas plataformas têm por objetivo prover serviços e funcionalidades básicas para as aplicações sensíveis ao contexto, usando mecanismos e interfaces que escondam destas a complexidade da manipulação de contexto. Uma dessas plataformas, que vem sendo desenvolvida no Laboratório de Pesquisas em Redes e Multimídia do DI/UFES, é a **Infraware** [Pereira Filho et al, 2006],[Pessoa, 2006], introduzida no Capítulo 5.

Do potencial apresentado por esse novo paradigma, pode-se prever que haverá uma grande propagação de novos serviços móveis e sensíveis ao contexto, que serão usados por um grande número de aplicações e usuários, provavelmente em vários domínios. Este cenário sugere a necessidade de existência de um importante requisito do *middleware*, que é o de levar a capacidade de gerenciamento eficiente e eficaz ao domínio da computação ubíqua, requisito esse ainda praticamente inexplorado pelas plataformas atuais.

Transportadas ao domínio dos serviços sensíveis ao contexto, as boas práticas de gerenciamento de serviços ITIL poderiam fornecer uma base sólida para a melhoria da arquitetura de gerenciamento dos serviços oferecidos pelo *middleware* aos vários usuários e aplicações. Por outro lado, modelos formais baseados em ontologias poderiam ser usados como insumo para evolução do próprio ITIL, à medida que conceitos específicos deste domínio da computação ubíqua poderiam ser introduzidos em novas versões do padrão.

Assim, projetando um cenário onde: (i) organizações e empresas utilizam intensamente o paradigma da computação ubíqua, (ii) qualquer dispositivo de computação é potencialmente um novo fornecedor de serviços, e (iii) a rede de provedores de serviços pode vir a ter natureza *ad hoc*, com dispositivos provedores alterando dinamicamente a topologia ao entrarem e saírem da rede, o gerenciamento de serviços torna-se ainda mais complexo e o entendimento e descrição precisa dos conceitos do domínio de gerenciamento de nível de serviços ITIL (ou seja, a sua formalização) um requisito ainda mais fundamental.

1.3. Objetivos

Os principais objetivos deste trabalho são:

1. Formalizar, usando ontologias, parte dos conceitos do domínio de Gerenciamento de Nível de Serviço do ITIL. O intuito é criar modelos conceituais deste domínio que possam ser aplicados em vários cenários e ambientes que usam alguma infra-estrutura de suporte a serviços, permitindo um entendimento comum e o compartilhamento de informações do domínio, além da realização de inferência e geração de novas informações. A construção dos modelos formais usa a metodologia SABiO [Falbo, 1998] junto com ontologias de fundamentação (UFO) [Guizzardi, 2005] [Guizzardi et al, 2008], dado o auxílio que proporcionam durante a construção de tais modelos. Para validar a especificação formal do domínio de Gerenciamento de Nível de

Serviço, uma implementação desse modelo em uma linguagem de descrição de ontologias se torna necessário e faz parte do escopo deste trabalho.

2. Validar os modelos desenvolvidos e formalizados através da sua implementação por meio de ontologias em um estudo de caso. O objetivo é gerar informações de gerência e controle com relação às solicitações e entrega de serviços, a partir de inferências dos dados modelados e instanciados no cenário de aplicação escolhido.
3. Investigar a introdução do Gerenciamento de Nível de Serviço ITIL na plataforma de serviços sensíveis ao contexto Infracore. Em outras palavras, adaptar a arquitetura do módulo responsável pelo fornecimento de serviços da Infracore (o Módulo de Serviço), para receber e incorporar as estruturas e os processos do Gerenciamento de Nível de Serviço do ITIL.

1.4. Estrutura do Trabalho

Além desta Introdução, este trabalho está dividido nos seguintes capítulos:

- Capítulo 2: Introduz o conceito de Gerenciamento de Serviço de TI, destacando sua importância para o desenvolvimento e crescimento das organizações. Esse capítulo apresenta também uma visão geral do padrão de gerenciamento de serviços ITIL, e descreve em detalhes o domínio de Gerenciamento de Nível de Serviço.
- Capítulo 3: Descreve o conceito de ontologias e como elas são aplicadas para a construção de modelos conceituais de domínio. O capítulo também apresenta a ontologia de fundamentação UFO, usada como base para a criação da ontologia de domínio de gerenciamento de nível de serviço resultante deste trabalho.
- Capítulo 4: Descreve a formalização do domínio de Gerenciamento de Nível de Serviço, através de um conjunto de modelos conceituais criados, a partir dos conceitos fundamentais definidos na UFO, que representam partes distintas desse processo.
- Capítulo 5: Introduz o paradigma de computação sensível ao contexto. Apresenta a plataforma de serviços Infracore e discute uma proposta de extensão da arquitetura conceitual do Módulo de Serviço dessa plataforma, com funções mais descentralizadas, e apta a incluir as funcionalidades do processo de Gerenciamento de Nível de Serviço ITIL.
- Capítulo 6: Apresenta o estudo de caso (prova de conceito) deste trabalho, consistindo de um cenário médico-hospitalar que simula um possível uso do Módulo de Serviço

da plataforma. São apresentadas inferências de informações sobre o fornecimento de serviços utilizando os modelos desenvolvidos e as regras de Gerenciamento de Nível de Serviço.

- Capítulo 7: Apresenta as considerações finais do trabalho, com base nos objetivos propostos e no estudo desenvolvido. Sugere ainda alguns desdobramentos futuros.

2. Gerenciamento de Serviços e Governança de TI

2.1. Gerenciamento e Governança

Atualmente, para a maioria das organizações, a Tecnologia da Informação (TI) é vista como algo essencial para o gerenciamento de transações, informações e conhecimento necessários para iniciar e manter atividades econômicas e para dar suporte, manter e fazer crescer o negócio [Kordel, 2004]. De fato, atualmente, a Tecnologia da Informação representa um dos mais importantes fatores para o alto desempenho das organizações na conquista dos seus objetivos de negócio.

Ao longo dos anos, a TI se tornou a espinha dorsal dos negócios a ponto de chegar a ser impossível para as muitas organizações funcionar sem ela. Ela é hoje um elemento essencial de uma empresa [Grembergen, 2003].

Segundo [Sallé, 2004], quando uma organização evolui, deixando de ser um fornecedor de tecnologia e passando a ser um parceiro estratégico, tal organização passa por três estágios, ilustrados na Figura 2-1 - Evolução da função da TI em uma organização [Sallé, 2004].



Figura 2-1 - Evolução da função da TI em uma organização [Sallé, 2004]

Cada estágio é alcançado a partir do anterior, e o processo começa no *Gerenciamento de Infra-estrutura de TI* (IT Infrastructure Management – ITIM). O foco nessa fase é melhorar o gerenciamento da infra-estrutura da empresa, o que significa maximizar o retorno sobre os

ativos de computação e assumir o controle da infra-estrutura, dos dispositivos nela contida, e dos dados gerados.

A próxima fase, a de *Gerenciamento de Serviço de TI (IT Service Management – ITSM)*, acontece quando a organização de TI consegue identificar, planejar, entregar e oferecer suporte aos serviços que seus clientes necessitam, de acordo com os níveis de serviço estabelecidos.

Por fim, quando a organização atinge o nível de *Governança de TI (IT Governance)*, seus processos de TI estão completamente integrados com seus processos de negócio, e ela se transforma em um verdadeiro parceiro de negócio.

2.1.1. Gerenciamento de Serviços de TI

O *Gerenciamento de TI* é uma combinação de duas áreas de estudo: Tecnologia da Informação e Gerenciamento [Kumar et al, 2007]. Segundo [Kumar et al, 2007], existem duas características particulares para essa definição. A primeira diz respeito ao gerenciamento de uma coleção de sistemas, infra-estrutura e informações que residem neles, sendo muito utilizada em manuais técnicos e publicações em vários fornecedores de TI. A segunda implica no gerenciamento da TI como uma função do negócio, e é originada da discussão e criação do *Information Technology Infrastructure Library (ITIL)* (ver seção 2.2.1).

Dessas duas características surgiu um novo modelo de gerenciamento, centrado no serviço e focado nos clientes, conhecido como **Gerenciamento de Serviço de TI (IT Service Management – ITSM)**. Segundo [Bon, 2002], “*os provedores de serviço de TI não podem mais sustentar o foco na tecnologia e em suas organizações internas, eles agora têm que considerar a qualidade dos serviços que eles fornecem e focar no relacionamento com os clientes*”.

ITSM é uma disciplina capaz de gerenciar sistemas de TI, focando na perspectiva dos clientes em relação à contribuição da TI sobre seus negócios. O ITSM consiste de um conjunto de processos que auxilia as organizações de TI a identificarem os serviços que seus clientes necessitam e a se concentrarem no planejamento, entrega e suporte desses serviços (dentro do prazo, com qualidade e respeitando os custos estabelecidos) para satisfazer os requisitos de disponibilidade, desempenho e segurança [Sallé, 2004].

O ITSM é usado em organizações (principalmente nas de Tecnologia de Informação e Comunicação – TIC) que fornecem bens e/ou serviços. Ele não se preocupa em detalhar como se usa um produto específico ou com detalhes técnicos de um sistema que está sendo gerenciado. O foco desse gerenciamento é fornecer um *framework* capaz de estruturar as atividades relacionadas à TI (arquitetura operacional), e as interações da equipe técnica de TI com os clientes do negócio.

O gerenciamento de serviço possui um papel muito importante na sustentação das operações do negócio. A gerência precisa coordenar e trabalhar em conjunto com o negócio, permitindo o crescimento da organização, e ela deve evitar que a tecnologia e a TI assumam o controle de todo o negócio [Rudd & Hodgkiss, 2004].

2.1.2. ITSM orientados a processos

Qualquer organização que deseja atingir sua visão, sua missão, seus objetivos e sua política precisa em primeiro lugar estruturar todas as suas atividades. Essa estruturação deve ser feita de forma que seja possível notar a contribuição de cada grupo de atividades para os objetivos do negócio, além de possibilitar a observação das relações entre esses grupos.

Tais grupos de atividades são conhecidos como **processos**, e se eles forem estruturados de acordo, eles são capazes de descrever e responder a alguns questionamentos sobre o objetivo da organização, o que ser feito e como deve ser feito para a organização chegar onde deseja. Muitas organizações já estabeleceram seus processos de como atingir seus objetivos. Entretanto, tais processos ainda precisam passar por melhorias. O ITSM pode ser visto como um mecanismo de melhoria dos processos de uma organização. A Figura 2-2 ilustra um modelo de melhoria de processos.

Um processo, segundo [Bon et al, 2006], é um conjunto de atividades logicamente relacionadas empenhadas em atingir um objetivo específico. Processos são normalmente descritos através de procedimentos e instruções de trabalho. Nesses procedimentos e instruções, informações a mais são detalhadas sobre o processo, por exemplo, os responsáveis na execução de um processo, e como e quando ele deve ser executado.

Normalmente os processos não ficam restritos às divisões e departamentos de uma organização. Eles transcendem tais fronteiras organizacionais. Por isso deve haver responsáveis

para os processos, isto é, pessoas encarregadas pela definição do processo e por garantir que todos os envolvidos na sua execução sejam informados sobre qualquer mudança.

Uma vez estabelecido o processo, é preciso avaliá-lo com relação à qualidade. Para isso, devem ser estabelecidos, pela organização, indicadores de qualidade e de desempenho.

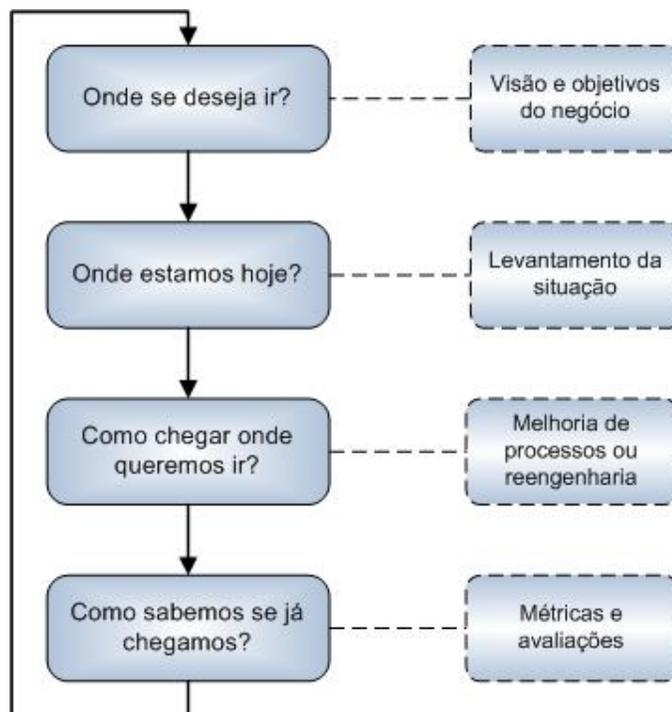


Figura 2-2 - Modelo simples de melhoria de processos [Bon et al, 2006]

Alguns dos benefícios do uso de processos para o Gerenciamento de Serviços de TI são listados abaixo [Calvi, 2007]:

1. Os objetivos (resultados) são descritos de uma maneira mais clara para serem atingidos;
2. Os processos podem ser monitorados individualmente ou como um todo, mostrando alguns caminhos para a melhoria das atividades de um único ou vários processos, promovendo, assim, uma melhoria contínua;
3. A criação de papéis e de responsabilidades mais claros, organizados de maneira eficiente e eficaz, tornando mais fácil a prevenção de conflitos de interesse e melhorando os resultados dos serviços;

4. Atividades com um objetivo comum que devem ser executadas em várias áreas da organização podem ser melhor controladas através de um processo bem definido, estruturado e abrangente.

2.2. Padrões de Gerenciamento de TI

Ao longo dos anos, diversos frameworks de gerenciamento de serviços de TI foram propostos: BS 15000 / ISO 20000 [BS 15000, 2002], [ISO 20000-1, 2005] [ISO 20000, 2008], HP ITSM [HP, 2000] [HP, 2003a], *Microsoft Operations Framework* [MOF, 2001] e a biblioteca de boas práticas ITIL [Bon et al, 2006] [HP, 2003b].

Apesar de todos esses *frameworks* ensinarem os mesmos princípios usados na melhoria da qualidade, o ITIL foi o primeiro padrão que conseguiu descrever de forma consistente os vários processos que são fundamentais para o gerenciamento de serviço de TI. Seu surgimento foi imprescindível para o desenvolvimento de muitos outros padrões de gerenciamento, incluindo o IT Process Model da IBM e o próprio MOF da Microsoft. Maiores detalhes podem ser vistos em [Sallé, 2004].

É por estas razões que este trabalho utiliza como *framework* de ITSM o ITIL, mais especificamente a sua versão 2, dado que a versão 3 foi lançada recentemente, e sua previsão de substituir totalmente a versão 2 é no final de 2008 ou início de 2009. A escolha da versão 2 para modelar o Gerenciamentos de Nível de Serviço não é afetada com a evolução da biblioteca para a versão 3, visto que a nova versão também dá suporte as essas disciplinas.

2.2.1. O Padrão ITIL

A biblioteca *Information Technology Infrastructure Library* (ITIL) foi criado em 1989 pela *Central Computer and Telecommunications Agency* (CCTA) do Reino Unido. Seu objetivo era o de melhorar a qualidade do serviço prestado pelo governo britânico. Atualmente, o ITIL é administrado pelo *Office of Government Commerce* (OGC) (antigo CCTA), também do Reino Unido, e sua administração é apoiada pelo *IT Service Management Forum* (itSMF).

O ITIL é uma descrição detalhada de um conjunto inter-relacionado, coerente, consistente e compreensível de boas práticas, incluindo *checklists*, tarefas, procedimentos e responsabilidades para o processo de gerenciamento de serviços de TI. Seu intuito é de melhorar a qualidade dos serviços de TI entregues aos clientes, fazendo com que as

organizações atinjam a excelência (obtenção de eficácia e eficiência) no uso sistemas de informação.

Entretanto, o ITIL reconhece que não existe uma solução única para projetar e implementar processos otimizados para o gerenciamento e a entrega de serviços de TI de qualidade [Rudd & Hodgkiss, 2004]. Ao longo do seu desenvolvimento, desde sua criação, o ITIL vem recebendo contribuições de vários especialistas das diversas organizações de TI. Conseqüentemente, tais contribuições resultaram em um *framework* que fornece uma abordagem estruturada para todas as atividades, relacionadas ao fornecimento de serviços, de uma organização de TI.

O ITIL não prescreve como implementar tais atividades. Ele não é um método e sim um *framework* que auxilia no planejamento dos processos, papéis e atividades essenciais para o fornecimento de serviços. Este *framework* está organizado em sete publicações, cada uma descrevendo uma área específica da manutenção e operação da infra-estrutura de TI.

A Figura 2-3 mostra o *framework* de publicação do ITIL.

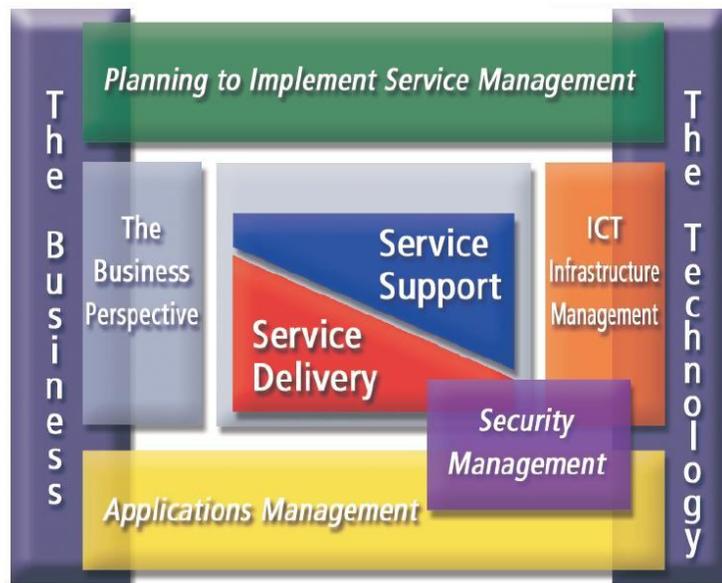


Figura 2-3 - Framework de publicações do ITIL [Macfarlane & Rudd, 2005]

As principais práticas para o gerenciamento de serviço são descritas nas áreas de Suporte a Serviço (*Service Support*) e Entrega de Serviço (*Service Delivery*). O Suporte a Serviços se

concentra na operação e no suporte dos serviços de TI, enquanto que a Entrega de Serviços preocupa-se com o planejamento e a melhoria da prestação de serviços de TI a longo prazo.

2.2.1.1. Suporte a Serviço: Processos Operacionais ITIL

O Suporte a Serviço descreve como os clientes e os usuários podem obter acesso aos serviços que eles necessitam para dar suporte às suas atividades e ao negócio. O Suporte a Serviço é composto por uma função Central de Serviço e cinco disciplinas ITIL que são: Gerenciamento de Liberação, Gerenciamento de Configuração, Gerenciamento de Incidente, Gerenciamento de Problema e Gerenciamento de Mudança. A Figura 2-4 apresenta os relacionamentos entre cada um desses processos.

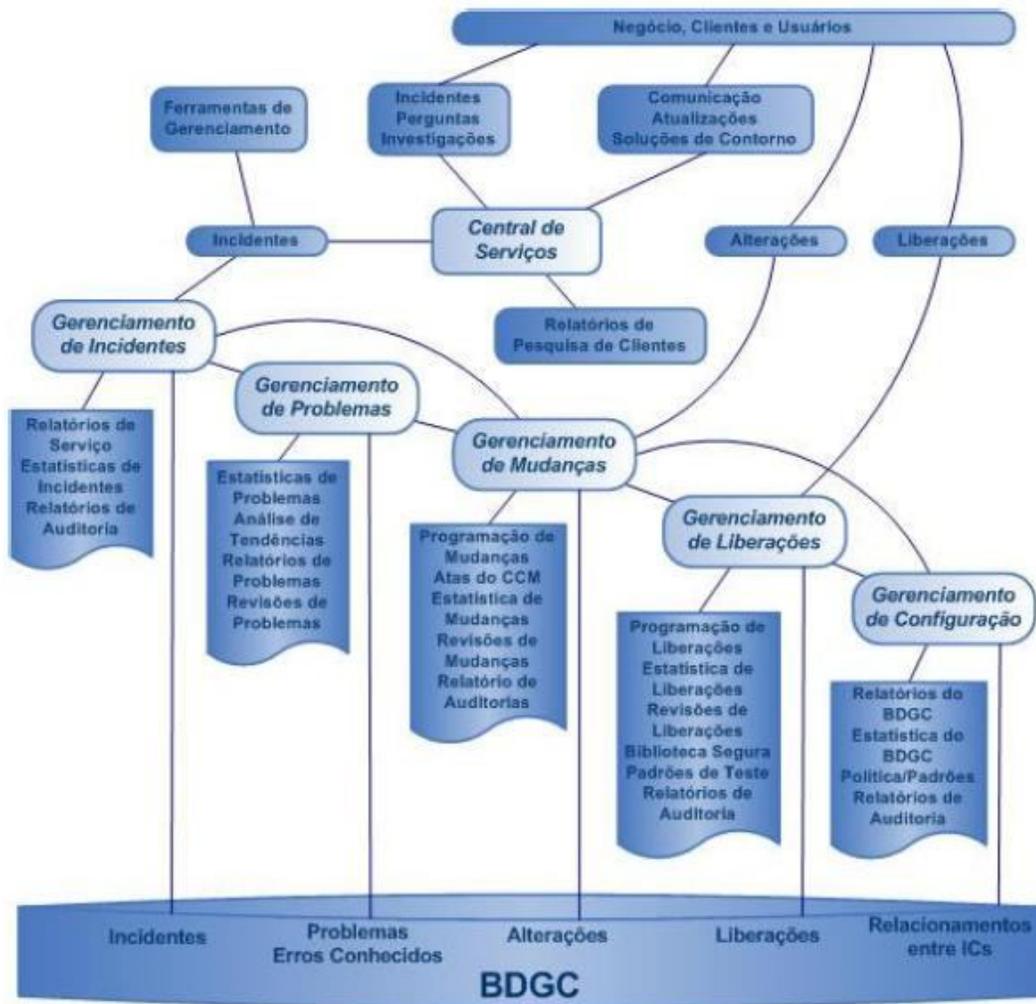


Figura 2-4 - Processos de suporte a serviço do ITIL [Macfarlane & Rudd, 2005]

- Central de Serviços (*Service Desk*): é uma importante função para um Gerenciamento de Serviços efetivo. Funciona como um ponto único de contato entre clientes, usuários e o gerenciamento de serviço. Trata incidentes, solicitações, requisições de serviços e oferece uma interface operacional para as demais atividades do ITIL. É uma função imprescindível no acompanhamento e gerenciamento do ciclo de vida de um incidente na infra-estrutura, gerando relatórios, comunicações e promovendo a TI a seus clientes.
- Gerenciamento de Configuração: O ITIL define o conceito *Item de Configuração* (IC) para referir-se aos componentes que fazem parte da infra-estrutura de TI. O objetivo do Gerenciamento de Configuração é gerenciar essa infra-estrutura, identificando, registrando e controlando todos os IC's. Esse processo também deve fornecer informações sobre a infra-estrutura de TI, monitorando-a e controlando-a, com o intuito de prover informações a todos os outros processos ITIL. Ademais, possui um importante papel na verificação e avaliação do impacto das mudanças no ambiente de TI.
- Gerenciamento de Incidentes: Um *incidente* é qualquer evento que não faz parte do funcionamento padrão de um serviço e que causa, ou pode causar, uma interrupção do mesmo ou uma redução da sua qualidade. Esta disciplina busca resolver incidentes e restabelecer, o mais rápido possível, o fornecimento do serviço ao cliente, procurando minimizar o impacto dos incidentes sobre o negócio. Precisa garantir também que a qualidade do serviço e a sua disponibilidade sejam condizentes com os níveis de serviço (prazo e custo, por exemplo) acordados.
- Gerenciamento de Problema: Um *problema* é uma condição definida e identificada a partir de um ou mais incidentes que apresentam sintomas comuns, mas que a causa é desconhecida. Um *erro conhecido* é um problema cuja causa raiz é conhecida, e que possui uma solução estabelecida. O objetivo do Gerenciamento de Problema é assegurar a estabilidade dos serviços de TI através da identificação e eliminação dos erros conhecidos na infra-estrutura de TI. É responsável por gerenciar problema, identificar sua causa raiz e propor soluções para eliminá-lo, armazenando todas essas informações em um banco de soluções.
- Gerenciamento de Mudança: Uma *mudança* é uma ação que gera um novo estado para um ou mais IC's. Uma *requisição para mudança* é a entrada principal para o processo de Gerenciamento de Mudança. Esta disciplina visa gerenciar as mudanças,

assegurando que elas sejam rápidas, fáceis, consistentes e autorizadas. Para isso, ela utiliza métodos e técnicas padronizadas para lidar com todas as mudanças na infraestrutura de TI, buscando minimizar incidentes relacionados à mudança.

- Gerenciamento de Liberação: O objetivo desta disciplina é administrar a distribuição e o controle de liberação de *software*, *hardware* e atualizações na infra-estrutura de TI. Apenas os elementos verificados, testados e aprovados é que são colocados disponíveis para a operação. Desta forma, se houver alguma falha, é possível retornar para as versões originais.

Para mais informações sobre as cada uma das disciplinas do Suporte a Serviço, consulte [Bon et al, 2006] e [Macfarlane & Rudd, 2005].

2.2.1.2. Entrega de Serviço: Processos Táticos ITIL

A Entrega de Serviço descreve os serviços que o cliente necessita para dar suporte o seu negócio e o que é necessário para prover esses serviços. Ele é composto por cinco disciplinas ITIL, a saber: Gerenciamento de Nível de Serviço, Gerenciamento de Capacidade, Gerenciamento de Disponibilidade, Gerenciamento de Continuidade dos Serviços de TI e Gerenciamento Financeiro. A Figura 2-5 mostra os relacionamentos entre cada um desses cinco processos.

- Gerenciamento de Nível de Serviço: Assegura a contínua identificação, monitoração e revisão dos níveis ideais (prazo, custo e disponibilidade do serviço, por exemplo) dos serviços de TI, que foram concordados para satisfazer as necessidades do negócio. Para que isso seja possível, é preciso haver cooperação entre os provedores de serviço de TI e os clientes. O objetivo desta disciplina é deixar claro os acordos entre os clientes e a organização de TI com relação ao tipo e à qualidade dos serviços de oferecidos. Por ser esta a disciplina ITIL objeto de estudo deste trabalho, maiores detalhes são apresentados adiante, na Seção 2.3.
- Gerenciamento de Capacidade: É o processo de otimização do custo, do tempo de aquisição e da instalação dos recursos de TI, que dão suporte aos acordos feitos com o cliente. Identifica e especifica a demanda e as necessidades do cliente, buscando alinhá-las com os recursos disponíveis. Objetiva planejar a capacidade dos serviços, dos recursos e das demandas na quantidade e momento certo, de maneira eficiente e a um custo efetivo.

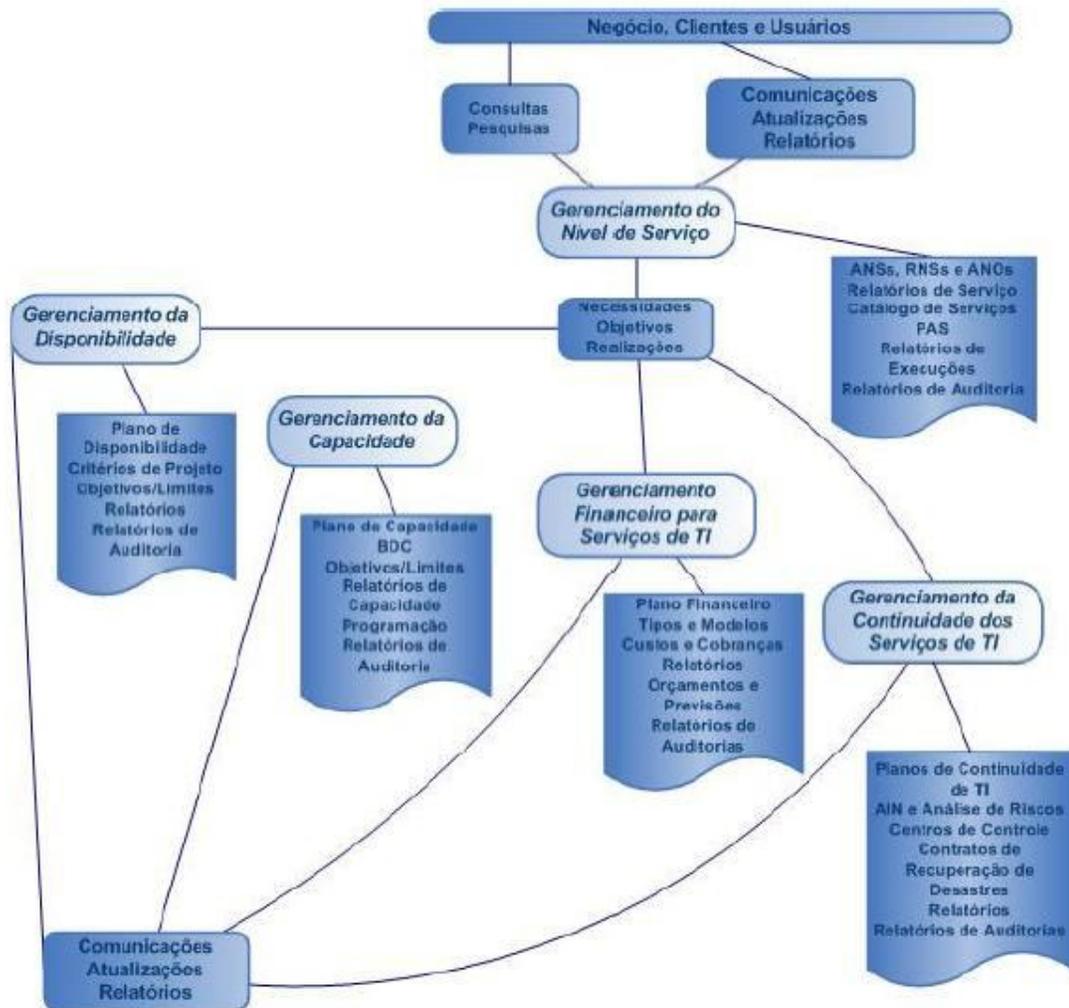


Figura 2-5 - Processos de entrega de serviço do ITIL [Macfarlane & Rudd, 2005]

- Gerenciamento de Disponibilidade: É o processo de assegurar a instalação apropriada dos recursos, métodos e técnicas, que dão suporte à disponibilidade dos serviços de TI acordados com o cliente. Isso deve ser feito a um custo justificável. Identifica, define e prepara as medidas necessárias para garantir a disponibilidade requerida pelos serviços. Além do mais, lida com a otimização da manutenção dos recursos e modela medidas para reduzir sempre o número de incidentes.
- Gerenciamento de Continuidade dos Serviços de TI: Busca assegurar a disponibilidade e a rápida restauração dos serviços de TI na ocorrência de um desastre. Mantém planos de contingência e de recuperação de desastres, sobrevivência do negócio, riscos e vulnerabilidades. Seu papel é de dar suporte à continuidade dos negócios de seus clientes.

- **Gerenciamento Financeiro:** Administra os custos, a alocação de recursos financeiros e o retorno do investimento. Realiza provisões orçamentárias dos serviços de TI levando em consideração os custos envolvidos e os possíveis benefícios nos investimentos, em especial, nas tomadas de decisões a respeito das mudanças no ambiente.

2.2.1.3. Evolução da Biblioteca ITIL

O ITIL desde sua concepção recebe constantes contribuições e atualizações de seu conteúdo. Sua primeira versão era baseada em funções que variavam de acordo com a TI. A segunda versão (versão atual) baseia-se em processos que formam um *framework* de gerenciamento de serviços de TI. Sua mais recente versão [ITIL, 2008] é baseada em ciclos de vida dos serviços, e incorpora o que houve/há de melhor nas suas versões anteriores, e substituirá a versão 2 gradativamente. Este trabalho está baseado na segunda versão do padrão.

2.3. Descrição do Domínio de Gerenciamento de Nível de Serviço

O *Gerenciamento de Nível de Serviço (Service Level Management – SLM)* é essencial em qualquer organização, pois é responsável pelo processo de negociação e definição dos serviços a serem utilizados pelo negócio. O SLM procura encontrar o serviço que se encaixa na demanda, no custo e em outros requisitos do usuário, além de levar em conta a qualidade do fornecimento do serviço por parte do provedor.

Ele é responsável por estabelecer, cumprir e manter os Acordos de Nível de Serviço (*Service Level Agreements – SLA*), os Acordos de Nível Operacional (*Operational Level Agreements – OLA*), os Contratos de Apoio (*Underpinning Contracts – UC*) e os Planos de Qualidade de Serviço. Ele inspeciona a realização dos serviços para assegurar que sua qualidade seja mantida e, quando necessário, melhorada. Além do mais, ele é responsável por assegurar que os impactos sobre a qualidade do serviço sejam os mínimos possíveis. Se algum dos acordos não está sendo cumprido, o SLM busca identificar os motivos.

Antes de detalhar o processo e suas atividades, alguns conceitos básicos que existem neste processo de gerenciamento são apresentados a seguir.

2.3.1. Conceitos básicos

Para o SLM, o *cliente* é o representante de uma organização que está autorizado a fazer acordos em nome da mesma para a obtenção de serviços de TI. O cliente possui um papel diferente do usuário final do serviço de TI. Por outro lado, o *provedor* é o representante de uma organização que está autorizado a fazer acordos em nome da mesma para o fornecimento de serviços de TI.

Uma Requisição de Nível de Serviço (*Service Level Requirement – SLR*) é um documento que traz as definições detalhadas das necessidades dos clientes, e é usado para desenvolver, modificar e iniciar serviços. Serve como um roteiro para a execução de um serviço e do SLA (definido a seguir) a ele associado.

O documento de Especificação de Serviço (*Spec Sheets*) descreve o relacionamento entre funcionalidade (acordado com o cliente) e tecnologia (implementada na organização de TI) e fornece a especificação detalhada do serviço. Este documento traduz as necessidades descritas na SLR (especificações externas) nas definições técnicas necessárias para implementar o serviço (especificações internas da própria organização).

Um Acordo de Nível de Serviço (*Service Level Agreements – SLA*) é um acordo entre a organização de TI e o cliente. Ele fornece a base para gerenciar o relacionamento entre o provedor e o cliente. O SLA descreve o serviço e ele serve como um padrão para mensurar e ajustar a execução do serviço de TI. SLAs devem ser estabelecidos para todos os serviços que são fornecidos. O SLA deve incluir no mínimo (1) uma descrição simples dos serviços e produtos; (2) disponibilidade do serviço acordado; (3) tempo de resposta ao usuário, tempo de resposta e resolução de incidentes; e (4) responsabilidades do cliente e do provedor.

Um Acordo de Nível Operacional (*Operational Level Agreements – OLA*) é um acordo feito entre os departamentos internos da organização de TI para prover o serviço, e nele é detalhado o fornecimento de certos elementos do serviço.

Um Contrato de Apoio (*Underpinning Contracts – UC*) é um contrato feito com um provedor externo que define o fornecimentos de certos elementos necessários para a realização do serviço. É uma implementação externa de um OLA. Só é criado caso haja necessidade.

O Plano de Qualidade de Serviço (*Service Quality Plan – SQP*) contém os parâmetros dos processos de gerenciamento de serviço e operacional. O SLA estabelece o que deve ser entregue, ao passo que o SQP estabelece como fazer a entrega. Ele contém metas para cada processo em forma de Indicadores de Desempenho. Tais indicadores são derivados da SLR e documentados na Spec Sheets.

O Catálogo de Serviço é um documento que auxilia a organização a montar um perfil de si mesma e a se apresentar como um Provedor de Serviço de TI ao invés de uma simples implementadora e mantenedora de tecnologia. Esse catálogo descreve detalhadamente, na linguagem dos clientes, os serviços fornecidos pela organização juntamente com os níveis de serviço que a mesma é capaz de prover aos clientes. O Catálogo de Serviço ajuda a conduzir a expectativa do cliente, e desta forma, facilitar o processo de alinhamento entre clientes e provedores de serviço.

2.3.2. Objetivos e benefícios do processo de Gerenciamento de Nível de Serviço

O SLM é o nome dado ao processo de planejamento, coordenação, estabelecimento, cumprimento, monitoramento e reportagem dos SLAs e do desempenho da organização de TI. Ele é processo que liga as áreas de Suporte a Serviço e Entrega de Serviço do ITIL. Ele não funciona de forma isolada e depende da existência e de um trabalho eficiente e efetivo dos demais processos de gerenciamento.

O objetivo desse processo de gerenciamento é manter e gradualmente melhorar a qualidade do serviço de TI, através de um ciclo constante de negociação, monitoração, informação e revisão dos serviços realizados, além de observar as ações a fim de erradicar serviços de baixa qualidade. Dessa forma, um melhor relacionamento entre o provedor de serviço e seus clientes vai se estabelecendo.

As melhorias na qualidade do serviço e a redução da interrupção do mesmo levam a uma economia de custos. Com isso, menos tempo e esforço são gastos para resolver falhas e os clientes podem realizar seu negócio sem impactos adversos.

Alguns benefícios específicos de implantar o SLM são:

- Melhoria do relacionamento com os clientes e satisfação dos mesmos;

- Ambas as partes têm uma visão clara de seus papéis e responsabilidades, e dessa forma, são evitados mal-entendidos e omissões;
- A qualidade do serviço pode ser mensurada, monitorada e reportada;
- A monitoração e a revisão dos serviços permitem a identificação dos pontos fracos, para que possam ser realizadas as ações corretivas;
- Os serviços de TI são projetados para satisfazer as expectativas, como definido no SLR;
- SLAs podem ser usados como base para a cobrança dos serviços e para demonstrar o que os clientes estão recebendo de volta pelo dinheiro investido.

2.3.3. O processo do SLM

O Gerenciamento de Nível de Serviço é um processo que liga o provedor de serviço de TI e o cliente desse serviço. Ele forma uma ponte com o cliente e fornece uma oportunidade de discutir as necessidades do negócio sem assustá-los com detalhes técnicos. A organização então traduz essas necessidades do negócio em especificações técnicas e atividades dentro dela. Esse processo requer uma excelente cooperação com o cliente, visto que as definições dos níveis de serviço apropriados necessita da contribuição e esforço por parte do cliente.

O objetivo desse processo é: (1) integrar os elementos necessários para o fornecimento do serviço; (2) criar documentos que claramente descrevem o serviço tanto pelos elementos necessários, quanto na terminologia que os clientes entendem; e (3) alinhar a estratégia de TI com as necessidades do negócio.

A Figura 2-6 mostra as etapas do processo de SLM, incluindo as seguintes atividades presentes no processo, que são detalhadas a seguir.

Identificação

O objetivo desta atividade é ajudar o usuário a entender e definir os serviços e os níveis de serviço que ele deseja, e os custos necessários para isso. Essa atividade é importante dado que muitos clientes assumem que vários aspectos do serviço são fornecidos sem a necessidade de haver um acordo claro sobre isso, o que normalmente gera muita confusão.

O primeiro passo para a realização de um SLA, tanto no presente quanto no futuro, é identificar e definir o que o cliente necessita na forma de Requisitos de Nível de Serviço (SLR). Todos os requisitos do cliente devem ser expressos através de valores mensuráveis, de forma que tais valores possam contribuir para o projeto e a monitoração do serviço. Se essas métricas não forem estabelecidas, fica muito difícil verificar se os níveis de serviço estão sendo cumpridos ou não, segundo os acordos.

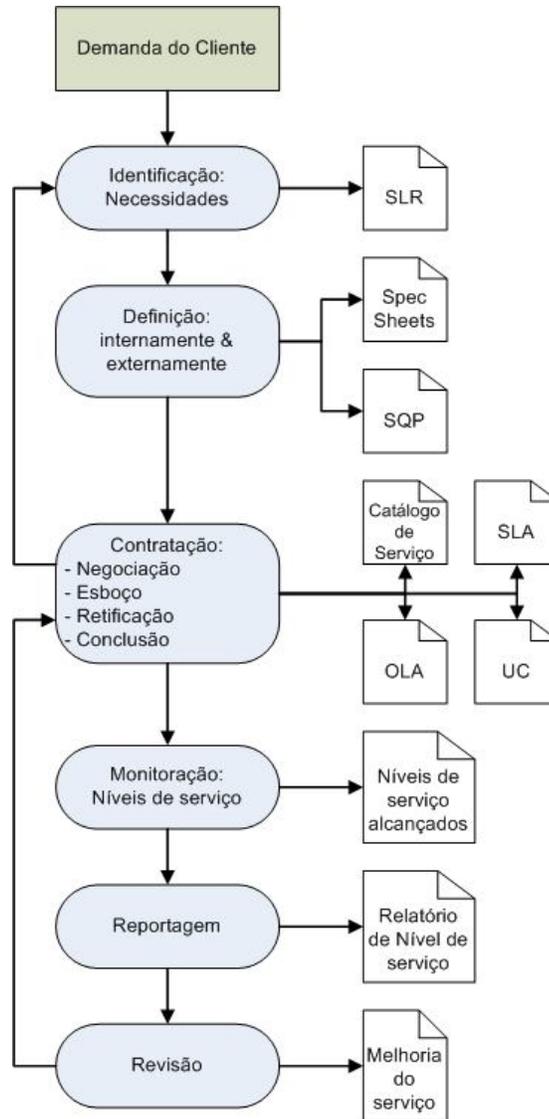


Figura 2-6 - Processo de Gerenciamento de Nível de Serviço [Bon et al, 2006]

Definição

A definição do escopo e detalhes dos requisitos do cliente é considerada um processo de *design* do SLM. Esse processo inclui vários passos que vão desde o detalhamento e definição dos requisitos do cliente através de padrões externos e internos claros, até o desenvolvimento dos requisitos técnicos para fornecer o serviço. É necessário definir de forma clara tanto os padrões externos, relacionado à comunicação com o cliente, quanto os padrões internos, que são os apoios técnicos dentro da organização.

O primeiro passo é definir ou redefinir as expectativas do cliente sobre o serviço em geral, no caso de ser um novo serviço ou um serviço existente, respectivamente. Essas expectativas são descritas no SLR. As seguintes informações são necessárias para definir o SLR:

- Uma descrição, segundo a perspectiva do cliente, das funções a serem fornecidas pelo serviço.
- Datas e horários nos quais o serviço deve estar disponível.
- Requisitos de continuidade do serviço, isto é, sua tolerância contra falhas graves.
- As funções de TI necessárias para o fornecimento do serviço.
- Referências aos métodos operacionais e aos padrões de qualidade a serem considerados na definição do serviço.

Durante a fase de definição, o SLR é criado em detalhes, e deve fornecer as seguintes informações:

- Descrição detalhada e não-ambígua do serviço e dos componentes necessários.
- Especificação da forma na qual o serviço deve ser implementado e fornecido.
- Especificação da qualidade desejada e dos procedimentos de verificação.

Durante a fase de definição do serviço, é importante separar os documentos de uso externo dos documentos de uso interno da organização, como ilustra a Figura 2-7.

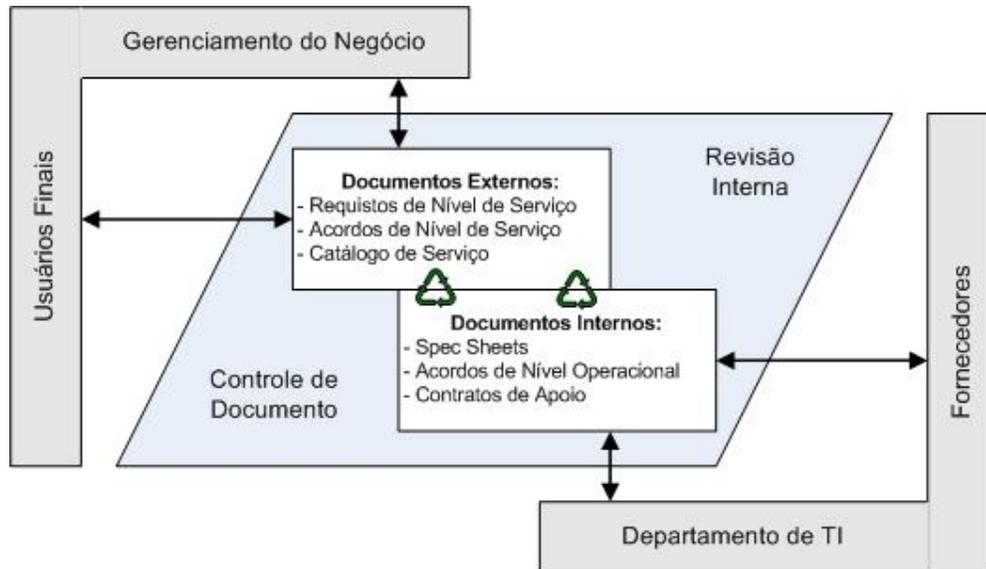


Figura 2-7 - Fase de Especificação [Bon et al, 2006]

Os documentos externos são acessíveis ao cliente, contêm os objetivos acordados com o cliente, e são as entradas dos documentos de uso interno. Esses últimos por sua vez, se referem aos objetivos internos da organização, que devem ser alcançados para satisfazer as demandas do cliente. O SLM está relacionado com o alinhamento entre as especificações internas e externas. O Controle de Documentos e a Revisão Interna contribuem para manter registros dos documentos e gerenciar versões.

Negociação/Contratação

Terminada a fase de especificação, ou seja, a tradução das necessidades do negócio em recursos e configurações da TI, essas informações podem ser usadas para estabelecer os SLA, OLA, UC e Catálogo de Serviço.

Para estabelecer o SLA, é necessário em primeiro estágio desenvolver um modelo geral de SLA geral baseado no serviço. O SLA pode ser organizado em uma estrutura hierárquica, sendo que cada camada dessa hierarquia possui seu próprio nível de detalhes. O número de camadas depende do que se deseja considerar, por exemplo, os aspectos físicos da organização (distribuição geográfica, complexidade, etc.), aspectos culturais (linguagem do documento, política de cobrança, etc.), e natureza das atividades do negócio (termos e condições gerais, carga horária do negócio - 5 x 8 ou 7 x 24, etc.).

Fechado o SLA inicia-se o estabelecimento das OLA's e UC's. Todos os envolvidos precisam estar cientes desses acordos para o fornecimento do serviço especificado.

O Catálogo de Serviço por sua vez, deve ser escrito usando a linguagem do cliente, evitando termos técnicos e relevantes ao negócio. O Catálogo de Serviço procura mostrar informações relevantes identificadas a partir do ponto de vista do usuário.

Monitoração

A monitoração do SLM só pode ser feita se os níveis de serviço forem claramente definidos com antecedência e corresponderem com os objetivos acordados. Os níveis de serviço devem ser medidos segundo a perspectiva do cliente e a monitoração não deve ficar limitada aos aspectos técnicos, mas incluir alguns procedimentos definidos entre o cliente e o provedor de serviço.

O Gerenciamento de Capacidade, de Disponibilidade e de Incidente desempenham um papel muito importante fornecendo informações técnicas que estão diretamente relacionadas com os níveis de serviço. Além dessas informações, outras como o tempo de resposta e de suporte devem ser medidos.

Reportagem

Relatórios periódicos devem ser fornecidos aos clientes para que se possa comparar os níveis que foram acordados e os níveis de serviço que foram medidos. O relatório deve incluir informações como: (1) a disponibilidade e a indisponibilidade ocorrida em um período específico, (2) média do tempo de resposta durante períodos de picos de utilização do serviço, (3) nível de capacidade do serviço utilizado, (4) custo do serviço fornecido, entre outras informações.

Revisão

Os níveis de serviço precisam ser revisados frequentemente, levando em conta os seguintes aspectos: (1) os SLAs desde a última revisão, (2) os problemas relacionados ao serviço, (3) mudanças no serviço, e (3) as conseqüências por não fornecer os níveis de serviço negociado.

Caso o serviço não esteja satisfazendo os níveis de serviço, ações devem ser providenciadas, por exemplo: (1) melhorias no serviço, e (2) modificações no SLA, OLA e UC.

2.3.4. Relacionamento com outros Processos ITIL

Para que se tenha um SLM efetivo, é necessário a contribuição dos outros processos de gerenciamento do ITIL. Todos os processos contribuem de alguma forma para o SLM e essa contribuição vai de acordo com o serviço e os níveis de serviço definidos. A seguir, é descrito os relacionamentos entre o SLM e alguns processos de gerenciamento do ITIL.

Service Desk

Embora o *Service Desk* seja uma função e não um processo, seu relacionamento com o SLM é muito importante. O *Service Desk* é o ponto de contato inicial para o usuário do serviço, e por causa disso, ele é capaz de fornecer informações importantes sobre a percepção da qualidade do SLM do ponto de vista do usuário. Ele é importante também para fornecer o tempo de resposta e solução sobre incidentes e falhas no serviço.

Gerenciamento de Disponibilidade

O Gerenciamento de Disponibilidade é responsável por manter e otimizar a disponibilidade dos serviços. O SLM fornece a este processo informações sobre a disponibilidade necessária para o funcionamento do serviço, enquanto que o Gerenciamento de Disponibilidade fornece sua atual capacidade ao SLM.

Gerenciamento de Capacidade

O Gerenciamento de Capacidade é responsável por gerenciar a capacidade de toda a infraestrutura de TI. Ele dá suporte ao SLM fornecendo informações sobre o impacto de um novo serviço ou a modificação de um serviço na capacidade geral da infraestrutura. Também indica se o uso feito sobre um serviço está dentro dos limites estabelecidos. O SLM por sua vez, diz ao Gerenciamento de Capacidade a capacidade atual e futura esperada para realizar os serviços.

Gerenciamento de Incidente e Problema

O Gerenciamento de Incidente e de Problema são bons indicadores de uma efetiva implementação dos acordos de nível de serviço. A resolução de incidentes e problemas é essencial para o fornecimento de serviços de alta qualidade.

Gerenciamento de Configuração

O Gerenciamento de Configuração é responsável por prover detalhes sobre os componentes e documentos relacionados ao serviço no Banco de Dados do Gerenciamento de Configuração (BDGC), e por fornecer informações a partir deste banco de dados. Portanto, a criação ou modificação de um serviço ou SLA afeta o BDGC. O BDGC é utilizado para determinar o impacto de um erro no serviço, e para checar os acordos sobre os tempos de resposta e resolução. O BDGC é usado também para permitir que o SLM reporte a qualidade do serviço fornecido.

2.3.5. Controle do Processo

O processo de SLM deve ser controlado a partir de um conjunto de fatores de sucesso e indicadores de desempenho, além de relatórios de gerenciamento. O sucesso do SLM depende de alguns fatores, como: (1) um Gerente que conheça a TI e uma infra-estrutura que dê suporte ao SLM; e (2) uma definição clara das tarefas e responsabilidades dentro do processo.

O relatório de gerenciamento fornece informações importantes para o controle interno de processo de SLM. Tais informações tratam dos níveis de serviço atualmente suportados, como: (1) número de SLAs concluídos; (2) quantidade de vezes que um SLA não foi concluído; (3) satisfação do cliente; (4) estatísticas sobre incidentes/problemas.

O processo de SLM precisa ser controlado por um gerente. Esse gerente do processo é responsável por: (1) criar e atualizar o Catálogo de Serviço; (2) definir e manter um processo de SLM efetivo, considerando SLAs, OLAs e UCs; (3) negociar, manter e concluir as SLAs, OLAs e UCs; (3) supervisionar o desempenho da organização e realizar melhorias quando necessário.

Apesar de toda a preocupação em desenvolver um processo de SLM efetivo, alguns problemas podem ocorrer. Alguns dos possíveis problemas que podem acontecer são: (1) dificuldade do cliente em especificar o SLR; (2) dificuldade em expressar as expectativas do cliente em termos mensuráveis, e os custos associados ao serviço; (3) garantir que os

objetivos sejam verificados e concretizáveis antes de concordar e se comprometer com eles; (4) monitorar, medir e informar os níveis de serviço realizados; (5) atribuição de autoridade insuficiente ao Gerente de Nível de Serviço para promover negociações e melhorias; (6) o SLA pode ser demasiado extenso, pouco conciso e sem foco no negócio; (7) os documentos e o processo de SLM podem deixar de ser um meio para a melhoria do relacionamento entre o provedor de serviço de TI e o cliente.

2.4. Considerações Finais do Capítulo

O processo de Gerenciamento de Nível de Serviço é fundamental para qualquer organização que queira estabelecer um bom relacionamento com seus clientes, fornecendo serviços com qualidade, que se encaixam nas necessidades dos clientes, e a custo efetivo. Ele é responsável por coordenar, estabelecer, cumprir, monitorar e reportar os acordos e a execução dos serviços, buscando manter, e se possível, melhorar a qualidade dos serviços, sendo, portanto, o processo de ligação entre as áreas de Suporte e Entrega de Serviço do ITIL. Por causa disso, esse processo não pode e nem deve funcionar de forma isolada, pois colabora para a execução dos demais processos ITIL, além de depender da existência e do trabalho eficiente e efetivo deles. Assim, um melhor nível de execução do SLM e da organização como um todo pode ser alcançado quando o domínio que o envolve for bem compreendido e compartilhado entre as outras partes que se relacionam com este processo, e isso pode ser alcançado com a construção de uma especificação formal deste domínio.

Apesar do ITIL estar em constante evolução, a construção de modelos formais de especificações dos domínios existentes nos processos ITIL ainda não é muito explorada. Tais especificações, se construídas de forma clara, precisa e completa, são capazes de eliminar inconsistências e contradições, facilitando o entendimento das boas práticas na sua utilização durante o gerenciamento de serviço. Além do mais, tais especificações também facilitam a melhoria das boas práticas e processos.

A construção de tais especificações pode ser apoiada pelo uso de ontologias, que seriam então usadas na formalização dos modelos conceituais dos processos ITIL e de seus domínios. Uma das grandes vantagens de se utilizar ontologias é que os modelos formais resultantes podem ser usados por máquinas de inferência automáticas. Componentes de gerenciamento de serviços de plataformas sensíveis ao contexto podem usar esses modelos para melhorar a qualidade durante suas atividades de publicação, descoberta, seleção e/ou composição de

serviços. O Capítulo 3 apresenta as ontologias defendendo seu emprego como técnica formal de modelagem conceitual.

3. Ontologia e Modelagem Conceitual

3.1. Ontologia: Definição e Breve Histórico

O crescente interesse por ontologias dentro da computação tem possibilitado a sua aplicação de diversas maneiras. Muitas vezes, ontologias são usadas para modelar (descrever) situações da realidade, outras para promover integração de sistemas. De forma geral, ontologias constituem uma ferramenta poderosa para ser usada na modelagem conceitual de domínios da realidade, com o intuito de dar suporte a tarefas como:

- Comunicação e aprendizagem do domínio: ontologias ajudam na comunicação acerca de um determinado conhecimento. Através delas, as pessoas podem raciocinar e entender o domínio do conhecimento, chegando a um consenso sobre os termos a serem usados nas interações.
- Representação e reuso do conhecimento: ontologias representam o conhecimento do domínio de forma explícita, em um alto grau de abstração, possuindo um potencial enorme de reuso. Esse conhecimento representado pode ser especializado para diferentes domínios ou conceituações de um domínio.
- Formalização do domínio: por ser uma especificação formal de um domínio, as ontologias eliminam contradições, inconsistências e ambigüidades envolvendo os conceitos, relações e restrições. Além do mais, permitem derivar novos conhecimentos através das máquinas de inferência e permitem serem verificadas e validadas automaticamente através de provadores de teoremas.

O termo *ontologia* teve sua origem na Filosofia, criado por Aristóteles. Ele e os filósofos, desde a Grécia Antiga, têm usado ontologias para descrever domínios naturais, isto é, as coisas naturais do mundo que os cercam, como os tipos de existência e as relações temporais. A palavra latina *ontologia* tem origem etimológica a partir da junção do prefixo *ont-* (derivado do verbo grego *enai* = ser), e do sufixo *-logia* (estudo), cuja tradução pode ser entendida como o estudo da existência ou dos tipos de coisas que existem [Guarino, 1998] [Guizzardi, 2005] [Chandrasekaran et al, 1999].

Segundo o dicionário Merriam-Webster Online [Merriam-Webster, 2008], a palavra *ontologia* possui dois significados:

1. Um ramo da metafísica que estuda a natureza e a relação dos seres;
2. Uma teoria particular sobre a natureza dos seres e os tipos de existências.

De acordo com a definição do dicionário, ontologia é um importante ramo da metafísica que estuda as entidades que existem em certo domínio da realidade e os relacionamentos que envolvem essas entidades, segundo uma visão deste domínio. Sendo uma teoria, as ontologias procuram categorizar, em taxonomias, tais entidades, suas propriedades e relacionamentos, definindo um conjunto de termos e axiomas que servem para estabelecer a semântica dos termos e restringir suas interpretações. Desta maneira, ontologias, segundo essa visão e a visão filosófica, se referem a um sistema particular de categorias, independente da linguagem utilizada, que aborda certa visão do mundo, e que, segundo [Guizzardi, 2007], se preocupa em estudar as características mais gerais da realidade e das entidades reais, ao invés de estudar entidades que se enquadram dentro de domínios específicos, como ocorre em várias disciplinas científicas como Biologia, Física e Química. As ontologias lidam com relações entre categorias, incluindo aquelas entre entidades que pertencem a diferentes domínios da Ciência e também as entidades reconhecidas pelo senso comum.

No início do século XX, o filósofo alemão Edmund Husserl introduziu o termo *Ontologia Formal*, fazendo uma analogia ao termo *Lógica Formal*. Essa se preocupa com as estruturas lógicas formais (ex.: verdade, validade, consistência) independente da veracidade, ao passo que aquela se preocupa com as estruturas ontológicas formais (ex.: teoria de relacionamentos todo-parte, tipos, instanciação, identidade, dependência e unidade, entre outras), que se aplica em todos os domínios. Em outras palavras, a Ontologia Formal se preocupa com os aspectos formais dos objetos, sem levar em consideração as características específicas desses objetos. Ela pode ser utilizada para o desenvolvimento de sistemas gerais de categorização e respectivos relacionamentos, que por sua vez, podem ser usados no desenvolvimento de teorias científicas e teorias específicas de um domínio do mundo.

O uso de ontologias cresceu tanto que ultrapassou os limites da Filosofia e alcançou os domínios da Ciência da Computação e de Sistemas de Informação, onde teve sua primeira aparição em 1967 com G. H. Mealy em um de seus trabalhos [Mealy, 1967]. Sua popularidade cresceu e seu foco se tornou mais restrito, abrangendo domínios menores. A partir daí, houve uma crescente aplicação de ontologias em vários trabalhos das diversas áreas da computação, desempenhando os mais variados papéis. O primeiro grande *boom* aconteceu

em meados da década de 90, na área de Inteligência Artificial (IA), com a necessidade de se ter uma representação, compartilhamento e reuso do conhecimento de um domínio. Dentro da área da IA, o termo ontologia é utilizado tanto como um conhecimento descrevendo algum domínio quanto um vocabulário de representação normalmente especializado para algum tipo de domínio ou assunto. Neste último caso, não é o vocabulário que qualifica a ontologia, mas sim a conceituação que os termos do vocabulário buscam capturar [Chandrasekaran et al, 1999].

A verdadeira explosão do uso de ontologias dentro da Ciência da Computação aconteceu mesmo no início deste século XXI, motivado pelo surgimento da Web Semântica e pelo importante papel que as ontologias desempenham nesta iniciativa [Berners-Lee et al, 2001]. Nesse escopo, existe um conjunto de linguagens, cujas descrições são baseadas em lógica, que podem ser utilizadas para representar as ontologias. *Resource Description Framework (RDF)* [RDF, 2004], *DARPA Agent Markup Language* junto com o *Ontology Inference Layer (DAML+OIL)* [DAML+OIL, 2001] e o *Web Ontology Language (OWL)* [OWL, 2004] são exemplos de linguagens de representação de ontologias para a Web Semântica. Tais linguagens, assim como outras para modelagem conceitual e representação de ontologias, como a *Unified Modeling Language (UML)* [OMG, 2007], são baseadas em meta-conceituações ontológicas simplificadas, sendo denominadas *lightweight ontology languages*.

[Guizzardi, 2007] destaca a diferença que existe entre o uso do termo ontologia na área de Sistemas de Informação e nas áreas de IA e Web Semântica. Na primeira, o significado do termo tem sido utilizado como na filosofia, ou seja, como um sistema de categorias independente de linguagem. Segundo [Guarino, 1998], a esta definição deve ser dado o nome de *conceituação*. Contrariamente, na maioria das outras áreas da computação, como Engenharia de Software (mais especificamente Engenharia de Domínio [Falbo, 2002b], IA e Web Semântica, as ontologias são tratadas como um artefato de engenharia projetado para um propósito específico, representado por uma linguagem específica. Neste caso, o termo é usado para designar o que Nicola Guarino denominou de *ontologia de domínio* [Guarino, 1998].

Assim, na visão filosófica, ontologia é o estudo da existência e dos tipos de existência e é referido como um sistema particular, independente de linguagem, que considera uma porção da realidade, cujo propósito é classificar as entidades desta realidade, definindo seu vocabulário e as formulações canônicas de suas teorias. Já na visão da computação, uma

ontologia de domínio é o estudo do que existe em um determinado universo, organizado em uma taxonomia de termos, possuindo definições e um conjunto de axiomas formais usados para criar novas relações e para restringir a interpretação dos termos segundo um sentido pretendido.

Nesta dissertação, ontologias são aplicadas de duas formas diferentes. A primeira como sendo um sistema de categorização para a formalização e a construção de um modelo conceitual para a biblioteca de gerenciamento de serviços do ITIL. Os modelos conceituais construídos descrevem o domínio do processo de Gerenciamento de Nível de Serviço. Já a segunda forma, as ontologias são utilizadas na definição de um modelo de implementação, utilizando uma linguagem específica de implementação, com o propósito de aplicar os conceitos capturados no modelo ontológico do processo de Nível de Serviço no Gerente de Serviço em uma plataforma de serviços sensíveis a contexto.

3.2. Metodologias de Desenvolvimento de Ontologias

O processo de construção de ontologias de domínio é conhecido como *Engenharia de Ontologia*. Entretanto, a construção de ontologias não é uma tarefa fácil, sendo necessária, então, a utilização de métodos e ferramentas que possam auxiliar nessa tarefa. Segundo [Falbo, 1998], a ausência de atividades padronizadas, ciclos de vida e métodos sistemáticos, assim como um conjunto de ferramentas, técnicas e critérios de qualidade, podem gerar, durante o desenvolvimento da ontologia, vários problemas, como inconsistência, falta de clareza e compromisso ontológico, entre outros, tornando o processo uma arte e não uma atividade de engenharia.

Como dito anteriormente, em muitas áreas da computação, incluindo IA e Engenharia de Software, o termo ontologia é geralmente usado como um artefato de engenharia projetado para propósitos específicos. Sendo um artefato, uma ontologia deve ser construída seguindo um processo sistemático semelhante ao processo tradicional usado na Engenharia de Software.

Nesta dissertação, a abordagem sistemática para a construção de ontologias SABiO (*Systematic Approach for Building Ontologies*) [Falbo, 1998] [Falbo, 2004] é adotada com o intuito de apoiar a construção das ontologias. O SABiO é uma abordagem que se baseia no esquema proposto por [Uschold & King, 1995], aperfeiçoando-a com novas características,

incluindo uma linguagem gráfica para expressar ontologias, uma classificação de axiomas e o uso de questões de competência [Grüninger & Fox, 1995].

A Figura 3-1 ilustra a abordagem do SABiO para o desenvolvimento de ontologias. As seguintes atividades compõem o processo de construção de ontologias proposto pelo SABiO: (i) identificação de propósito e especificação de requisitos, (ii) captura da ontologia, (iii) formalização da ontologia, (iv) integração com ontologias existentes, e (v) avaliação e documentação da ontologia.

A seguir, são detalhadas cada uma dessas atividades.

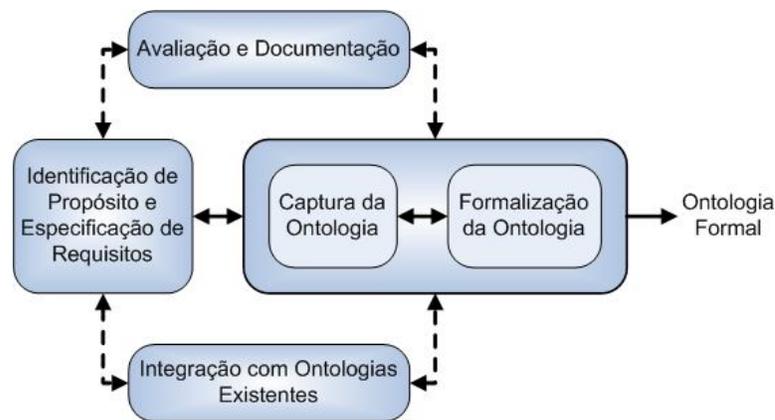


Figura 3-1 - Abordagem do SABiO para o desenvolvimento de ontologias [Falbo:1998]

3.2.1. Identificação de propósito e especificação de requisitos

Esta fase se preocupa em identificar, de forma clara, o propósito da ontologia e seu uso esperado, ou seja, busca definir as *competência da ontologia* [Grüninger & Fox, 1995]. A competência diz respeito àquilo que a ontologia é capaz de responder ou suportar, delimitando o que é relevante para ela e o que não é.

Para definir isso, são usadas as questões de competência [Grüninger & Fox, 1995], que tratam das questões que a ontologia deve ser capaz de responder. Essas questões são usadas para avaliar os compromissos de uma ontologia, verificando sua expressividade.

3.2.2. Captura da ontologia

É a etapa mais importante no desenvolvimento de uma ontologia e objetiva capturar a conceituação presente no domínio, tendo como referência a competência da ontologia, isto é, os propósitos e requisitos dela. As entidades relevantes do domínio, como conceitos, propriedades, relações, papéis, entre outros, devem ser identificados e organizados (em categorias e sub-categorias interconectadas) nesta etapa. Um modelo usando uma linguagem gráfica (ex.: perfil UML), junto com um dicionário de termos (consistente, claro e não-ambíguo), deve ser utilizado com o intuito de facilitar a comunicação com o especialista do domínio.

Além de identificar e organizar as entidades relevantes do domínio, que formam a base da ontologia, é necessário estabelecer os axiomas para definir a semântica dos termos do dicionário. Esses axiomas (epistemológicos, ontológicos e de consolidação) especificam definições de termos na ontologia e restringem sua interpretação [Falbo et al, 2002a]. Assim, ao se construir uma ontologia, é importante considerar axiomas que considerem a estruturação dos conceitos e relações (axiomas epistemológicos), seus significados e restrições (axiomas ontológicos) e as leis de integridade que os regem (axiomas de consolidação).

O processo de definição de axiomas não é simples, porém, ele pode ser guiado pelas questões de competência. Não deve haver axiomas a mais e nem a menos, ou seja, todos os axiomas devem ser necessários e suficientes para expressar as questões de competência e para caracterizar suas soluções. Ademais, as soluções para as questões de competência devem ser descritas por axiomas da ontologia, e ser consistentes com eles.

3.2.3. Formalização da ontologia

Nesta etapa, o objetivo é representar de forma explícita a conceituação capturada em uma linguagem formal, de forma que testes sobre a clareza e a correção de uma dedução possam ser efetuados com mais facilidade e precisão. Isso envolve o comprometimento com alguma meta-ontologia, a escolha de uma linguagem formal e a criação da ontologia formal. Essa linguagem deve ser capaz de representar de forma precisa e não-ambígua as entidades do domínio, além de permitir a escrita de axiomas formais para restringir a interpretação das estruturas formadas por tais entidades.

É importante verificar nesta etapa se já não existe algum formalismo disponível que possa ser usado diretamente ou adaptado. Caso não exista, é preciso então definir um formalismo para representar as entidades da ontologia, estabelecendo uma base para fixar a terminologia da ontologia e, principalmente a sua semântica.

3.2.4. Integração com ontologias existentes

O objetivo nesta fase é reutilizar conceituações previamente estabelecendo e integrá-las à ontologia que está sendo capturada e formalizada, visto que, durante o processo de construção da ontologia, isso pode se fazer necessário.

[Falbo et al, 2002a] sugere então que se desenvolva ontologias funcionais modulares, que sejam gerais e mais amplamente reutilizáveis, e, quando necessário, integrá-las, obtendo o resultado desejado.

3.2.5. Avaliação e documentação da ontologia

Toda ontologia deve ser avaliada para verificar se os propósitos e requisitos estão sendo realmente atendidos. Esta etapa pode ser realizada em paralelo com as etapas de captura e formalização da ontologia.

SABiO propõe que a ontologia seja avaliada com relação à sua competência e também com relação a alguns critérios de qualidade, como os propostos por Gruber (ex.: clareza, coerência, comprometimento ontológico mínimo, entre outros) [Gruber, 1995].

As questões de competência desempenham um papel muito importante na avaliação da completude da ontologia, principalmente no que se refere aos seus axiomas.

Finalmente, todo o processo de construção de uma ontologia deve ser documentado. Isso inclui os propósitos, requisitos, descrições textuais da conceituação, os modelos gráficos e a ontologia formal. Assim, como a avaliação, a documentação é uma atividade que deve ocorrer em paralelo com as demais.

3.3. Modelagem Conceitual e Linguagem

Modelagem conceitual é uma atividade muito importante na computação e desempenha um papel fundamental nas áreas de projeto de sistemas de informação, sistemas baseados em

conhecimento e banco de dados, engenharia de software, de domínio e de requisitos, integração de informação e interoperabilidade semântica, entre outras áreas. Ela não se preocupa inicialmente com o projeto do sistema, mas sim em descrever os aspectos físicos e sociais do mundo existente [Guizzardi, 2005].

[Guizzardi, 2005] destaca que um dos grandes fatores de sucesso no uso de alguma linguagem de modelagem se encontra na capacidade da mesma em fornecer aos seus usuários um conjunto de primitivas de modelagem capazes de expressar, de forma direta, os conceitos relevantes de um domínio, denominando a isso de *conceituação do domínio*. Defini-se *conceituação* como um conjunto de conceitos usados para articular abstrações de situações em um dado domínio [Guizzardi, 2005]. Por sua vez, as abstrações de uma parte da realidade, articulada segundo uma conceituação do domínio são denominadas de *abstrações do domínio* [Guizzardi, 2005]. Observe que para modelar certo fenômeno ou parte do mundo (domínio), é necessário focar em um número limitado de conceitos, suficientes e relevantes, para criar uma abstração do fenômeno em questão.

Para clarear tais definições, considere um domínio de atendimento de emergência para pessoas cardíacas em um hospital. Neste domínio, os seguintes conceitos fazem parte da conceituação: médico, enfermeiro, paciente, cirurgia, equipamento cirúrgico, Unidade de Tratamento Intensivo (UTI), realizar cirurgia em, recupera-se na, entre outros. A combinação desses conceitos pode gerar abstrações do domínio, como “O médico Joaquim realizou uma cirurgia no paciente Manuel. Manuel está se recuperando na UTI”.

É importante observar que as Conceituações e Abstrações são entidades não materiais que existem apenas na mente dos usuários de uma linguagem. Uma linguagem é o meio utilizado para representar, em termos de um vocabulário, as conceituações de uma realidade. Através dela então, é possível capturar, documentar, analisar e comunicar as Abstrações de um domínio de forma concisa, completa e não-ambígua, representando tais Abstrações em termos de algum artefato real [Guizzardi, 2005]. A Figura 3-2 exibe o *triângulo de Ullmann* [Ullmann, 1972], que representa o relacionamento entre uma linguagem, uma conceituação e uma porção da realidade (abstraído pela conceituação).

Nesta figura, a relação *abstrai* implica a abstração que uma conceituação faz sobre uma realidade. A relação *representa*, está relacionada com a definição da semântica da linguagem, ou seja, esta relação implica que os conceitos são representados pelos símbolos da linguagem.

Já a relação *refere-se a*, ilustrada pela linha pontilhada, entre a linguagem e a realidade estabelece que os relacionamentos entre as coisas de uma realidade e os símbolos de uma linguagem são sempre feitos, intermediados por uma conceituação.

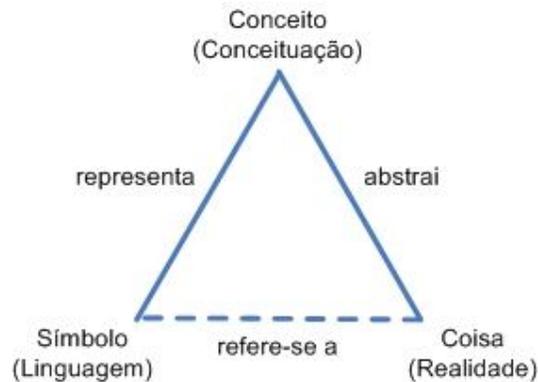


Figura 3-2 - Triângulo de Ullmann: relacionamento entre uma linguagem, uma conceituação e uma porção da realidade

A Figura 3-3 detalha melhor esta última relação, mostrando a diferença que existe entre Abstração e sua representação, e seus relacionamentos com uma Conceituação e linguagem de representação. A representação de uma abstração é chamada de *Modelo* (ou *Especificação*). Já a linguagem de representação usada para criar este modelo é chamada de *Linguagem de Modelagem* (ou *Especificação*).

A linguagem L é usada para representar a conceituação C de uma realidade, e para compor um modelo M . O modelo M , através da linguagem L , consegue representar uma abstração A , que por sua vez é uma instância da conceituação C . Então, segundo [Guizzardi, 2007], para que M represente fielmente A , as primitivas de modelagem de L usadas para produzir M , devem representar fielmente a C , usada para articular a abstração A representada.

A figura também mostra que o modelo, utilizando uma linguagem, corresponde à descrição do que as primitivas da linguagem são capazes de representar sobre uma conceituação. Portanto, uma especificação é um artefato concreto que corresponde a uma descrição formal e explícita da estrutura de uma conceituação de uma parte da realidade. A esse artefato concreto dá-se o nome de *ontologia de domínio*.

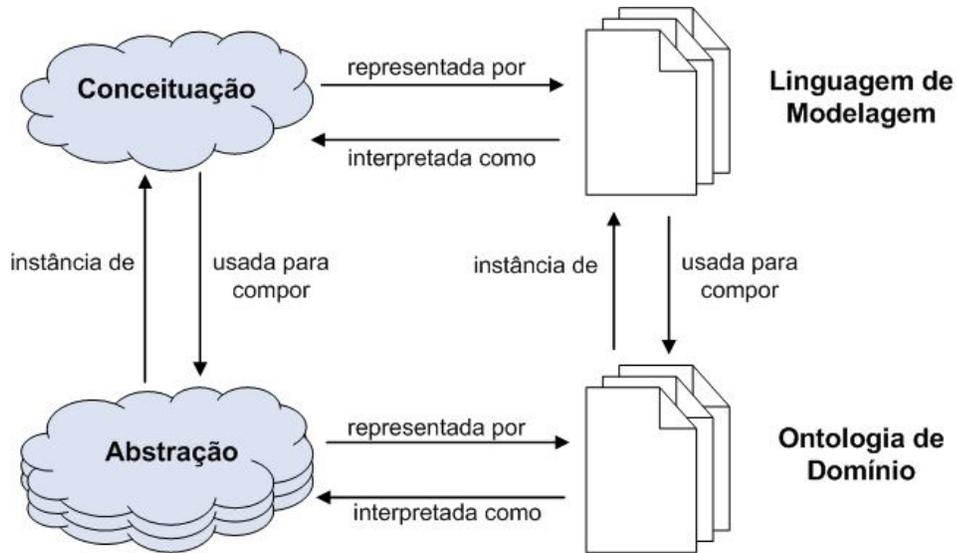


Figura 3-3 - Relacionamento entre conceituação, abstração, linguagem de modelagem e especificação [Guizzardi, 2005]

A qualidade de um sistema ou serviço baseado em uma ontologia depende da qualidade da mesma. Por sua vez, esta ontologia depende da qualidade da linguagem de modelagem usada para descrevê-la. A adequação de uma linguagem para descrever ontologias depende fortemente da semelhança que há entre a ontologia construída utilizando esta linguagem e as abstrações do domínio que ela visa representar. Caso a linguagem seja imprecisa e grosseira para a descrição de domínios, podem existir situações descritas por ela que, apesar de serem válidas gramaticalmente, não correspondem a estados possíveis da realidade. A Figura 3-4 ilustra esta situação.

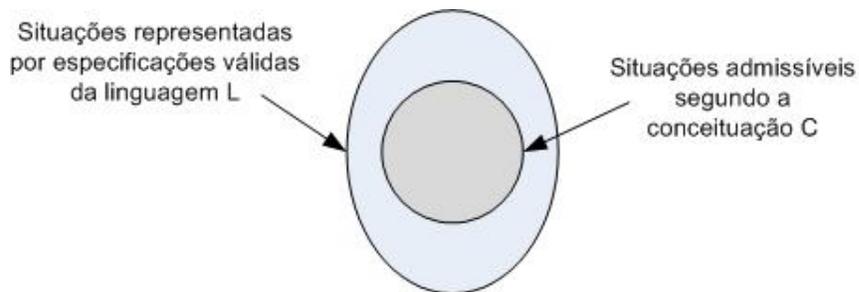


Figura 3-4 - Conseqüência de uma Linguagem de Modelagem imprecisa para a representação da conceituação de um domínio [Guizzardi, 2005]

A figura mostra o que [Krogstie, 2000] chamou de *adequação ao domínio* de uma linguagem de modelagem conceitual, que indica o grau de adequação da linguagem para modelar

fenômenos de certo domínio, ou seja, a confiabilidade da linguagem com relação ao domínio. Outro critério de qualidade para avaliar uma linguagem de modelagem, relatado por [Krogstie, 2000], é o de *adequação à compreensão*, que se refere ao grau de facilidade que um usuário da linguagem tem para reconhecer o significado dos construtores da linguagem com relação ao domínio, e a facilidade que ele tem para entender, comunicar e raciocinar sobre os modelos produzidos pela linguagem.

Portanto, uma linguagem de modelagem deve ter expressividade suficiente para representar de forma adequada a conceituação do domínio, e possuir uma semântica clara, de forma que seja fácil reconhecer o significado dos construtores da linguagem. Além do mais, o modelo produzido pela linguagem deve permitir o fácil entendimento e raciocínio do domínio representado.

[Gurr, 1999] e [Guizzardi, 2005] apresentam um conjunto de propriedades que devem ser usadas para avaliar a adequação de uma linguagem de modelagem e a ontologia O representando um domínio D . Talvez a mais importante delas seja a completude: “*Uma linguagem L é completa com relação a um domínio D se todo conceito na ontologia O daquele domínio é representado em uma primitiva de modelagem da linguagem*”. Isso porque a incompletude de uma linguagem gera a falta de expressividade por parte dela, ou seja, isso implica na existência de certas situações do domínio que não podem ser representadas pela linguagem [Guizzardi, 2005].

Existem diversas linguagens, como LINGO [Falbo et al, 1998] [Falbo et al, 2002a], RDFS [RDFS, 2004], OWL [OWL, 2004], UML [OMG, 2007], entre outras, que podem ser usadas para fazer a modelagem conceitual de ontologias sobre algum domínio. Uma questão que se levanta a partir daí é: qual linguagem utilizar para modelar certo domínio? A próxima seção aprofunda a discussão sobre essa questão.

3.4. Modelos de Especificação

Uma ontologia para um domínio precisa ser construída como sendo a melhor representação possível de uma conceituação. Um de seus propósitos consiste em ser uma representação mais próxima possível da realidade desse domínio. Nesse sentido, na escolha da linguagem de representação da conceituação de um domínio deve ser observada a expressividade dos modelos de especificação obtidos.

Este trabalho advoga a construção de modelos de especificação baseados em alguma ontologia fundacional (*upper-level ontology* ou *foundational ontology*), por exemplo a *Unified Foundational Ontology* (UFO), descrita adiante. Ontologias de fundamentação são capazes de expressar suficientemente os conceitos de um dado domínio, possuindo, portanto, as características de serem adequadas ao domínio e à compreensão. Essas características buscam compatibilizar a visão de mundo transcrita no modelo de especificação e a visão da realidade, entendida e aceita por especialistas do domínio.

As especificações construídas utilizando os conceitos da UFO são mais expressivas e exatas do que uma especificação representado pela linguagem UML, por exemplo. A vantagem de se ter especificações mais precisas e expressivas se encontra no compartilhamento, reuso, entendimento e aprendizado adequado do conhecimento do domínio, além da capacidade de operar semanticamente com outros sistemas. Entretanto, todas essas vantagens são conseguidas sacrificando-se a decidibilidade (resposta em tempo finito) e eficiência computacional. Essas características são muito importantes quando se deseja usar essas especificações em máquinas de processamento automático.

É importante destacar o que [Guizzardi, 2007] considera como a fase de mapeamento e codificação entre o modelo de especificação utilizando *upper-level ontologies* e o modelo escrito em termos de linguagens específicas, baseadas em *lighweight ontologies*, como DAML+OIL, RDF e OWL, cujos propósitos são garantir decidibilidade e um pouco de eficiência computacional. Essa fase consiste na derivação do modelo de especificação do domínio para o modelo de implementação.

A Figura 3-5 mostra o relacionamento entre uma abstração e seu modelo de especificação. Além disso, ilustra que esse modelo de especificação (ME) pode ser utilizado para derivar modelos de implementação (MI), em diversas linguagens específicas, que podem ser usadas por máquinas de processamento automático. As relações “representa” estabelece que um MI é uma representação de uma abstração. Todavia, devido à expressividade de cada linguagem, cada MI tem um grau de precisão diferente em relação aos conceitos do domínio modelado.

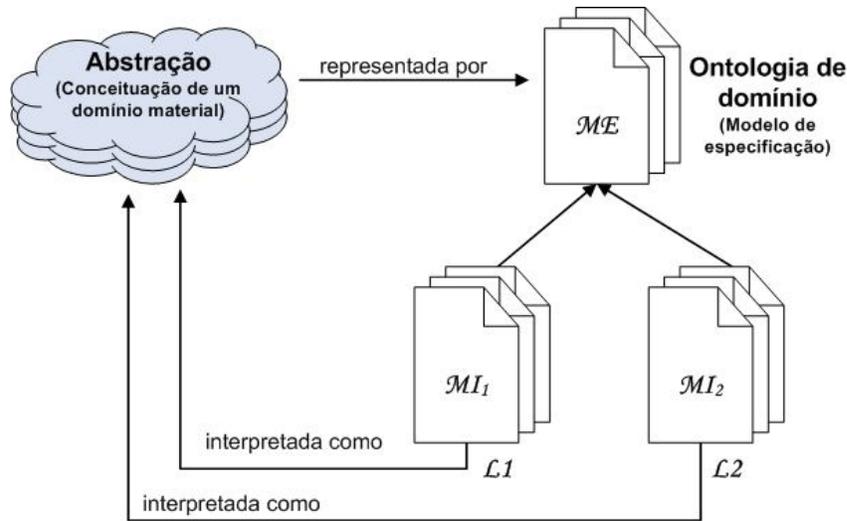


Figura 3-5 - Relacionamento entre ontologia de domínio e modelos de implementação [Guizzardi, 2005]

3.5. Ontologias de Fundamentação

Ontologia Fundacional (*Foundational Ontologies*) é um sistema de categorias e relações formais construídas a partir de teorias independentes de domínio que agregam contribuições de áreas como metafísica descritiva, ciência cognitiva, lingüística, filosofia da linguagem e outros [Guizzardi, 2005] [Guizzardi, 2006a].

Uma ontologia fundacional pode ser usada para definir primitivas de modelagem que expressam semânticas do mundo real, e para restringir as possíveis interpretações dessas primitivas, reduzindo assim ambigüidades e aumentando a clareza das interpretações. Com isso, é possível construir linguagens de caráter geral, baseados em metamodelos expressivos, para a representação ontológica de conhecimentos [Guizzardi, 2005]. Além do mais, o uso de ontologia fundacional é capaz de solucionar alguns problemas de interoperabilidade semântica que normalmente aparecem ao representar o conhecimento utilizando linguagens de *lightweight ontologies*. Esses problemas ocorrem muito em aplicações para a Web Semântica, como relatado em [Rios, 2003] e [Guizzardi, 2006a].

A *Unified Foundational Ontology* (UFO) é uma ontologia fundacional desenvolvida em [Guizzardi, 2005], baseada na OntoClean [Guarino & Welty, 2004] e na *Generalized Formalized Ontology* (GFO) [Heller & Herre, 2004]. A UFO é organizada em três conjuntos de ontologias incrementalmente estratificadas:

- 1) **UFO-A:** ontologia de endurantes (coisas, objetos, propriedades, entre outros). Constitui o núcleo da UFO e define os principais conceitos da ontologia.
- 2) **UFO-B:** ontologia de perdurantes (ocorrência, eventos, processos, entre outros). Criada sobre a UFO-A, define os termos relacionados a eventos.
- 3) **UFO-C:** ontologia de entidades sociais (planos, ações, objetivos, agentes, intencionalidade, entre outros). Construída sobre a UFO-A e UFO-B. Define os termos relacionados a coisas intencionais e sociais.

Nesta seção são descritos apenas alguns dos conceitos definidos pela UFO, conceitos esses que são importantes para o entendimento geral da ontologia e dos modelos construídos no próximo capítulo, referentes ao gerenciamento de nível de serviço do padrão ITIL. Maiores detalhes sobre a UFO podem ser encontrados em [Guizzardi, 2005] [Guizzardi & Wagner, 2008] [Guizzardi et al, 2008] [Guizzardi & Guizzardi, 2008]. A descrição da UFO feita nas próximas seções utiliza linguagem natural e é ilustrada usando o diagrama de classes da UML, com o intuito de auxiliar na visualização dos conceitos.

3.5.1. UFO-A: Uma ontologia de endurantes

A UFO-A, ilustrada na Figura 3-6, é uma ontologia de endurante, ou seja, aborda os conceitos de objetos, propriedades dos objetos e relacionamentos entre os objetos. O que caracteriza os endurantes é que eles não possuem partes temporais e persiste ao longo do tempo mantendo sua identidade. Toda vez que um endurante está presente, todas suas partes temporais também estão presentes. A base desta ontologia se encontra nos pares de categorias *Substantial-Substantial Universal (Object-Object Universal)* e *Moment-Moment Universal*.

Uma característica importante da UFO-A está na distinção entre as categorias *Particular* (ou *Individuals*) e *Universal* (ou *Type*). Os primeiros são entidades que existem na realidade e que possuem uma identidade única (ex.: o carro de José, o irmão de Maria, o teclado que usei para escrever este trabalho). Os últimos, por sua vez, são entidades que possuem padrões independentes espaço-tempo de características (ex.: uma pessoa, um carro, um teclado, uma casa). Os *universals* podem ser instanciados em vários *particulars*.

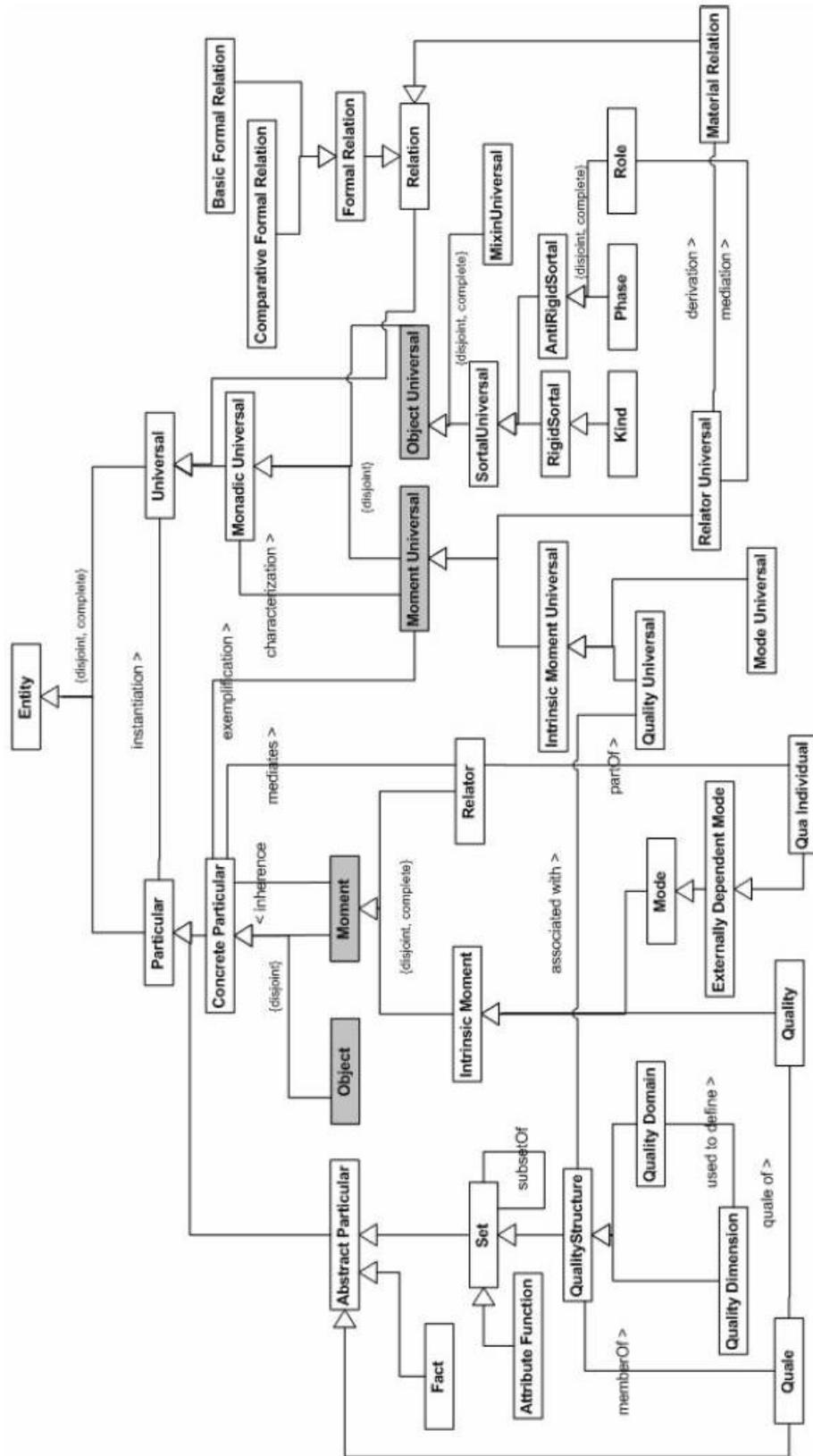


Figura 3-6 - Um fragmento da UFO-A [Guizzardi, 2008]

Na UFO-A, um *Moment* significa uma propriedade de uma instância (de um *individual*). Alguns exemplos de *moments* são: uma cor, uma massa, uma forma geométrica, um sintoma, uma idade. É importante observar que todos os *moments* só existem em outros *individuals*, isto é, a idade de uma pessoa só pode existir se existir uma pessoa, a massa e a cor de uma maçã só existem se a maçã existir. Daí, pode-se dizer que os *moments individuals* são existencialmente dependentes em outros *individuals*.

Um *individual* x é existencialmente dependente (abreviado por [Guizzardi, 2005] de *ed*) de outro *individual* y , se só se, para x existir, necessariamente y deve existir. E isso deve acontecer em todos os mundos possíveis. Este conceito de dependência existencial pode ser usado para diferenciar *Intrinsic Moments* de *Relational Moments*. Os primeiros dependem apenas de um único *individual* (ex.: uma temperatura, uma massa, uma idade). Já os *relational moments* (ou *Relators*) dependem de um conjunto de *individuals* (ex.: um tratamento médico, um emprego, um casamento). Existem também os *moments externamente dependentes* (*externally dependent moments*), que são *intrinsic moments* dependentes de um indivíduo único, mas que também são existencialmente dependentes de outros indivíduos: um *moment* x é externamente dependente se só se ele for existencialmente dependente de um indivíduo que seja independente do portador do *moment*.

Como exemplo, considere as responsabilidades legais que um homem adquire ao casar com uma mulher. Esses novos *moments* adquiridos pelo homem são existencialmente dependentes dele e são externamente dependentes da mulher. Outro tipo especial de relação de dependência existencial que há entre um *moment* x e um *individual* y na qual x depende de y é a relação de inerência (*inherence* - *i*). Assim, para que um *individual* x seja um *moment* de um outro *individual* y , a relação $i(x,y)$, x é inerente a y , deve existir entre os dois. Inerência é uma relação irreflexiva, assimétrica e intrasitiva entre *moments* e outros tipos de endurantes. A UFO-A admite que *moments* sejam inerentes a outros *moments*. *Individuals* que não são inerentes a outros *individuals* são chamados de *Substantials* (*Objects*).

Substantials (*Objects*) são *individuals* que possuem qualidades espaciais-temporais e que são formados por matéria. Como exemplo tem-se uma pessoa individual, uma montanha, um carro, uma sala de estar de uma casa, uma caixa postal. Eles são endurantes que não são inerentes a nenhum outro endurante, e com isso possuem o mais alto grau de independência.

Um *substantial* x é independente de todos os *substantials* que são desconexos a ele (que não compartilham de uma parte comum de x). Isso não é válido para a dependência entre um *substantial* e suas partes essenciais e inseparáveis, e a dependência entre um *substantial* e seus *moments* essenciais.

Para terminar de definir a base da UFO-A, existem as categorias de *Substantial Universal* (*Object Universal*) e *Moment Universal*. O termo *universal* é usado em um sentido mais amplo, não fazendo alusão, à princípio, a nenhuma teoria específica de universais. Um *universal*, é considerado como algo que (i) pode ser predicado de outras entidades e (ii) pode ser representado em uma linguagem por termos predicativos. Também é utilizada nesta ontologia a relação de classificação (ou a relação inversa de instanciação) entre *invididuals* (*particulars*) e *universals*. *Substantial universals* classificam *substantials* e *moments universais* classificam *moments*. Exemplos de *substantial universals* incluem Carro, Planeta e Pessoa. Exemplos de *moments universais* incluem Cor, Massa e Idade.

Dentro da categoria de *substantial universals*, existe uma diferença muito importante entre as categorias de *Sortal Universal* e *Mixin Universal*. Todos os *universals* possuem um princípio de aplicação (princípio que permite verificar se um *individual* é uma instância de um determinado *universal*), mas apenas os *sortals* possuem um princípio de identidade para suas instâncias. Este princípio permite verificar se dois *individuals* são o mesmo *individual* ou não.

Dentro dos *sortal universals*, existe a distinção entre os *RigidSortal* e os *AntiRigidSortal*. Essa distinção é baseada na noção de rigidez e anti-rigidez. Um *universal* U é dito rígido se para cada x , instância de U , x deve ser necessariamente uma instância de U em todos os mundos possíveis. Por outro lado, um *universal* U é dito anti-rígido se para cada x instância de U , x pode deixar de ser uma instância de U , ou seja, existe um mundo w no qual x não é instância de U . Todo *sortal* rígido é denominado de *Kind*, ao passo que, os *sortais* anti-rígidos são chamados de *Phased-Sortals*. Para exemplificar isso, considere um *kind* Pessoa e um *phased-sortal* Paciente e Criança instanciado pelo *individual* Joaquim. É fácil perceber que Joaquim pode deixar de ser um Paciente ou uma Criança (existem circunstâncias na qual ele não é), mas ele não pode deixar de ser uma Pessoa. Percebe-se também que, o fato de Joaquim instanciar Paciente e Criança não afeta a sua identidade, porém se ele deixar de instanciar o *universal* Pessoa, ele deixa também de existir como um *individual*. Existem algumas

restrições que envolvem estas categorias, a listar: (i) um universal rígido não pode ser subclasse de um anti-rígido; (ii) todo *substantial* deve instanciar um e apenas um *kind*.

É importante destacar que, para cada *phased-sortal* *PS* existe um *kind* *K*, tal que: (i) *PS* é uma especialização de *K*; (ii) *K* fornece o princípio de identidade único para as instâncias de *PS*. Além do mais, existe uma condição de especialização *ce* tal que, se *x* é instância de *PS*, então ele é instância de *K* e satisfaz a condição *ce*, e vice-versa. A partir daí, da condição de especialização, é possível fazer a distinção entre *Phases* e *Roles* (subcategorias de *AntiRigidSortal*). Os primeiros constituem os possíveis estágios na história de um indivíduo (ex.: Vivo e Morto como possíveis estágios de Pessoa; Cidadela, Metrópole como estágios de Cidade), e a condição de especialização depende apenas dos *intrinsic moments* dos *phases*. Já nos *roles*, a condição de especialização depende dos *relational moments*. Um *role* é um papel desempenhado por uma entidade em um determinado contexto, e esse papel é demarcado pelos relacionamentos com outras entidades. Por exemplo, se Joaquim é uma pessoa que está na fase (*phase*) viva, então ele é uma Pessoa que tem o *intrinsic moment* de estar vivo. Agora, se Joaquim desempenha o papel (*role*) de Paciente, então ele é uma Pessoa que está sendo tratada em alguma unidade de saúde.

Segundo [Welty & Guarino, 2001], o tipo *Mixin* é muito importante na especificação de modelos conceituais, pois representam os tipos mais abstratos como Coisa (*Thing*) e Entidade (*Entity*), mas também conceitos como *Category*, que é um *substantial* rígido que representa uma abstração de propriedades importantes que são comuns a todas as instâncias de vários *kinds* disjuntos. Em outras palavras, um tipo *category* é um *mixin* rígido que é supertipo de vários *kinds*. Existem também alguns *mixins* que são anti-rígidos e representam abstrações de propriedades comuns a múltiplos *roles* disjuntos. A esse tipo de *mixin* é dado o nome de *RoleMixin*, que são tipos não-sortais, anti-rígidos e dependentes. Ainda existe um terceiro tipo de *mixin*, neste caso um não-sortal não-rígido denominado *Mixin*, que representa propriedades essenciais a algumas de suas instâncias e acidentais a outras. A UFO-A impõe algumas restrições ao tipo *mixin* apresentado, a listar: (i) *mixins* não podem ser subtipos de algum *sortal*; (ii) o tipo *category* (*mixin* rígido) não pode ter como supertipo um tipo anti-rígido (ex.: um *role mixins*); (iii) todos os *mixins* (*category*, *role mixin* e *mixin*) não possuem instâncias.

A teoria de espaços conceituais [Gärdenfors, 2000] apresenta a relação entre *intrinsic moments* e sua representação segundo a percepção humana. Ela utiliza a idéia de *estrutura de*

qualidade (quality structure). Para diversos *moments universals* percebíveis ou concebíveis, há uma estrutura de qualidade associada segundo a percepção humana. Por exemplo, comprimento e massa são associadas a estruturas unidimensionais isomórficas à semireta dos números inteiros não-negativos. Outras propriedades, como cor, são representadas por estruturas multidimensionais. A percepção ou o conceito de um *intrinsic moment* pode ser representado como um ponto em uma estrutura da qualidade. Esse ponto é chamado de *quale*. Estruturas de qualidade e *quales* são exemplos de *Moments*.

Relações (*Relations*) são entidades que conectam outras entidades. Existem dois tipos de relações: relações formais (*formal relations*) e relações materiais (*material relations*). As primeiras conectam diretamente duas ou mais entidades, sem a necessidade de haver uma terceira entidade intervindo. Essa relação, também chamada de *relação interna* [Schneider, 2002] inclui a dependência existencial (*ed*), inerência (*i*), parte-de (<), subconjunto-de, instanciação, caracterização, exemplificação, entre outras. As relações de comparação (ex.: mais alto que, mais leve que, mais velho que) também são consideradas relações formais, mas neste caso, as entidades conectadas não são substantiais, e sim *intrinsic moments*.

As relações materiais, por outro lado, possuem uma estrutura material em si próprias e incluem exemplos tais como “trabalhar em”, “sendo tratado em”. Uma relação material entre duas ou mais entidades só existe, se houver uma outra entidade mediadora que relacione as entidades envolvidas. Por exemplo, a relação “sendo tratado por” entre Joaquim e uma Unidade de Saúde só existe, se houver uma entidade mediadora “tratamento” que conecta Joaquim à Unidade de Saúde. As entidades mediadoras são chamadas de *relators* e possuem a capacidade de conectar entidades. Um contrato conecta um empregado a uma empresa, e uma matrícula conecta um estudante a uma instituição de ensino. *Relators* desempenham um papel muito importante dentro da UFO.

Como dito anteriormente, um *moment* é externamente dependente se só se ele for existencialmente dependente de um indivíduo que seja independente do portador do *moment*. Voltando ao exemplo dado do casamento entre um homem e uma mulher, é possível definir um *individual (particular)* “marido” que é portador de todos os *moments* externamente dependentes desse homem casado. Este particular recebe o nome de *qua individual*. *Qua individuals* são um tipo especial de *moments* complexos externamente dependentes. Os

moments complexos inerentes ao homem que carregam todas as responsabilidades que este homem adquire ao casar com uma mulher pode ser chamado de *homem-qua-marido*.

Por fim, tem-se o conceito de *Situation* [Heller & Herre, 2004], um tipo especial de *endurante*, que é uma entidade complexa constituída de outros *endurantes* (inclusive outras *situations*), ou seja, *endurantes* estão presentes em *situations*. Esse conceito é um sinônimo de “*state of affair*”, que significa uma parcela da realidade que pode ser entendido como um todo. Para maiores detalhes sobre *situations* consulte [Dockhorn et al, 2006].

3.5.2. UFO-B: Uma ontologia de perdurantes

A UFO-B é uma ontologia de *perdurantes*. A diferença que existe entre os *individuals* *endurantes* e *perdurantes* está ligada com o comportamento que estes *individuals* apresentam com relação ao *tempo*. Intuitivamente, essa distinção pode ser entendida pela diferença entre “objetos” e “processos”, respectivamente. Como dito anteriormente, os *endurantes* não possuem partes temporais e persiste ao longo do tempo mantendo sua identidade. Toda vez que um *endurante* está presente, todas suas partes temporais também estão presentes. Os *perdurantes*, por outro lado, são compostos por partes temporais, e eles “acontecem no tempo”, no sentido de que eles se estendem no tempo acumulando partes temporais. Exemplos de *perdurants* incluem: uma corrida, uma conversa, uma partida de tênis, uma festa de aniversário e um processo de negócio. Sempre que um *perdurante* estiver presente não é necessário que todas as suas partes temporais estejam presentes. Por causa disso, eles não podem apresentar mudanças, genuinamente falando, ao longo do tempo, dado que nenhuma de suas partes temporais mantém o princípio de identidade durante esse tempo.

A Figura 3-7 apresenta uma parte da ontologia de *perdurantes*, a UFO-B. A principal categoria desta ontologia é *Event* (ou *Perdurant, Occurent*), que pode ser dividida em *Atomic* ou *Complex*, dependendo da estrutura deste *event*. *Atomic Events* não podem ser subdivididos em outros *events*, ao passo que *Complex Events* são dois ou mais *events* (*atomic* ou *complex*) agregados. *Events* podem alterar a realidade, mudando as características de uma situação (pré-estado) para outra situação (pós-estado). *Events* são entidades existencialmente dependentes de seus participantes. Por exemplo, considere um *event* que representa uma cirurgia de transplante de coração. Participam desse *event* a equipe médica responsável pela cirurgia, o paciente que receberá o coração e o próprio coração. Sem esses participantes, o *event* não existe. Este *event* é composto da participação (*Participation*) individual de cada uma das

entidades (equipe médica, paciente e coração), portanto existencialmente dependente desses *participations*, sendo que cada *participation* é um *event* (*atomic* ou *complex*) e existencialmente dependente de algum *substantial* (*object*).

A UFO-B define as propriedades espaciais dos *events* em termos das propriedades espaciais de seus participantes. Por outro lado, as propriedades temporais dos *substantials* são definidas em termos dos *events* nos quais participam. O valor (*quale*) dessas propriedades temporais é obtido projetando tais propriedades em uma estrutura de qualidade (*quality structure*). A estrutura de qualidade de “tempo” é composta por intervalos de tempo (*Time Intervals*), que por sua vez, são compostos por pontos de tempo (*Time Points*). Estes podem ser representados como números reais e os *Time Intervals* como conjuntos de números reais. Tais intervalos podem ser: (i) delimitados por pontos de início e fim, ou abertos; (ii) contínuo ou não-contínuo; (iii) com ou sem duração.

3.5.3. UFO-C: Uma ontologia de entidades sociais

A UFO-C, mostrada na Figura 3-8, é uma ontologia de entidades sociais (tanto endurantes quanto perdurantes), construída sobre a UFO-A e UFO-B. A sua base é formada pelos conceitos de *substantial individual* (*object*) e *moment individual* (*moment*) vindos da UFO-A, e pelo conceito de *event* vindo da UFO-B.

A partir do conceito de *substantial individual* é possível estabelecer a diferença entre *Agents* e *Objects*. *Objects* podem ser classificados como *Resource* (*Non-agentive objects*). Neste caso, eles são utilizados pelos *agents* para um propósito específico e normalmente são controlados por um algum *agent*. Exemplos de *resource* incluem um computador, um estetoscópio, uma ambulância, entre outros.

Objects também podem ser categorizados como *Social Object*. Dentre os exemplos de *social objects* tem-se o dinheiro, uma linguagem, umas *descrições normativas (normative descriptions)*. Uma *normative description* define uma ou mais regras (ou normas) reconhecidas por pelo menos um *social agent* e é capaz de definir *universals* nominais, como *Social Moment* (ex.: *social commitment*), *social objects* (ex.: a coroa de um rei) e *social roles* (ex.: um presidente, um candidato a um cargo profissional, um pedestre). Exemplos de descrições normativas incluem a Constituição Brasileira, o estatuto de alguma federação, ou um conjunto de diretrizes orientadoras em como executar algumas ações dentro de uma organização. Descrições normativas pode ser melhor entendido em [Bottazzi & Ferrario, 2008].

Agents também podem ser entidades físicas (uma pessoa, por exemplo) ou entidades sociais (uma organização, por exemplo). Os agentes físicos por sua vez podem ser categorizados como agente biológico ou humano (*Human Agent*), agente artificial (*Artificial Agent*), ou agente institucional (*Institutional Agent*). Os primeiros são *agents* representados por seres humanos, os segundos por entidades computacionais (*software* normalmente), e os últimos por organizações e unidades de uma organização (como departamentos, divisões e áreas). Os *institutional agents* são compostos por diversos outros *agents*, que podem ser eles mesmos humanos, artificiais ou institucionais.

Agents são *substantials* capazes de ter tipos especiais de *moments* chamados de *Intentional Moments*. A intencionalidade é vista como a capacidade que algumas propriedades de certos *individuals* têm para referenciar algumas possíveis situações da realidade. Cada *intentional moment* possui um conteúdo proposicional (*Proposition*), que é uma representação abstrata de um conjunto de situações referenciadas pelo *intentional moment*. O conteúdo proposicional de uma *intention* é um objetivo (*Goal*). *Situations* podem satisfazer (logicamente falando) o conteúdo proposicional de um *intentional moment*.

Além do mais, cada *intentional moment* pode ser especializado em um *Mental Moment*, estabelecendo que um *intentional moment* é existencialmente dependente de um *agent*, sendo uma parte inseparável do seu estado mental. Pode concluir que um *mental moment* é inerente a um agente físico. Exemplos de *mental moments* incluem uma Crença (*Belief*), um Desejo (*Desire*) ou uma Intenção (*Intention*). Um *belief* é justificado por situações na realidade (*Situations*). Exemplos incluem a crença na imortalidade da alma e a crença de que o planeta

Terra gira em torno do Sol. *Desires* expressam vontades de um *agent* sobre a realização de certas situações (ex.: desejo de ganhar na megasena), e *intentions* (*internal commitment*) são situações desejáveis que o *agent* se compromete em perseguir (ex.: a intenção de passar um final de semana em Guarapari). Ambos se referem a um *goal* do *agent*, e ambos podem ser cumpridos (realizados) ou frustrados. As *intentions* fazem com que um *agent* (e somente ele) execute ações (*Actions*), ou seja, *actions* são causadas pelas *intentions* dos *agents*.

Actions são eventos intencionais, portanto instanciam um *Plan*, também considerado pela UFO-C como um *Action Universal*. Elas visam satisfazer a *proposition* de alguma *intention*, por exemplo, a compra de um carro, a doação de sangue, um processo de negócio e até um ato de se comunicar (*communicative act*). Um *communicative act* (ato de discurso) é uma *atomic action*. *Actions* também podem ser categorizadas como *atomic* ou *complex*, assim como os *events*. A idéia é semelhante, isto é, uma *complex action* agrega duas ou mais *participation*. Essas *participations* podem ser intencionais ou eventos não-intencionais. Nem toda participação de um *agent* é considerada uma *action*, mas somente aquelas participações intencionais, chamadas de *Action Contributions*.

Como comentado anteriormente, somente os *agents* podem executar *actions*. Os objetos físicos que participam de uma ação são chamados de *Resources*. Uma *complex action* composta de *action contributions* de diferentes *agents* é denominada *Interaction*. O conjunto de médicos e enfermeiros que trabalham juntos durante a realização de uma cirurgia de transplante de coração, é um exemplo de *interaction*. Neste caso, a sala, os equipamentos, o coração são exemplos de *resources*, entre outros. Na UFO-C existem quatro formas de *Resource Participation*, apesar de existirem outras formas. São elas: *Creation*, *Termination*, *Change* e *Usage*. A definição formal para elas encontra-se em [Guizzardi et al, 2008].

Uma *resource participation* pode ser a causa de uma *resource dependence* e o resultado de uma *resource acquisition* entre *agents*. Em uma *resource acquisition*, o agente *A* dá uma permissão de resource *r* ao agente *B*. Para que isso aconteça, *A* deve ter o direito de conceder a permissão ao agente *B* e, além disso, a possibilidade de conceder o modo correto de permissão (por exemplo, permissão de usar ou modificar). A Figura 3-9 mostra a diferença entre uma *Dependency* e uma *Delegation*, além de mostrar o relacionamento delas com uma *resource acquisition*.

G em nome de A. Existem dois tipos de *delegations*: aberta e fechada. Um *Goal Delegation* é um tipo de delegação aberta, isto é, a decisão a respeito da estratégia para a realização do *goal* cabe ao *depender*. Já um *plan* descreve uma estratégia específica (definida pelo plano), que o *depender* deve adotar para a realização do *goal*, sendo categorizado então, como uma delegação fechada.

Como visto anteriormente, *communicative acts* são *atomic events*, que podem ser usados para criar *Social Moments*. *Social moments* são tipos de *intentional moments* criados pela troca de *communicative acts* e pelas conseqüências dessas trocas (ex.: adoção de um *goal* ou uma *delegation*). Por exemplo, considere a compra de um apartamento em um edifício que está sendo construído. Ao assinar um acordo de negócio, um *agent* (construtora) executa um *communicative act* (uma promessa). Este ato cria um *Social Commitment* da construtora em relação ao cliente, que é o compromisso de construir e entregar o apartamento na data prevista, em perfeitas condições de uso e de acordo com as especificações da planta, etc. (o conteúdo proposicional). Além do mais, o *communicative act* cria uma *Social Claim* do cliente para a construtora, com relação ao conteúdo proposicional. *Social Relator* é uma material relation composta de dois ou mais pares de compromissos/reivindicações associados (*social moments*).

Finalmente, um *Commitment* (interno ou social) é cumprido (*Fulfilled*) ou Não-Cumprido (*Unfulfilled*). *Unfulfilled commitments* podem ser categorizados como *Pending*, *Dismissed*, ou *Broken*. Um *social commitment* necessariamente conduz para a criação de um *internal commitment*, que por sua vez conduz um *agent* para a realização de uma *action*. Um *Commitment* (interno ou social) é cumprido por um *agent A* se esse *agent* executar uma *action* tal que o pós-estado dessa *action* seja uma *situation* que satisfaça a *proposition* desse *commitment*.

Appointment é um tipo especial de *commitment*, cuja *proposition* explicitamente se refere a um *time interval*. Compare as seguintes sentenças: “Minha namorada e eu assistiremos a uma peça teatral” e “Minha namorada e eu assistiremos a uma peça teatral neste final de semana às 19h”. A primeira sentença é um exemplo de *social commitment* e a segunda um exemplo de *appointment*. *Appointments* podem ser classificados como *Self-Appointment* ou *Social Appointment*. Um *Closed Appointment* é um *closed commitment* cujo conteúdo proposicional se refere explicitamente a um *time interval*.

3.6. Considerações Finais do Capítulo

Este capítulo apresentou uma breve introdução às ontologias destacando porque este conceito e um breve histórico de quando e onde surgiu, e como e porque este conceito vem sendo utilizado na área da Ciência da Computação e para a modelagem conceitual em especial. Um dos motivos de usar de ontologias é a possibilidade de se criar modelos formais, consistentes e não ambíguos dos conceitos, relações e restrições que existem em um domínio. A construção desses modelos se torna mais fácil quando se usa alguma metodologia. A sua construção também depende de uma linguagem de especificação que seja capaz de expressar tudo que há neste domínio.

Um modelo de referência para um domínio precisa ser construído o mais próximo possível da realidade desse domínio. Assim, recomenda-se construí-lo usando alguma ontologia fundacional, por exemplo, a UFO, já que essas são capazes de expressar melhor os conceitos de um dado domínio.

No caso específico deste trabalho, a construção de um modelo do domínio de Gerenciamento de Nível de Serviço a partir da conceituação deste domínio, como mostra o próximo capítulo, é feita usando a metodologia SABiO para a construção de ontologias com o suporte dos conceitos ontológicos descritos na UFO. O mapeamento do modelo conceitual para o modelo de implementação é feito utilizando a linguagem OWL e é apresentado no Capítulo 6. A Figura 3-10 mostra como a fundamentação teórica vista neste capítulo é usada no desenvolvimento e implementação deste trabalho.

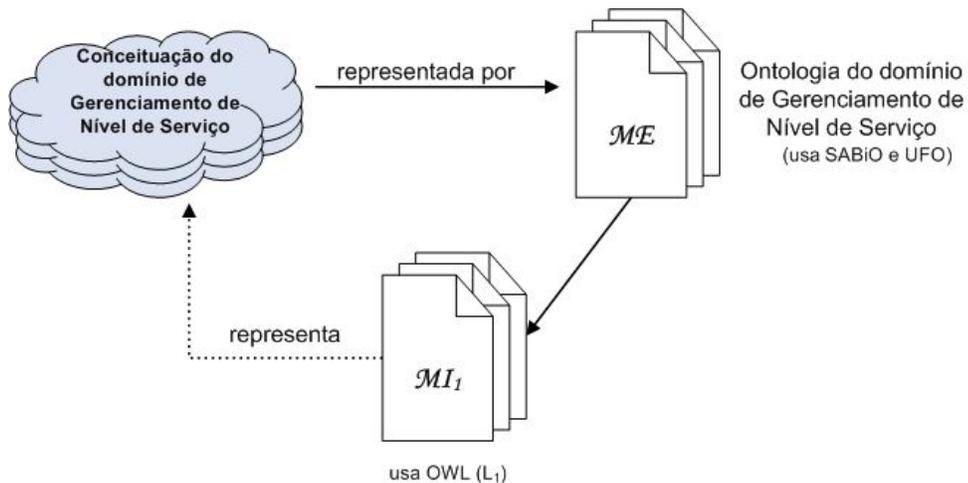


Figura 3-10 - Desenvolvimento e implementação do domínio de Gerenciamento de Nível de Serviço usando ontologias

4. Ontologia do Gerenciamento de Nível de Serviço do Padrão ITIL

4.1. Ontologia do domínio de Gerenciamento de Nível de Serviço

Este capítulo apresenta a ontologia do domínio de Gerenciamento de Nível de Serviço. Uma parte dos conceitos presentes na ontologia está relacionada ao Gerenciamento de Nível de Serviço da biblioteca ITIL. Esses conceitos explicados com detalhes na seção 2.3.1. Os demais conceitos, são conceitos definidos na UFO e estão detalhados no capítulo anterior.

4.1.1. Questões de Competência e Base da Ontologia

O desenvolvimento desta ontologia é baseada na metodologia SABiO, apresentada também no capítulo anterior deste trabalho. Portanto, antes de apresentar a ontologia de domínio desenvolvida, é necessário definir as questões de competência que essa ontologia visa responder. Essas questões são mostradas a seguir:

- Q1: O que é um serviço de TI para uma infra-estrutura?
- Q2: O que é um cliente?
- Q3: O que é um provedor de serviço de TI? Quais os tipos de provedores existentes?
- Q4: Quais são os serviços oferecidos por um provedor?
- Q5: O que é uma requisição de nível de serviço?
- Q6: Quais os acordos que podem ser firmados para a realização de um serviço? Eles são firmados entre quem?
- Q7: O que é um nível de serviço?
- Q8: Como medir os níveis de serviço e saber se um determinado serviço satisfaz os requisitos do cliente?

A Erro! Fonte de referência não encontrada. mostra um fragmento com os conceitos que formam a base da ontologia do domínio de Gerenciamento de Nível de Serviço e a origem desses conceitos a partir da UFO. É importante destacar que nessa figura, só são apresentadas as relações de subtipo (herança) entre os conceitos do domínio de Gerenciamento de Nível de Serviço. Além do mais, existem outros conceitos do domínio que não foram apresentados nessa figura. Para fim de entendimento, os conceitos em branco são os conceitos importados da UFO.

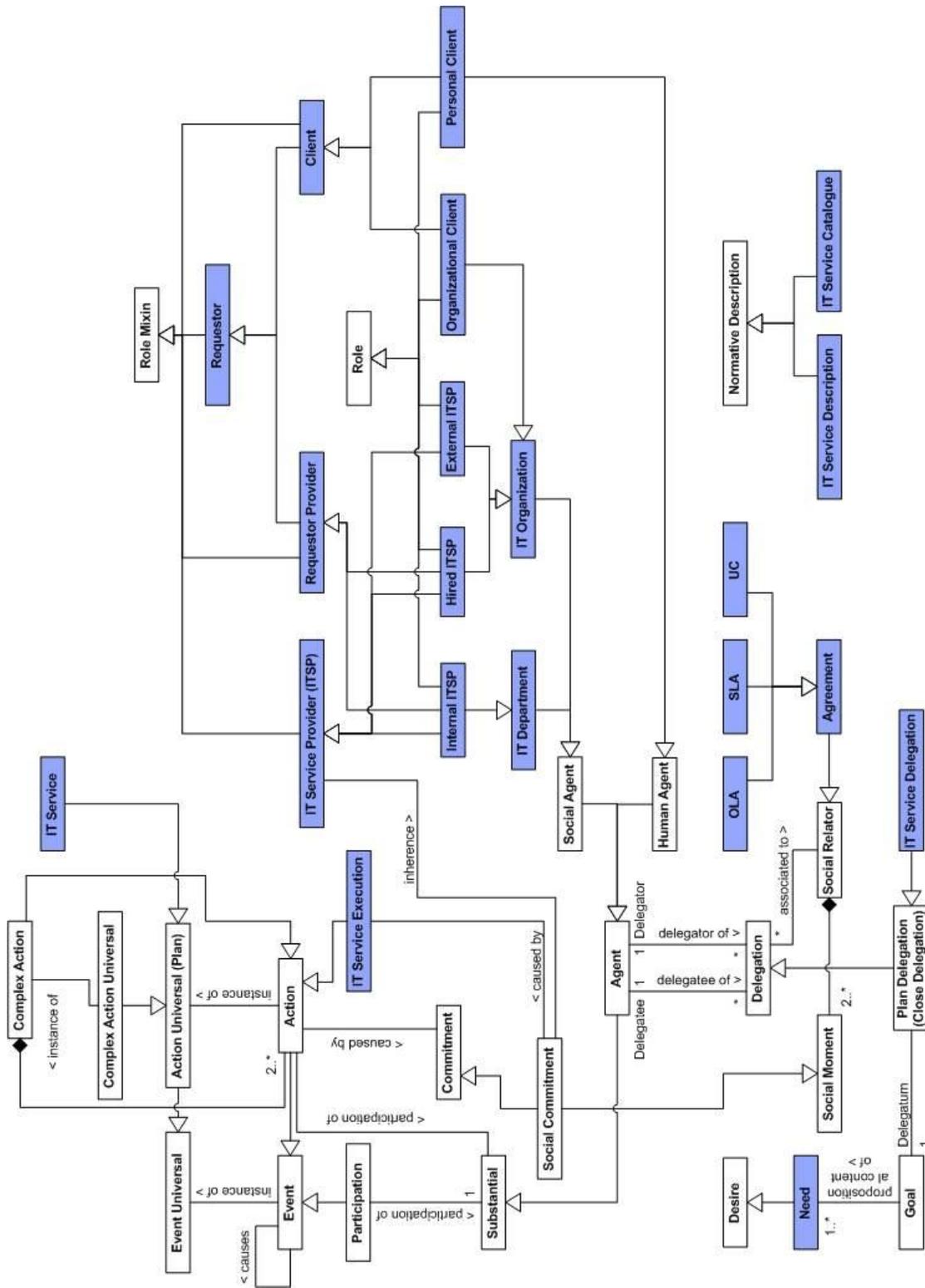


Figura 4-1 - Base da ontologia do domínio de Gerenciamento de Nível de Serviço

Essa ontologia é uma extensão e adaptação de ontologia criada por [Calvi, 2007]. Nesta ontologia um Serviço de TI (*IT Service*) é um tipo de Plano (Action Universal), ou seja, um evento intencional. Uma Execução do Serviço de TI (*IT Service Execution – ITSE*) é uma execução intencionada e ordenada de uma ou mais ações visando satisfazer o conteúdo proposicional (atingir o objetivo) de um compromisso feito com um agente (requestor). Portanto, uma ITSE é um tipo de ação que instancia um Serviço de TI. Por ser uma ação, uma ITSE pode ser atômica ou complexa e ela acaba herdando as relações de causalidade (*causes*) que existe entre eventos (*Event*). A relação *causes*, formalizada pelos axiomas (A1), (A2), (A3) e (A4), é irreflexiva (um evento x não pode causar ele mesmo), assimétrica (se um evento x causa outro evento y, então y não pode causar x) e transitiva (se um evento x causa um evento y que causa outro evento z, então x causa z). Mais detalhes sobre o que é um evento e sobre as relações de causalidade são encontradas em [Calvi, 2007].

(A1) $\forall e1, e2 ((event(e1) \wedge event(e2) \wedge causes(e1, e2)) \rightarrow \exists s (situation(s) \wedge posState(s, e1) \wedge preState(s, e2)))$

(A2) $\forall x (event(x) \rightarrow \neg causes(x, x))$

(A3) $\forall x, y (event(x) \wedge event(y) \wedge causes(x, y) \rightarrow \neg causes(y, x))$

(A4) $\forall x, y, z (event(x) \wedge event(y) \wedge event(z) \wedge causes(x, y) \wedge causes(y, z) \rightarrow causes(x, z))$

4.1.2. Ontologia de Gerenciamento de Nível de Serviço

A Figura 4-2 - Fragmento principal da ontologia de Gerenciamento de Nível de Serviço mostra o fragmento principal da ontologia de domínio de Gerenciamento de Nível de Serviço, desta vez, apresentando a relação entre os conceitos do domínio. Como apontado anteriormente, um Serviço de TI (*IT Service*) é um tipo de Plano (Action Universal) e Execução do Serviço de TI (*IT Service Execution – ITSE*) é um tipo de ação que instancia um Serviço de TI.

As execuções de serviços (ITSE) são ações realizadas por um tipo especial de agente denominado na ontologia de Provedor de Serviço de TI (*IT Service Provider – ITSP*) a pedido de um Solicitante (Requestor), e elas são causadas por compromissos que um ITSP assume com um solicitante. Tais compromissos podem ser entre um Cliente (*Client*) e um provedor de serviço contratado por este cliente (*Hired ITSP - HITSP*); pode ser entre agentes internos que fazem parte deste provedor (*Internal ITSP - IITSP*); e entre um HITSP e fornecedores de serviço externos (*External ITSP - EITSP*). Um ITSP, portanto, pode assumir o papel de um Hired ITSP, de um Internal ITSP ou de um External ITSP dependendo do compromisso assumido por este provedor. Já o solicitante pode tanto um cliente quanto um provedor de

Todo ITSP fornece pelo menos um Serviço de TI. Cada serviço fornecido por um ITSP precisa ser descrito no único Catálogo de Serviço (*IT Service Catalogue*) deste provedor, através de uma Descrição de Serviço de TI (*IT Service Description*) que é um tipo de Descrição Normativa (*Normative Description*). Portanto, um ITSP fornece um Serviço de TI se e somente se existir uma Descrição deste serviço no Catálogo de Serviço deste ITSP.

(A5) $\forall x (ITSP(x) \rightarrow \exists y (IT_Service(y) \wedge provides(x,y)))$

(A6) $\forall x,y,z (ITSP(x) \wedge IT_Service(y) \wedge provides(x,y) \wedge Service_Catalogue(z) \wedge has(x,z) \rightarrow \exists w (IT_Service_Description(w) \wedge describes(w, y) \wedge has(z,w)))$

Toda requisição (*request*) de serviço é feita por um solicitante (*Requestor*), e essa requisição é sempre motivada por uma necessidade (*Need*) do solicitante. Quando um solicitante requer um serviço ele está delegando um desejo que ele tem no momento a um ITSP e essa delegação tem o objetivo de satisfazer o conteúdo proposicional da necessidade do solicitante. O processo de requisição de um serviço é melhor detalhado mais adiante na Figura 4-3-Fragmento de requisição de serviço da ontologia de Gerenciamento de Nível de Serviço.

Da requisição de um serviço feito por um solicitante a um ITSP, surge um Acordo (*Agreement*) entre ambos que vai mediar todo o fornecimento do serviço. Neste acordo estão todas as responsabilidades pertinentes a ambas as partes. Uma das responsabilidades diz que a requisição gerada precisa ser atendida, justamente para satisfazer a objetivo do solicitante. Esse atendimento é feito por um ITSP via a realização de uma ITSE, causada pelo acordo, que busca satisfazer a requisição. Assim, quando um solicitante requisita um serviço, vai existir um acordo que vai mediar o fornecimento do serviço, e esse acordo vai causar uma ITSE. Quando o solicitante é um cliente, o acordo que vai mediar é do tipo SLA.

(A7) $\forall x,y (Requestor(x) \wedge IT_Service(y) \wedge request(x,y) \rightarrow \exists z,w (Agreement(z) \wedge mediates(z,x) \wedge mediates(z,y) \wedge ITSE(w) \wedge caused-by(w,z)))$

(A8) $\forall x,y (Client(x) \wedge IT_Service(y) \wedge request(x,y) \rightarrow \exists z,w (SLA(z) \wedge mediates(z,x) \wedge mediates(z,y) \wedge ITSE(w) \wedge caused-by(w,z)))$

A relação realiza (*carries out*) é uma relação do tipo Participação (*Participation*, segundo a UFO-C). Por ser uma ação, uma ITSE pode ser particionada em várias sub-ações existencialmente dependentes de agentes, já que elas são realizadas pelos agentes, ou seja, cada sub-ação conta com a participação intencional de um agente (ITSP), caracterizando o que se denominou na UFO-C de *Action Contribution*.

Toda execução de um serviço que o ITSP realiza para um solicitante é causado por um Acordo (*Agreement*) entre eles. Todo acordo é uma Relação Social (*Social Relator*) que existe entre um solicitante e um ITSP. Como definido na UFO, uma relação social causa uma execução de serviço pelo fato dela ser composta de Momentos Sociais (*Social Moment*), mais especificamente ser composta por um tipo de social moment conhecido como Compromisso Social (*Social Commitment*). O *Social Commitment* por sua vez, é um tipo de Compromisso (*Commitment*) e, portanto, é a causa motivadora de uma ação realizada por um agente específico (*Action Contribution*). De acordo com [Calvi, 2007], dada uma ITSE x é causada por um Compromisso Social y , então existe um Acordo z composto por y , na qual x também é causada por z .

(A9) $\forall x,y (ITSE(x) \wedge Social_Commitment(y) \wedge caused-by(x,y) \rightarrow \exists z (Agreement(z) \wedge partOf(y,z) \wedge caused-by(x,z)))$

Como dito anteriormente, toda ITSE é causada por um social relator (mais especificamente por um Acordo) entre um cliente e um ITSP. Esse social relator, e conseqüentemente o Acordo, se associa (*associated to*) à delegação de um objetivo entre agentes. Em outras palavras, um agente a possui um objetivo o que não pode ser alcançado por ele. Assim, a depende de outro agente b para atingir tal objetivo, e para satisfazê-lo delega o a b . De acordo com a UFO, a depende de b se e somente se a tem um objetivo o que não consegue atingir por conta própria, seja por incapacidade ou pelo fato de o ir de encontro com seus outros objetivos, mas que pode ser satisfeito por b . Neste caso então, a delega o objetivo o para b , e b por sua vez, se compromete a alcançar o em favor de a . Esse compromisso assumido por b é o *social commitment* que compõe a *social relator*.

Portanto, se existe um Acordo, então vai existir uma delegação de serviço. Essa delegação que existe para satisfazer do objetivo é uma Delegação Fechada (*Close Delegation*), dado que ela descreve um plano específico para alcançar o objetivo que o agente b precisa adotar a fim de atingir tal objetivo que fora delegado a ele. Na ontologia, uma Delegação de Serviço de TI (*IT Service Delegation – ITSD*) representa esse tipo de delegação, e essa delegação se compromete a alcançar um objetivo de acordo com o plano de um Serviço de TI. Segundo a modelagem feita por [Calvi, 2007], a relação *commits-to-a-plan* entre uma ITSD x e um Serviço de TI y representa esse compromisso. A relação *commits-to-an-execution* entre uma

ITSD x e uma ITSE z existe pelo fato de z ser uma instância do Serviço de TI y e pelo qual x está comprometido para atingir um objetivo delegado.

$$(A10) \quad \forall x,y,z (\text{commits-to-a-plan}(x,y) \wedge \text{commits-to-an-execution}(x,z) \rightarrow \text{ITSD}(x) \wedge \text{ITService}(y) \wedge \text{ITSE}(z) \wedge \text{instance-of}(z,y))$$

Em qualquer delegação, sempre vão existir dois agentes envolvidos, um que é o delegador (*delegator*) e o outro que é o delegado (*delegatee*). Resumindo, para todo Acordo que causa uma ITSE, existe uma ITSD, associada ao Acordo, comprometida com o serviço e com a execução do mesmo, e existem também dois Agentes que são assumem o papel de *delegatee* e o delegador deste Acordo.

$$(A11) \quad \forall x,y,z (\text{Agreement}(x) \wedge \text{IT_Service}(y) \wedge \text{ITSE}(z) \wedge \text{instance-of}(z,y) \wedge \text{caused-by}(z,x) \rightarrow \exists a,b,w (\text{ITSD}(w) \wedge \text{associated-to}(w,x) \wedge \text{commits-to-a-plan}(w,y) \wedge \text{commits-to-an-execution}(x,z) \wedge \text{agent}(a) \wedge \text{agent}(b) \wedge \text{delegator-of}(a,x) \wedge \text{delegatee-of}(b,x)))$$

Como visto no domínio de Gerenciamento de Nível de Serviço, existem três tipos de acordos, conhecidos como Acordo de Nível de Serviço (*Service Level Agreement – SLA*), Acordo de Nível Operacional (*Operational Level Agreement – OLA*) e Contrato de Apoio (*Underpinning Contract – UC*). Basicamente, o que diferencia esses três acordos são os tipos de agentes que desempenham o papel de *delegator* e de *delegatee* na ITSD associada ao Acordo. Assim, o SLA é um acordo feito entre um *Hired ITSP (delegatee)* e um *Client (delegator)*, sendo que este *Client* não pode ser uma parte do *Hired ITSP*. Então, dada uma ITSD associada a um acordo do tipo SLA e dois agentes tal que um é o delegador e o outro o delegado, o agente delegado é um *Hired ITSP* e o agente delegador é um *Client*, e este não faz parte do ITSP.

$$(A12) \quad \forall x,y,a,b (\text{ITSD}(x) \wedge \text{SLA}(y) \wedge \text{associated-to}(x,y) \wedge \text{agent}(a) \wedge \text{delegatee-of}(a,y) \wedge \text{agent}(b) \wedge \text{delegator-of}(b,y) \rightarrow \text{ITOrganization}(a) \wedge \text{Client}(b) \wedge \neg \text{parteDe}(b,a))$$

O ITSP Contratado, recebendo esta ITSD adota o objetivo do Cliente e se compromete a alcançar esse objetivo através de uma ITSE. Esse provedor pode ser composto de outros tipos de agentes provedores de serviço (como departamentos, pessoas, sistemas computacionais), denominados na ontologia de ITSP Interno (*Internal ITSP*). Já a ITSE realizada por ele pode ser uma ação complexa, de forma que ele pode dividi-la em vários sub-serviços (ITSEs) e delegá-los aos diversos *Internal ITSP* que fazem parte deste ITSP. Dessa maneira, cada um

desses agentes contribui para a ITSE como um todo. Essa delegação de partes da ITSE a agentes internos de um provedor de serviço ocorre via acordos do tipo OLA. Portanto, um OLA é um acordo entre *Internals* ITSP de um *Hired* ITSP e ele vai existir se a ITSE causada pelo SLA for uma ação complexa.

Entretanto, se o ITSP contratado, juntamente com seus agentes provedores de serviço internos, for incapaz de realizar uma ITSE por inteiro, as partes desta execução que não podem ser realizadas pelo ITSP podem ser delegadas a um outro provedor de serviço externo a este (*External ITSP*), desde que esse provedor externo não seja o cliente do serviço, isto é, não seja o delegador do SLA. Essa delegação é feita via acordos do tipo UC. Assim, um UC é um acordo entre um *External* ITSP e um *Hired* ITSP, na qual este último delega partes da ITSE ao primeiro. De forma análoga ao que acontece no OLA, cada *External* ITSP contribui para a ITSE como um todo.

Os próximos três axiomas definem a condição de existência de um OLA e de um UC, assim como a relação de contribuição que um OLA e UC fazem para um SLA. O axioma (A14) diz que dada uma ITSE causada por um SLA for uma ação complexa (*Complex Action*), então vai existir um Acordo que não é do tipo SLA e uma parte desta ITSE (sub-serviço) que é causado por esse Acordo. A partir daí, o próximo axioma diz que dadas duas ITSEs x e y distintas tais que x é parte y , x é causada por um Acordo w , que não é o SLA, e y é causada por um SLA, então esse Acordo w contribui para o SLA e vice-versa. Além do mais, se isso for verdade, o Acordo w é um OLA ou um UC e também vai existir uma ITSD associada a este Acordo w . Essa axiomatização está descrita em (A16).

$$(A13) \quad \forall x,y (ITSE(x) \wedge Complex_Action(x) \wedge SLA(y) \wedge caused-by(x,y) \rightarrow \exists z,w (ITSE(z) \wedge parteDe(z,x) \wedge Agreement(w) \wedge \neg SLA(w) \wedge caused-by(z,w)))$$

$$(A14) \quad \forall x,y,z,w (ITSE(x) \wedge ITSE(y) \wedge parteDe(x,y) \wedge SLA(z) \wedge Agreement(w) \wedge \neg SLA(w) \wedge caused-by(x,z) \wedge caused-by(y,w) \leftrightarrow contributes-to(w,z))$$

$$(A15) \quad \forall x,y,z,w (ITSE(x) \wedge ITSE(y) \wedge \neg igual(x,y) \wedge parteDe(x,y) \wedge SLA(z) \wedge Agreement(w) \wedge \neg SLA(w) \wedge caused-by(x,z) \wedge caused-by(y,w) \rightarrow \exists k ((OLA(w) \vee UC(w)) \wedge ITSD(k) \wedge associated-to(k,w)))$$

Agora que foi estabelecida a condição de existência de um OLA ou de um UC, é preciso determinar a diferença que existe entre eles, dado que ambos são Acordos associados a uma delegação de sub-serviços que são parte de uma ITSE causada por um SLA. O que determina

se um Acordo é do tipo OLA ou UC são os papéis assumidos pelos agentes delegador e delegado na delegação associada ao Acordo. Portanto, dada uma ITSD associada a um OLA e um agente delegador e outro agente delegado deste OLA, e o OLA contribui para um SLA, então ambos os agentes são Internals ITSPs (IITSP) e ambos fazem parte de um *Hired* ITSP (HITSP) que é o delegado do SLA.

$$(A16) \quad \forall x,y,z,b,c (ITSD(x) \wedge SLA(y) \wedge OLA(z) \wedge contributes\text{-}to(z,y) \wedge associated\text{-}to(x,z) \wedge delegatee\text{-}of(b,z) \wedge delegator\text{-}of(c,z) \rightarrow \exists a (HITSP(a) \wedge IITSP(b) \wedge IITSP(c) \wedge delegatee\text{-}of(a,y) \wedge parteDe(b,a) \wedge parteDe(c,a)))$$

De forma análoga, dada uma ITSD associada a um UC e um agente delegador e outro agente delegado deste UC, e o UC contribui para um SLA, então o delegador do UC é um HITSP, que por sua vez é o delegado do SLA, e delegado do UC é um *External* ITSP (EITSP) que não faz parte do HITSP e não é o delegador do SLA.

$$(A17) \quad \forall x,y,z,b (ITSD(x) \wedge SLA(y) \wedge UC(z) \wedge contributes\text{-}to(z,y) \wedge associated\text{-}to(x,z) \wedge delegatee\text{-}of(b,z) \wedge delegator\text{-}of(a,z) \rightarrow \exists a (HITSP(a) \wedge delegatee\text{-}of(a,y) \wedge EITSP(b) \wedge \neg parteDe(b,a) \wedge \neg delegator\text{-}of(b,y)))$$

4.1.3. Ontologia de Requisição de Serviço

A Figura 4-3- Fragmento de requisição de serviço da ontologia de Gerenciamento de Nível de Serviço a seguir mostra outro trecho da ontologia que diz respeito ao processo de requisição de um serviço e à questão Q5.

Como discutido anteriormente, todo o processo de fornecimento de um serviço se inicia com uma necessidade do cliente, necessidade esta que motiva o cliente a pedir (*ask for*) por um serviço de TI que possa atendê-la. O cliente então descreve suas necessidades em um documento chamado Requisitos de Nível de Serviço (*Service Level Requirement – SLR*), que é usado para pedir os serviços existentes que atendem as necessidades descritas neste documento. Dentre os vários serviços, alguns podem atender esta necessidade e outros não. Assim, dado uma necessidade e um serviço, este serviço pode ou não atender esta necessidade (A18). Um serviço de TI atende uma necessidade se o pós-estado da execução deste serviço gerar uma situação que satisfaz uma proposição que é o conteúdo proposicional da necessidade (A19). Dado um SLR que descreve uma necessidade que pode ser atingida por um serviço, esse SLR pode ser usado para pedir este serviço (A20).

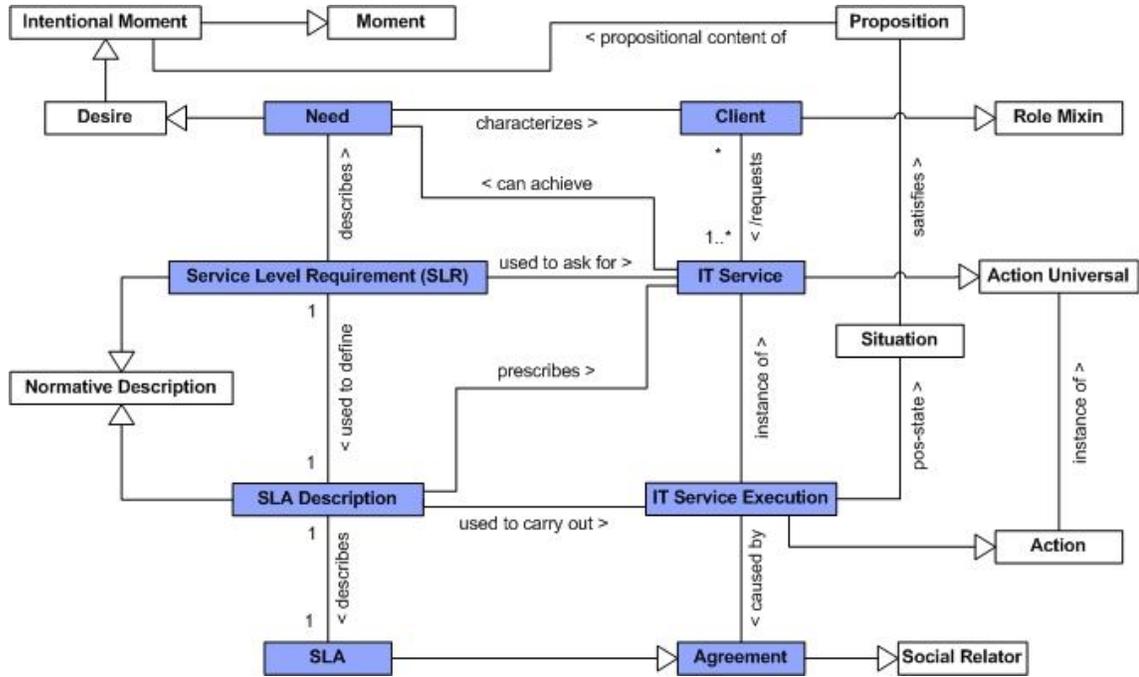


Figura 4-3- Fragmento de requisição de serviço da ontologia de Gerenciamento de Nível de Serviço

$$(A18) \quad \forall x,y (IT_Service(x) \wedge Need(y) \rightarrow can-achieve(z,y) \vee \neg can-achieve(x,y))$$

$$(A19) \quad \forall x,y (IT_Service(x) \wedge Need(y) \wedge can-achieve(x,y) \leftrightarrow \exists a,b,c (ITSE(a) \wedge instance-of(a,x) \wedge Situation(b) \wedge post-state(b,a) \wedge Proposition(c) \wedge satisfies(b,c) \wedge propositional-content-of(c,y)))$$

$$(A20) \quad \forall x,y,z (IT_Service(x) \wedge Need(y) \wedge SLR(z) \wedge describes(z,y) \wedge can-achieve(x,y) \rightarrow used-to-ask-for(z,x))$$

Essa necessidade do cliente expressa o seu desejo de encontrar um ITSP que possa realizar tal serviço. Entretanto, esse desejo não necessariamente gera uma ITSD, e conseqüentemente uma ITSE, pois pode haver uma situação na qual não exista um ITSP, ou uma situação na qual existam vários ITSPs que podem fornecer esse serviço. Fornecer um serviço é basicamente a capacidade que um ITSP tem de realizar uma execução, segundo a descrição deste serviço. Essa execução pode ser realizada pelo próprio ITSP (*can achieve*), ou ele pode delegar a execução ou parte dela a um outro agente que se compromete a realizá-la (*socially can achieve*) [Guizzardi, 2006b]. Assim, dado um ITSP que forneça um serviço pedido (*asked for*) pelo cliente e que satisfaça o desejo dele, o cliente então, pode requisitar (*/request*) este serviço ao ITSP e vice-versa (A20).

$$(A21) \quad \forall x,y,z,w (ITSP(x) \wedge IT_Service(y) \wedge provides(x,y) \wedge Client(z) \wedge ask-for(x,y) \wedge Need(w) \wedge characterizes(w,z) \wedge can-achieve(y,w) \leftrightarrow requests(z,y))$$

Quando um cliente requisita um determinado serviço, o que este cliente está fazendo na verdade é contratar (*/hires*) o ITSP que fornece este serviço (A21). Dado que este cliente contratou um ITSP, então existe um acordo do tipo SLA entre ambos que vai causar uma ITSE, como já mostrou o axioma (A8).

$$(A22) \quad \forall x,y,z (Client(x) \wedge IT_Service(y) \wedge requests(x,y) \wedge ITSP(z) \wedge provides(z,y) \rightarrow hires(x,z))$$

Esse acordo do tipo SLA, assim como o SLR, é descrito em um documento que é chamado de Descrição do SLA (*SLA Description*). Esse documento é criado a partir do SLR, ou seja, o SLR é usado para definir (*used to define*) a Descrição do SLA. Esse documento prescreve (*prescribes*) o serviço que foi solicitado, e também é usado para realizar (*used to carry out*) uma ITSE.

É importante destacar que a Descrição do SLA é um documento que só vai existir se o cliente estabelecer um acordo com um ITSP para o fornecimento de um serviço. Esse acordo é estabelecido via uma requisição, que por sua vez ocorre depois que o cliente pedir por um serviço. Se não houver a requisição, e conseqüentemente um acordo, seja porque não existe um serviço ou porque a descrição dos serviços não agradou o cliente, a Descrição do SLA não existirá. A partir do momento que o cliente requisita um serviço, o SLR usado para pedir um serviço, é usado para definir a descrição do SLA. Assim, verifica-se que uma descrição do SLA depende existencialmente e historicamente do SLR. Depende existencialmente porque para uma descrição do SLA existir, um SLR tem que existir, e depende historicamente porque uma descrição do SLA sempre é precedida por um SLR.

Para finalizar a discussão sobre este fragmento da ontologia, a descrição do SLA restringe o serviço que pode ser requisitado pelo cliente. Graças ela, o cliente não pode estabelecer o acordo de um serviço e solicitar outro. Em outras palavras, se um cliente requisita um serviço, então este serviço tem que estar prescrito na descrição do SLA que descreve o acordo firmado pelo cliente.

$$(A23) \quad \forall x,y (Client(x) \wedge IT_Service(y) \wedge request(x,y) \rightarrow \exists z,w (SLA_Description(z) \wedge SLA(w) \wedge describes(z,w) \wedge mediates(w,x)))$$

4.1.4. Ontologia de Nível de Serviço

O fragmento da ontologia exibido na **Erro! Fonte de referência não encontrada.**, trata das questões de competência Q7 e Q8, que dizem respeito aos níveis de serviço e satisfatibilidade de uma ITSE. Porém para entender a modelagem feita na **Erro! Fonte de referência não encontrada.** é necessário compreender o meta-conceito de *powertype*, que dá suporte aos conceitos modelados no fragmento da ontologia relacionado ao nível de serviço. Segundo [Guizzardi et al., 2002] um *powertype* é uma classe especial (meta-elemento) definida pelo usuário cujas instâncias são classes no modelo. O *powertype* representa um *universal* de alta ordem (*High Order Universal*) de ordem n cujas instâncias são universais de ordem n-1. A Figura 4-4 - Exemplo de powertype

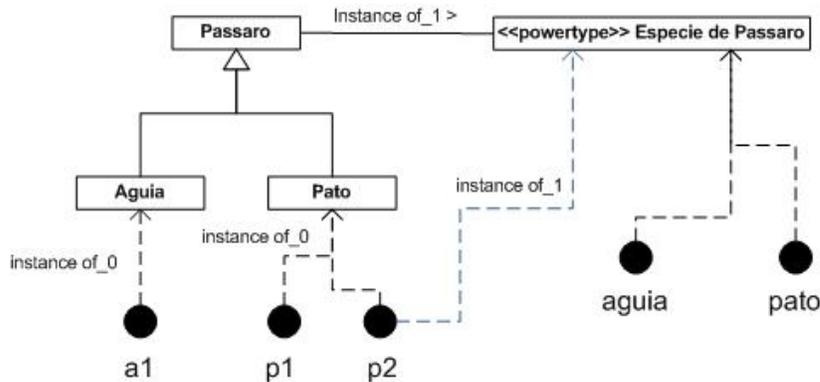


Figura 4-4 - Exemplo de powertype

Nesta figura, Águia e Pato são tipos de Pássaros, ou seja, universais de primeira ordem, cujas instâncias são águias e patos particulares (p1, p2, a1). Observando mais atentamente, percebe-se também que espécies individuais de pássaros como Águia e Pato também são instâncias do universal de segunda ordem Espécie de Pássaro. Assim, a classe Espécie de Pássaro é um powertype, já que suas instâncias são classes que representam universais de primeira ordem.

A partir daí, a Figura 4-5 - Fragmento de modelagem de powertype da ontologia mostra uma extensão da UFO usada para permitir a modelagem de *powertypes*, feita especificamente para a ontologia de Gerenciamento de Nível de Serviço. A figura mostra dois níveis de modelagem ontológica, o nível de indivíduos (*Individual*) e o de universais de primeira ordem (*First Order Universal*), aqui chamado de nível 0 (zero), e o de universais de alta ordem (*High Order Universal*), denominado de nível 1 (um). Iniciando do nível 1, tem-se que um *High Order Action Universal* é um tipo de *High Order Universal*, que por sua vez pode ser

Figura 4-5 - Fragmento de modelagem de powertype da ontologia

Universais de primeira ordem são instâncias de universais de alta ordem. A relação de instanciação que existe entre o nível 0 e o nível 1 é chamada de *instance of_1*, diferente da relação *instance of_0* que há entre indivíduos e universais de primeira ordem, ou seja, relação de instanciação entre classes da mesma ordem. Dentro da categoria de universais de primeira ordem, existem *Action Universals* e *Quality Universals*. Esses últimos são *intrinsic moment universals* que possuem uma representação em alguma dimensão da cognição humana (por exemplo, massa, comprimento estão associados à dimensão dos números positivos). Por serem *intrinsic moments*, *quality universals* são usadas para caracterizar (*characterizes*) *Action Universals*. Mais especificamente, *IT Service Level Universal*, que um tipo de *Quality Universal*, caracteriza *IT Services* que são *Action Universal*, e essa relação de caracterização é chamada de *characterizes_0*, pois relaciona elementos de primeira ordem.

Entretanto, alguns *Quality Universals* também podem caracterizar universais de alta ordem (*characterizes_1*), que é o caso do *General IT Service Level Universal*. Para exemplificar essa distinção, considere o exemplo mostrado na Figura 4-5. Idade e Peso são alguns *quality universals* que caracterizam Passaro. Da mesma forma que Expectativa de Vida e Peso Médio são *quality universals* que caracterizam Especie de Passaro. No primeiro caso existe uma relação de *characterizes_0* enquanto que no segundo caso existe uma relação de *characterizes_1*.

As *quality universals IT Service Level Universal* e *General IT Service Level Universal* são instanciadas no nível de individual como *IT Service Level* e *General IT Service Level*, respectivamente. Elas por sua vez são inerentes ao *IT Service Execution (inherence_0)* e ao *IT Service (inherence_1)*. Esta última relação *inherence_1* acontece pelo fato de um *General IT Service Level Universal* caracterizar (*characterizes_1*) um universal de alta ordem, isto é, o *powertype Service Type*.

Apresentada o fragmento da ontologia que explica sobre *powertype*, é possível agora exibir a parte da ontologia que trata sobre as questões de competência Q7 e Q8, ou seja, que trata sobre os níveis de serviço e a satisfabilidade de um serviço quanto ao nível que sua ITSE consegue alcançar. **A Erro! Fonte de referência não encontrada.** mostra essa parte da ontologia de nível de serviço.

Como apresentado na figura anterior, os *IT Services* e suas ITSEs possuem propriedades (*Quality*) inerentes a eles. As propriedades do *IT Service* são chamadas de *General IT Service Level* e as da sua ITSE são chamadas de *IT Service Level*. Uma *Quality*, sendo uma instância de uma *Quality Universal* está sempre associada a uma Estrutura de Qualidade (*Quality Structure*), que é o conjunto dos possíveis valores que uma *Quality* pode assumir. Uma *Quality* por sua vez, é um ponto em uma estrutura da qualidade, e que qualifica (*qualifies*) uma *Quality*. Os *qualities* de um *IT Service Level*, propriedade inerente a um ITSE, são o *Expected IT Service Level Value* e o *Current IT Service Level Value*. O primeiro é descrito no *SLA Description* e o segundo é produzido durante a execução de um serviço.

O conceito de Medida (*Measure*) é definido como uma *Quality Structure*, e é usado para saber se um objetivo de uma ITSE é atingido ou não. Um objetivo é medido (*measured-by*) por uma medida. Neste caso, ela é usada para medir se os *current IT service level value* pertencem (*belongs to*) ao conjunto dos *expected IT service level value*. Para isso, os valores atuais são comparados com os valores esperados, que são o(s) *quality*(s) satisfatório(s) para uma situação proposta no *goal* em questão.

A ITSE tem como meta satisfazer um *Goal*. Um *Goal* é o conteúdo proposicional de um compromisso (*Commitment*) assumido por um agente. No caso específico da execução de um serviço, esse compromisso é assumido por um ITSP. Um compromisso causa uma ação, e uma ação tem como pós-estado uma situação. Essa ação que é causada por esse compromisso é a própria ITSE do serviço requisitado e suas partes que a compõem, ou seja, os sub-serviços que fazem parte da ITSE. Esse pós-estado criado pela ITSE é uma situação gerada para satisfazer o conteúdo proposicional do compromisso que gerou a ação, ou seja, o objetivo da ação. De acordo com a UFO-C, todo objetivo, se satisfeito, é satisfeito por uma situação gerada por uma ação. Assim, um *Goal* de uma ITSE, se satisfeito, é satisfeito por uma situação gerada por uma ITSE.

Uma situação é constituída de *qualities* (relação *constituent of*). Então uma situação que satisfaz os requisitos de nível de serviço de um cliente é uma situação na qual os níveis de serviço produzidos por uma ITSE satisfazem os níveis de serviços esperados pelo cliente, ou seja, pertencem ao conjunto dos níveis de serviços aceitáveis por ele. Formalmente falando, uma ITSE satisfaz as necessidades de um cliente se os somente se valores atuais inerentes a uma ITSE e que constituem uma situação pertencerem ao conjunto dos valores esperados

descritos no SLA que é composto de um compromisso que tem como conteúdo proposicional o objetivo que é satisfeito pela situação.

$$(A24) \quad \forall x,y,z,a,b,c,d,e,f,g,h \quad (\text{Client}(x) \wedge \text{Need}(y) \wedge \text{characterizes}(y,x) \wedge \text{ITSE}(z) \wedge \\ \text{Current_IT_Service_Level_Value}(a) \wedge \text{IT_Service_Level}(b) \wedge \text{qualeOf}(a,b) \wedge \text{inheresIn}(b,z) \wedge \\ \text{Situation}(c) \wedge \text{constituentOf}(a,c) \wedge \text{postState}(c,z) \wedge \text{Expected_IT_Service_Level_Value}(d) \wedge \\ \text{belongsTo}(a,d) \wedge \text{SLA_Description}(e) \wedge \text{describes}(e,d) \wedge \text{SLA}(f) \wedge \text{describes}(e,f) \wedge \text{commitment}(g) \wedge \\ \text{isComposedOf}(f,g) \wedge \text{Goal}(h) \wedge \text{hasPropositionalContent}(g,h) \wedge \text{satisfies}(c,h) \leftrightarrow \text{satisfies}(z,y))$$

A partir de toda essa ontologia, dada uma requisição de serviço é possível saber os serviços que atendem os requisitos do cliente, o ITSP que vai fornecer este serviço, e quais são os acordos estabelecidos para o fornecimento deste serviço.

4.2. Considerações Finais do Capítulo

É defendido ao longo deste trabalho que a formalização, através de ontologias, dos conceitos presentes no domínio do processo de Gerenciamento de Nível de Serviço reduz inconsistências, ambigüidades e erros de interpretação característicos de uma descrição informal e escrita em linguagem natural. Como mostrado neste capítulo, as ontologias descritas permitem que os conceitos modelados possam ser entendidos de uma maneira mais clara e explícita, além de possibilitar a automatização do processo, capacitando máquinas a gerarem informações gerenciais a partir de dados existentes.

Cabe observar que os conceitos modelados neste capítulo abrangem somente uma parte do domínio relacionado ao Gerenciamento de Nível de Serviço. Este domínio é mais extenso do que o apresentado e apresenta outros conceitos também importantes que podem ser modelados posteriormente. Para avaliar as vantagens e benefícios de se formalizar esse domínio de Gerenciamento de Nível de Serviço do ITIL, é necessário demonstrar as capacidades desta formalização, e até mesmo motivar outros trabalhos voltados à formalização das demais disciplinas ITIL de Gerenciamento de Serviço de TI.

Nesse sentido, o Capítulo 6 apresenta um estudo de caso onde os modelos formais desenvolvidos no presente capítulo são aplicados a uma plataforma (middleware) de serviços sensíveis ao contexto (a Infracore). É mostrada a implementação desses modelos e o seu uso na geração de inferências automáticas e no compartilhamento de informações relacionadas ao

processo de gerenciamento de serviços. O Capítulo 5 apresenta uma visão geral da plataforma de serviços utilizada, denominada Infraware, e que vem sendo desenvolvida no LPRM/UFES.

5. Gerenciamento de Serviços da Plataforma

Infraware

5.1. Aplicações Sensíveis ao Contexto

Qualquer pessoa que parar um instante e observar ao seu redor é capaz de perceber informações que a ajuda a identificar a situação, ou o contexto no qual está inserida. Como exemplos de informações presentes em um contexto (informações contextuais) temos a localização, objetos e pessoas próximas, tempo e data. Estes últimos são fáceis de perceber, mas existem outros que não são tão triviais, como a atividade realizada por essas pessoas, o nível de bateria de um celular ou as condições do trânsito local.

No dicionário Houaiss a palavra contexto significa a “*inter-relação de circunstâncias que acompanham um fato ou uma situação*”. Segundo o dicionário, contexto significa tudo (objetos e acontecimentos) que está presente em um determinado espaço. Várias são as definições já criadas e cada área do conhecimento usa uma definição à sua maneira. A conceituação apresentada por [Dey, 2000] vem sendo uma das mais utilizadas na comunidade de computação ubíqua, e é a adotada neste trabalho. Dey define que “*Contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade (pessoa, lugar ou objeto) que é considerada relevante na interação entre usuário e a aplicação, incluindo o usuário e a própria aplicação*”.

As aplicações sensíveis ao contexto podem agir e tomar decisões a favor do usuário em função das informações provenientes do contexto em que ele está inserido. Uma situação que ilustra o papel de uma aplicação sensível ao contexto pode ser observada na Figura 5-1. Neste exemplo, a aplicação é executada no dispositivo móvel do usuário (João). O dispositivo e as aplicações executadas nele monitoram e recebem informações de localização do usuário, proximidades de outros usuários, informações temporais, temperatura, características de perfil e desejos do usuário, além de outras.

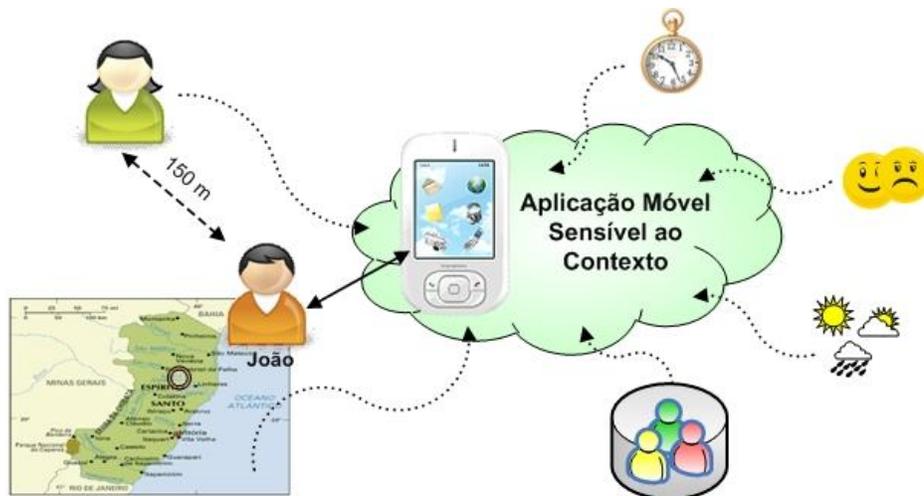


Figura 5-1 - Exemplo de aplicação sensível ao contexto [Calvi, 2007]

A classe de sistemas computacionais sensíveis ao contexto permite o desenvolvimento de aplicações mais ricas e elaboradas, que exploram a dinamicidade dos ambientes e a mobilidade do usuário. Essas aplicações tentam explorar as mudanças de contexto ocorridas dentro de um domínio dinâmico para aprimorar a interação com seus usuários.

5.2. A Plataforma de Serviços Infraware

A Infraware é um middleware de suporte à construção e execução de aplicações móveis sensíveis ao contexto, derivado da plataforma WASP [Dockhorn, 2003]. A Infraware vem sendo desenvolvida no Laboratório de Pesquisa em Redes e Multimídia (LPRM) da UFES, através dos projetos DBMWare [DBMWare, 2005], INFRAWARE [Infraware, 2005] e mais recentemente o TeleCardio [TeleCardio, 2005].

Sua arquitetura foi definida a partir do estudo de uma série de requisitos importantes para a computação sensível ao contexto [Pessoa, 2006], e ela foi projetada para superar as dificuldades encontradas nas arquiteturas distribuídas normais no que se refere à computação ubíqua e à manipulação de informações contextuais, por exemplo, a aquisição de dados contextuais heterogêneos, a definição de um modelo de contexto genérico, a descrição semântica de serviços, a interpretação semântica de contexto, o tratamento de privacidade, a coordenação de aplicações, uma linguagem de subscrição, dentre outras. Ontologias são usadas na plataforma para especificar modelos formais extensíveis que descrevem não somente o domínio das aplicações, mas também os serviços.

A Infraware é uma plataforma bastante flexível. Ela pode ser estendida com a inclusão de novos serviços e entidades, bastando estender as ontologias de domínio e de tarefa utilizadas pela plataforma. Essa flexibilidade da plataforma permite que ela seja utilizada para o desenvolvimento de várias aplicações sensíveis ao contexto, para os mais diferentes domínios.

Em sua arquitetura atual (Figura 5-2), a plataforma Infraware privilegia a execução e a entrega dos serviços, não dispondo de um mecanismo preparado para controlar eficientemente os serviços oferecidos às aplicações (não existe implementada a funcionalidade mais ampla de gerenciamento de serviços como recomendado pelo ITIL). Cada componente da Infraware possui seu próprio mecanismo de gerenciamento, otimização e monitoração dos recursos e dados contextuais. Não existe esse mesmo controle de forma integrada, ou seja, sob o prisma de toda a plataforma. Como a Infraware se propõe a fornecer serviços sensíveis ao contexto para muitas aplicações e usuários, um gerenciamento integrado de tais serviços torna-se imprescindível.

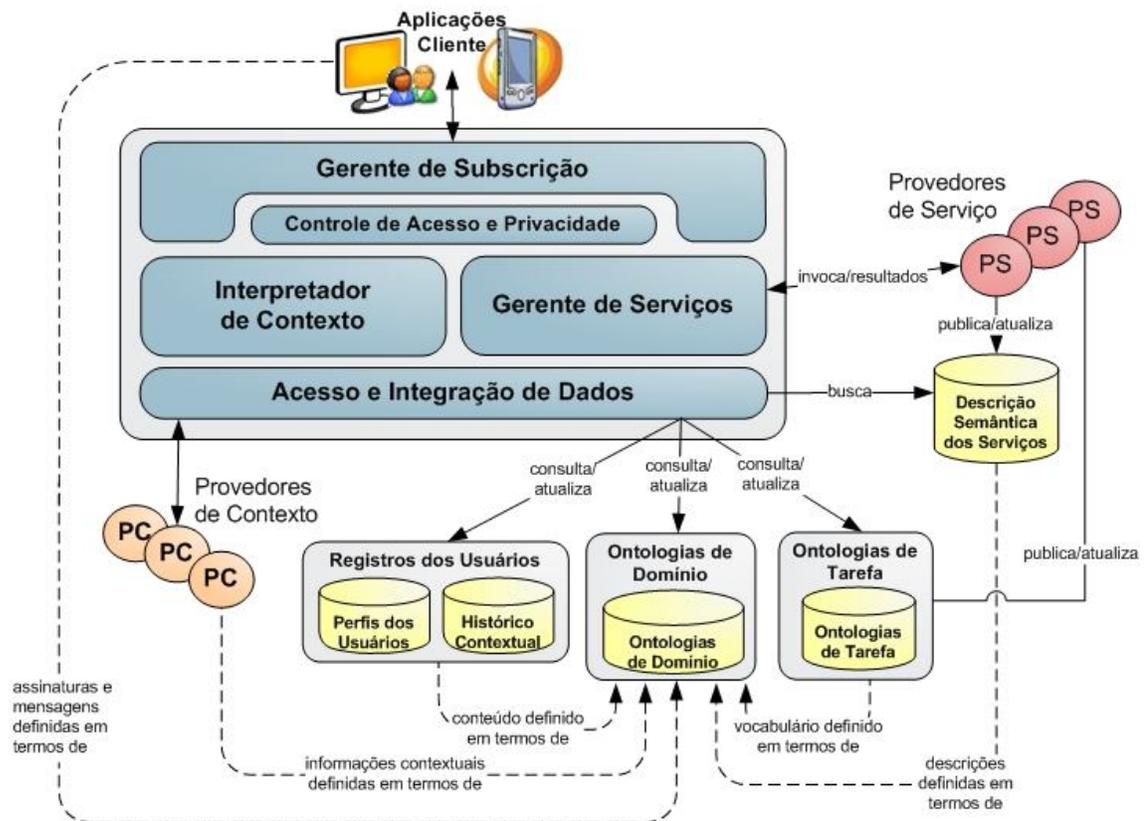


Figura 5-2 - Plataforma Infraware

Partindo desse problema, [Calvi, 2007] propõe uma nova estruturação da Infracore, dividindo-a em três grandes componentes funcionais. Essa divisão visa adequar a plataforma ao Gerenciamento de Serviços de TI segundo as visões Estratégica, Tática e Operacional. A visão Estratégica, a ser realizada pela camada Gerencial, lida com processos e atividades que permitem estabelecer um direcionamento a ser seguido pela infra-estrutura nas execuções dos serviços e no relacionamento com aplicações e usuários. Essa visão respeito à definição de quais recursos devem ser alocados para se atingir determinados objetivos. A visão Tática engloba o modo de utilização dos elementos envolvidos com a estratégia definida pela infra-estrutura, analisando dados e informações com a finalidade de se atingir a meta anteriormente especificada. A visão Tática é realizada pela Plataforma de Suporte. Já a visão Operacional, desempenhada pela camada de Sensoriamento, se preocupa com a execução propriamente dita dos serviços e dos processos acerca dos serviços, oferecendo suporte à manutenção de sua operação normal e entrega de informações a aplicações e usuários. [Calvi, 2007].

As novas camadas funcionais da Infracore, do ponto de vista arquitetural, são mostradas na Figura 5-3. Os principais componentes da plataforma Infracore, Controle de Acesso e Privacidade, Interpretador de Contexto e Gerente de Serviço, por exemplo, fazem parte da Plataforma de Suporte. A camada Gerencial criada acima da Plataforma de Suporte, contém o Gerente de Subscrição e é responsável por gerenciar as requisições enviadas pelas aplicações clientes. Já a camada de Sensoriamento foi definida para lidar com os provedores de contexto, tirando da Plataforma de Suporte a complexidade de conhecer a interface com os diversos sensores. Maiores detalhes sobre essa arquitetura podem ser encontrados em [Pereira Filho et al, 2006], [Pessoa, 2006] e [Calvi, 2007].

Com base nessa nova arquitetura dividida em camadas funcionais proposta por [Calvi, 2007], este trabalho propõe uma nova arquitetura para o Módulo de Serviço (Gerente de Serviço), componente da Plataforma de Suporte, que é melhor detalhado na Seção 5.5.

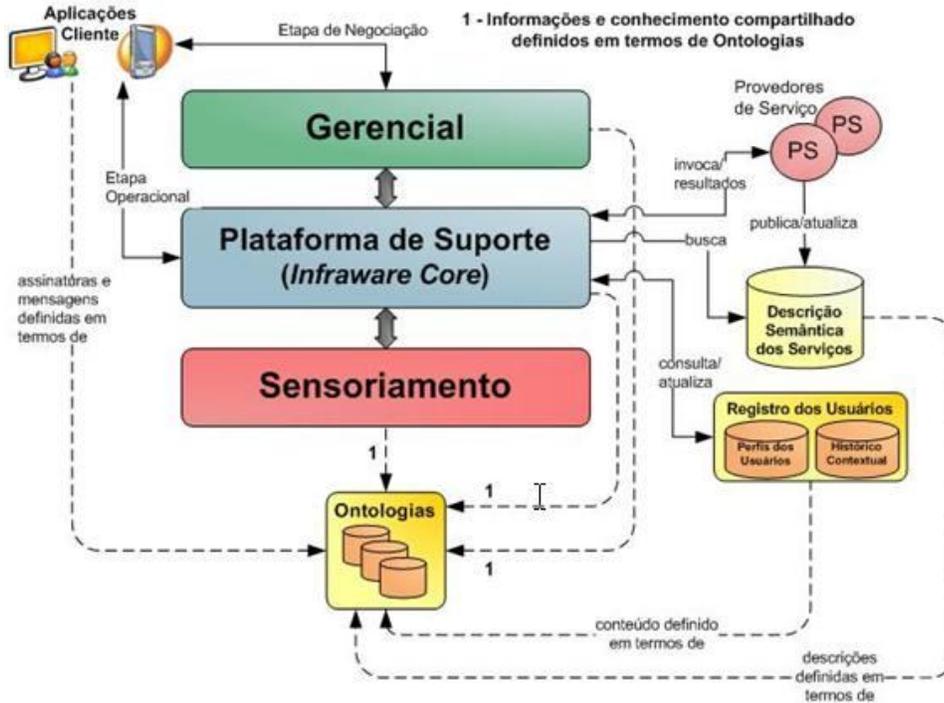


Figura 5-3 - Camadas funcionais da plataforma Infracore [Calvi, 2007]

5.3. Aplicação das Boas Práticas ITIL na Plataforma Infracore

Como defendido ao longo deste trabalho, a utilização de boas práticas de gerenciamento de serviços permite criar novas oportunidades de serviços e reduz os custos gerais de gerenciamento e de suporte através do desenvolvimento de novos modelos para manter ou melhorar a qualidade dos serviços já prestados. Para isso, é necessário desenvolver e implementar os processos de negócio e de TI apropriados, de acordo com o que as práticas ITIL defendem sobre o uso eficiente dos três “Ps”, que são as pessoas, os processos e os produtos (ferramentas e tecnologias que dão suporte ao processo).

Por sua vez, vislumbrando um cenário de uso intensivo das suas facilidades por várias aplicações e usuários simultaneamente, a plataforma Infracore necessitaria de implementar métodos capazes de lidar com o gerenciamento de suporte de serviços, como os preconizados pelo ITIL, de modo a melhorar a qualidade dos serviços prestados, além de diminuir o custo dos mesmos.

Fazendo uma analogia com os três “Ps” defendidos pelo ITIL, teríamos na plataforma Infracore: (i) as aplicações, os componentes da Infracore e os provedores de serviço e de informação contextual poderiam ser vistos como as pessoas, como se fossem agentes (em

termos na UFO) da infra-estrutura que se relacionam entre si e geram constantemente informações para o gerenciamento; (ii) os processos são as interações entre os componentes e as atividades ITIL em conjunto aplicadas na plataforma, objetivando oferecer um suporte de qualidade aos serviços; e (iii) os produtos são a tecnologia empregada na implementação dos componentes da plataforma.

A filosofia do ITIL adota uma abordagem orientada a processos que é escalável de modo a adaptar-se às grandes e às pequenas organizações de TI. Essa capacidade de adaptação permite que as práticas de gerenciamento sejam aplicáveis não somente para a plataforma Infracore como um todo, mas também para cada componente, de forma isolada (ex: Módulo de Serviço), com o intuito de melhorar a qualidade dos serviços. Para fornecer serviços de alta qualidade, alinhados ao contexto das pessoas, a Infracore deve interligar as aplicações e os serviços/informações contextuais de forma efetiva, eficiente e econômica. Isso pode ser feito, inicialmente, em duas “frentes”: (1) na plataforma Infracore como um todo; (2) e no componente Gerente de Serviço.

Sob a ótica da Infracore como um todo, a importância de se implantar as boas práticas do ITIL é motivada pela necessidade de gerenciar e sincronizar as atividades dos macro-processos e dos componentes internos da plataforma. A Infracore então, seria vista como uma “caixa-preta”, um sistema único de fornecimento de serviços aos seus usuários. Sob a ótica do Gerente de Serviço, a justificativa para implantar o ITIL neste componente pode ser explicada pelo fato do Gerente de Serviço ser o responsável pela execução e monitoração dos serviços internos e externos, além de ser o responsável por zelar pela qualidade dos serviços prestados pelos provedores externos. Como a arquitetura atual da Infracore não possui controle sobre os provedores de serviço e nem sobre como os serviços são fornecidos, ela precisa estabelecer requisitos mínimos de qualidade, com o intuito de não prejudicar a qualidade dos serviços disponibilizados aos clientes e nem a credibilidade da plataforma. Os demais componentes da Infracore não possuem contato com fornecedores externos de serviço, sendo, portanto, entidades internas que estão acobertadas pelos processos e requisitos do ITIL estabelecidos para plataforma. A implantação das boas práticas na plataforma como um todo é consequência da implantação das mesmas em cada módulo da Infracore individualmente.

A próxima seção apresenta uma nova proposta para o Módulo de Serviço da Infracore, acompanhada de uma discussão da adaptação/implantação do ITIL neste módulo. Antes,

porém, os seguintes conceitos devem ser estabelecidos, relacionando algumas definições ITIL com o contexto da Infraware:

- *Clientes* são as aplicações que se comunicam e interagem com a plataforma, são os agentes que recebem o serviço.
- *Provedor* é a unidade responsável pela prestação dos serviços de TI, ou seja, neste caso, é a visão da plataforma como um todo.
- *Fornecedor* é um terceiro responsável pelo fornecimento ou pelo suporte dos elementos que servem de base para os serviços, que no caso são os provedores de serviço, que anunciam seus serviços para a plataforma, e os provedores de informação contextual responsáveis pelo fornecimento ou suporte de dados contextuais usados para a realização dos serviços.
- *Usuários* são os agentes que utilizam o serviço diariamente, ou seja, os agentes (leia-se pessoas ou agentes humanos) que usam as aplicações ligadas à Infraware.

5.4. Requisitos do Gerente de Serviços da Infraware

Ambientes ubíquos têm como uma de suas características a agregação dinâmica de novos dispositivos, potenciais provedores de novos serviços. Esse cenário tipicamente *ad hoc* requer a existência de mecanismos que possibilitem a descoberta automática de serviços, permitindo a cooperação entre dispositivos e aplicações sem que seja necessária a intervenção direta do usuário. O processo de descoberta requer a utilização de métodos de descoberta de serviços, que, por sua vez, são comumente descritos com base em nomes, identificadores numéricos ou pares de atributo-valor. Esse tipo de descrição possui a desvantagem de ser pouco expressiva. Descrições semânticas dos serviços, baseadas no uso de ontologias, como as sugeridas nas arquiteturas SOCAM [Gu et al, 2004] e CoBrA [Chen, 2004], propiciam maior interoperabilidade entre serviços e dispositivos, além de permitirem métodos de comparação mais elaborados. As plataformas sensíveis ao contexto, portanto, necessitam de um módulo capaz de gerenciar toda a descoberta e a publicação de serviços fornecidos por ela.

Os serviços oferecidos pela Infraware podem ser classificados de duas formas: serviço interno (SI) ou serviço externo (SE). O primeiro está relacionado às descrições das capacidades internas fornecidas pela plataforma às suas aplicações clientes. Tais capacidades não estão ligadas a nenhum Web Service. A descrição do SI define as tarefas necessárias para a realização do SI, os termos utilizados e os parâmetros requisitados, tudo descrito em termos

da ontologia de domínio e de tarefa. Por outro lado, um serviço (ou serviço externo) está relacionado aos Web Services que anunciam (descrevem) suas capacidades junto à plataforma. A descrição do serviço fica armazenada num repositório de serviços e é constantemente atualizada pelo provedor. Esse serviço é selecionado (e possivelmente executado) mediante uma requisição.

Para facilitar a interação entre as aplicações e os serviços, para o fornecimento dos serviços, o Módulo de Serviço pode desempenhar dois papéis, o de Facilitador e de *Broker*. No primeiro papel, é criada uma requisição de serviço por parte do cliente, em busca de serviços disponibilizados pelos provedores de serviço e pela própria plataforma. Após selecionar os serviços disponíveis, o Módulo de Serviço retorna para ao cliente a lista de descrições de serviços compatíveis com a requisição. O usuário ou a aplicação do usuário analisam as descrições de serviços e seleciona o mais apropriado, segundo critérios como preço, QoS, tempo de resposta confiabilidade, entre outros. Depois disso, a aplicação cliente diretamente acessa o serviço interagindo com ele.

Atuando como *Broker*, após selecionar os serviços, o Módulo de Serviço invoca o serviço mais apropriado, monitora a execução do mesmo, recebe os resultados, tudo isso em favor da aplicação cliente. No final, envia a resposta ao cliente. Para que o Módulo de Serviço atue como *Broker*, a plataforma deve realizar algumas tarefas de inferência complexas, como:

- Interpretar as descrições de serviços e os requisitos do cliente;
- Encontrar o provedor de serviço ideal baseado no contexto do usuário e no seu perfil;
- Acessar o provedor selecionado;
- Interagir com o provedor de serviço sempre que necessário para atender à requisição;
- Retornar o resultado à aplicação cliente.

Algumas vantagens de se ter o Módulo de Serviço atuando como um *Broker* são:

- Tolerância a falhas: caso um serviço se torne indisponível, o módulo pode tentar encontrar outro provedor que seja compatível;
- Privacidade e segurança: o ponto central de confiança entre clientes, provedores de contexto e de serviço é a plataforma, ou seja, todos confiam nas tarefas realizadas pela plataforma. Portanto, os clientes não precisam se preocupar se estão interagindo diretamente com os provedores de serviço confiáveis e vice-versa. Além do mais, a

plataforma é capaz de esconder a identidade de ambas as partes (clientes e provedores), tornando ambas anônimas uma para a outra.

Portanto, alguns dos requisitos para o Módulo de Serviço são:

- Publicação de descrições de serviços internos e serviços externos: o Módulo de Serviço deve fornecer meios para os provedores externos publicarem as descrições dos seus serviços, além de meios de publicar as descrições dos serviços oferecidos pela própria plataforma.
- Descoberta e seleção de serviços: depois que a aplicação cliente envia uma requisição de serviço, o Módulo de Serviço deve descobrir os serviços (internos ou externos) que realizam a tarefa descrita na requisição. Após isso, o módulo deve selecionar os serviços mais apropriados, baseado nos requisitos não funcionais (QoS, custo, confiabilidade, disponibilidade, etc.) definidos na requisição e/ou nas informações contextuais do usuário, como sua localização.
- Composição de serviços: em alguns casos não é possível encontrar um único serviço que realiza todas as tarefas definidas na requisição. O Módulo de Serviço deve então, descobrir e selecionar o conjunto de serviços que juntos realizam a tarefa. É a composição deles, ou seja, é a definição da ordem de execução dos serviços (seqüencial, paralela, com ordem de precedência), que atende as necessidades do cliente. A composição do serviço é publicada como um novo serviço para evitar refazer a composição.
- Execução de serviços: o Módulo de Serviço, funcionando como um broker deve invocar os serviços envolvidos, que consiste em acessar o Web Service fornecendo os parâmetros de entrada necessários e recebendo o resultado de saída. A execução do serviço deve ser monitorada e no caso de falha, outro serviço deve ser selecionado para substituir o anterior.
- Suporte semântico: o Módulo de Serviço requer que os termos utilizados nas trocas de mensagens, descrições de serviços (internos e externos) e dados armazenados (perfil do usuário, informações contextuais) sejam semanticamente anotadas para permitir raciocínio e inferência. Essa anotação semântica é fornecida pelas ontologias.

5.5. Proposta de uma Nova Arquitetura para o Gerente de Serviços

A primeira proposta para o Módulo de Serviço foi descrita em [Pessoa, 2006] [Pereira Filho et al, 2006] e fora baseada na arquitetura apresentada por [Santos, 2004]. O Módulo de Serviço era o módulo responsável por publicar descrições de serviços, descobrir e selecionar, compor, e executar serviços, responsabilidades essas mantidas na proposta do Módulo de Serviço apresentada neste trabalho. Entretanto, com a nova divisão da Infraware em camadas funcionais, o Módulo de Serviço foi reprojetoado para se adequar à nova arquitetura, e também para permitir a implementação das funcionalidades descritas pelo ITIL.

Na nova arquitetura, o Módulo de Serviço atua como um provedor de serviço tanto para os elementos externos à Infraware quanto para os módulos internos da plataforma. Já que atua como provedor, este módulo se encarrega de gerenciar as requisições, execuções e delegações de serviços, além de estabelecer e gerenciar os acordos demais provedores de serviços (internos e/ou externos), para, assim fornecer de forma adequada os serviços necessários.

A nova arquitetura do Módulo de Serviço também é mais descentralizada. Antes, todas as ações eram comandadas pelo Coordenador, já que ele era o único que conhecia a ordem de execução das tarefas no Módulo de Serviço. Além do mais, qualquer comunicação que precisava ser feita entre os componentes do Módulo de Serviço era intermediada pelo Coordenador. Isso tudo gerava uma sobrecarga neste Coordenador.

A proposta da nova arquitetura torna as tarefas do Módulo de Serviço mais descentralizadas e torna os componentes mais independentes. Isso ocorre porque cada componente é dotado de conhecimento sobre suas tarefas e com quem ele é capaz de se comunicar. Portanto, cada componente sabe exatamente as suas funções específicas, além de executar poucas funções. Em virtude da descentralização das tarefas e de cada componente executar poucas funções, foram introduzidos novos componentes para suprir as tarefas excedentes.

Com esta proposta nova de arquitetura todos os componentes podem realizar suas tarefas (publicar, descobrir, compor, executar, etc) de forma simultânea. Essas tarefas e outras são atribuídas aos componentes: Coordenador, Descobridor, Compositor, Executor, Publicador e aos dois novos componentes, nomeados de Coletor de Informação Contextual e Entregador. A Figura 5-4 mostra a nova arquitetura de componentes do Gerente de Serviço e suas interações. Uma descrição das novas funcionalidades de cada um dos módulos é feita a seguir.

O componente *Coordenador* é responsável por gerenciar as solicitações no Módulo de Serviço. Ele é o componente que tem o controle de todas as solicitações que estão em execução e as que estão pendentes, monitorando todas elas. Sabe a quantidade de solicitações e quais são elas. Desta forma, esse componente é responsável por gerenciar os acordos com os clientes (SLAs). O Coordenador é a única interface de entrada para os outros módulos da plataforma (*Infraware core*) e uma das interfaces de comunicação com a camada Gerencial. Ele recebe as mensagens (requisições de serviço) de entrada, cria uma identificação para esta mensagem, armazena em uma fila, e a encaminha ao Descobridor.

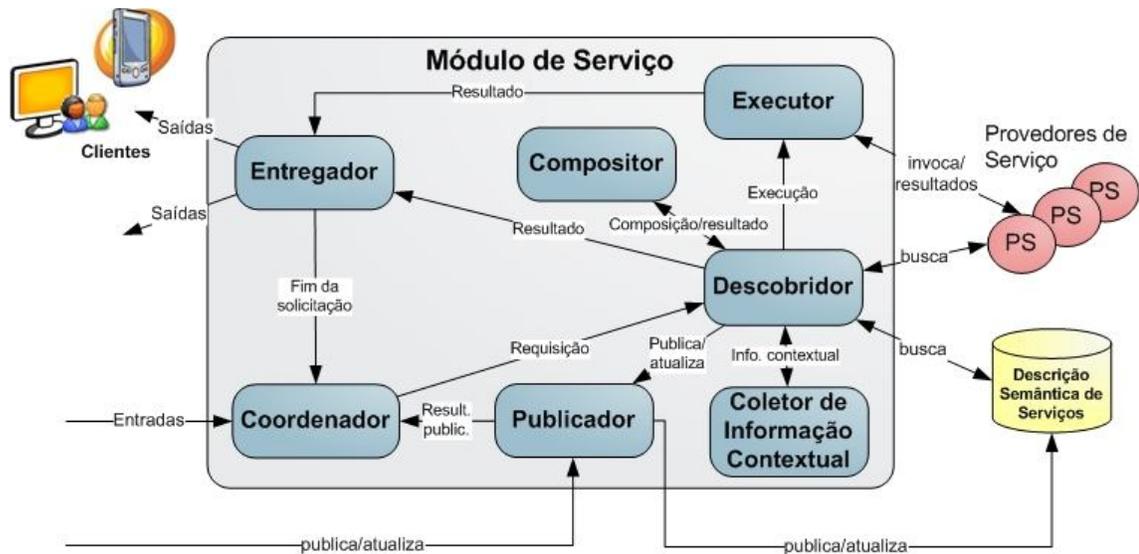


Figura 5-4 - Módulo de Serviço da Infraware

O *Descobridor* é responsável por consultar o repositório e os provedores de serviços em busca de descrições semânticas de serviços disponíveis. Esse componente recebe uma requisição do Coordenador de serviço. É ele quem sabe dizer quais são os serviços que atendem às diversas requisições. O processo de descoberta é realizado pelo protocolo de descoberta de serviço (SCaSDP) [Carmo, 2006] e o algoritmo de comparação semântica (AIComp) [Costa, 2006]. Informações contextuais para a realização da descoberta são fornecidas pelo componente Coletor de Informações Contextuais. O resultado da descoberta é enviado ao Entregador, caso o Módulo de Serviço esteja atuando como facilitador, ou é enviado ao Executor, caso a módulo esteja atuando como um broker. Se nenhum serviço for encontrado, a solicitação é enviada ao Compositor para tentar realizar a composição do serviço. Ao final da composição, caso tenha sido realizada, o Descobridor envia a nova descrição de serviço ao Publicador para ser inserida no repositório.

O *Compositor* é responsável pelo processo de composição semântica de serviços. Ele recebe do Descobridor uma solicitação para descobrir uma possível composição de serviços que satisfaça (que atende) a solicitação. Este componente decompõe a requisição em suas sub-tarefas e solicita ao Descobridor a descoberta dos serviços que realizam cada uma dessas sub-tarefas. Após receber a lista de serviços, o Compositor realiza a composição desses serviços, cria a nova descrição de serviço e entra na fase de coreografia, gerando o plano de execução dos serviços. A nova descrição e possivelmente o plano de execução são retornados ao Descobridor.

O componente *Publicador* é responsável por publicar tanto as descrições semânticas de serviços externos como as dos serviços internos. Na arquitetura antiga, a publicação dos serviços era feita diretamente entre os provedores de serviço e o Módulo de Serviço, agora é feito via camada Gerencial, ou seja, qualquer solicitação ao Módulo de Serviço passa primeiro pela camada Gerencial. Quando a solicitação é uma requisição de serviço, a camada Gerencial encaminha ao Coordenador do Módulo de Serviço e quando a solicitação é uma atualização ou inclusão de uma nova descrição de serviço, esta é passada ao Publicador. Além disso, o Publicador também recebe ordens de publicação vindas do Descobridor. Este componente faz a validação da descrição de serviço e diretamente publicação ou atualiza a descrição no repositório de Descrição Semântica de Serviços. Após publicar um novo serviço, o Publicador informa o Coordenador sobre um novo serviço disponível, este por sua vez informa as atualizações no repositório de serviços para a camada Gerencial, que providencia a atualização do Catálogo de Serviço.

O *Executor* é responsável por invocar um serviço e monitorar sua execução. Ele é também responsável por estabelecer e gerenciar acordos de execução de serviço (UCs) com os provedores de serviços externos. O Descobridor envia ao Executor o plano de execução criado pelo Compositor. O Executor aplica o plano invocando os serviços na ordem em que foram definidos e coordena a troca de mensagens entre eles. No caso de se ter apenas um serviço, não há a necessidade de um plano de execução e o Executor diretamente invoca o serviço e recebe o resultado.

O novo componente denominado *Coletor de Informações Contextuais* é responsável por gerenciar e solicitar as informações contextuais necessárias para o processo de descoberta de serviço, e por estabelecer e gerenciar os acordos (OLAs) com os módulos da camada de

suporte (*Infraware core*) da plataforma. Essas solicitações vêm do Descobridor. O Coletor se comunica diretamente ao módulo de Acesso e Integração de Dados, módulo de Interpretação de Contexto e/ou módulo de Acesso e Privacidade da plataforma, enviando a solicitação da informação contextual. Ao receber um resultado, ele o encaminha de volta ao Descobridor.

O *Entregador* é a interface de saída tanto para o Módulo de Subscrição quanto para as aplicações. É o componente responsável por entregar o resultado das solicitações enviadas ao Módulo de Serviço. Se o resultado tiver que ser entregue diretamente à aplicação, o Entregador deve conhecer a ID da aplicação, o endereço de acesso, a porta e o protocolo a ser usado. Se o resultado for entregue ao Módulo de Subscrição, basta conhecer a ID da aplicação. Quando um resultado for entregue, o Entregador comunica ao Coordenador sobre o fim de uma solicitação, caso contrário, ele informa ao Coordenador que a solicitação se tornou pendente.

O *Registro de Descrição Semântica de Serviços* é o componente responsável por armazenar as descrições de serviços publicadas pelos provedores de serviço e pela plataforma. Este componente fornece uma interface que permite a inclusão, atualização e exclusão de descrições. O Registro de Serviços interage com a plataforma via o Descobridor e Publicador do Módulo de Serviço apenas.

5.5.1. Implementação do ITIL no Módulo de Serviço

Como descrito no início desta seção, um dos motivos que levaram ao projeto da nova arquitetura do Módulo de Serviço foi a possibilidade de se implementar, na Infraware, as atividades prescritas no processo de Gerenciamento de Nível de Serviço do ITIL e, deste modo, introduzir uma melhoria significativa na gerência de serviços oferecida pela plataforma.

O fato da nova arquitetura ser mais descentralizada (ou seja, não concentrar muitas funções no componente Coordenador mas, ao contrário, distribuir funcionalidades dentre os vários componentes do módulo na camada *Infraware Core* e também entre os componentes da chamada Camada Gerencial da Infraware) auxilia a implementação, já que as atividades podem ser realizadas em paralelo, considerando requisições de serviço diferentes. Para todas as atividades deste processo existe um ou mais componentes, seja no próprio Módulo de Serviço ou na Camada Gerencial, responsáveis pela sua execução.

Nesta nova proposta do Módulo de Serviço, o Coordenador ainda é o componente que mais acumula atividades, já que ele atua como o Gerente de Nível de Serviço. Como gerente, ele é responsável por manter sempre atualizado o Catálogo de Serviço da plataforma, que fica disponível na Camada Gerencial. É o Coordenador também quem define e mantém efetivo o processo de Gerenciamento de Nível de Serviço, mesmo com todos os SLAs, OLAs e UCs existentes. Ele realiza a atividade de monitoração do módulo como um todo a fim de verificar se o gerenciamento de serviço está mesmo efetivo. Caso não esteja efetivo, o Coordenador reporta esse fato via Gerente de Configuração, além de revisar o processo a fim de melhorá-lo. Essa melhoria acontece através do relacionamento do Gerenciamento de Nível de Serviço com os demais processos de gerenciamento ITIL.

A atividade de *identificação* é implementada na Camada Gerencial da Infracore e no componente Descobridor, do Módulo de Serviço. A Camada Gerencial é a interface de recebimento dos pedidos e requisições de serviços por parte dos clientes. Estas solicitações são encaminhadas ao Módulo de Serviço, mais especificamente ao componente Descobridor, que é o responsável por identificar as necessidades contidas nessas solicitações e encontrar o(s) serviço(s) que pode(m) atendê-la.

A atividade de *definição* é realizada pela Camada Gerencial, especialmente quando o cliente requisita um serviço diretamente do Catálogo de Serviço. Quando isso ocorre, a Camada Gerencial é responsável por traduzir o serviço requisitado bem como as necessidades e restrições impostas pelo cliente para o formato que o Módulo de Serviço entende.

Toda a parte de *negociação* e *contratação* de serviços é implementada nos componentes que interagem com os clientes e os provedores de serviços internos e externos que, neste caso, são a Camada Gerencial e os componentes Descobridor e Coletor de Informação Contextual. O estabelecimento do SLA é implementado na Camada Gerencial. Ela cria o documento que descreve o SLA e encaminha ao Coordenador do Módulo de Serviço para que este último fique ciente do novo acordo estabelecido com um cliente, e para que ele possa definir o processo que realiza a execução do serviço requisitado. Fechado o SLA, inicia-se o estabelecimento das OLAs e UCs, feitos, respectivamente, pelo Coletor de Informação Contextual e Descobridor. O Descobridor define acordos com os provedores de serviços externos para o fornecimento de serviços que auxiliam na execução do serviço solicitado e o

Coletor estabelece acordos com o Módulo de Acesso e Integração de Dados e o Módulo de Interpretação de Contexto para a busca e inferência de informações contextuais.

A atividade de monitorar o cumprimento dos acordos e a execução dos serviços causados por eles é feito pelos próprios componentes que estabelece e/ou mantém esses acordos dentro do Módulo de Serviço, ou seja, pelo Coordenador, Descobridor e Coletor. Esses últimos sempre informam ao Coordenador sobre o estado de seus acordos.

Qualquer situação anormal sobre a execução de um serviço é relatado ao Coordenador, que, por sua vez, é responsável por reportar qualquer incidente ao restante da plataforma. Além de reportar incidentes, o Coordenador também informa o estados dos processos de execução de serviço e os níveis de serviços atuais produzidos pelo módulo. O processo e os níveis de serviço são constantemente revisados pelo Coordenador a fim de tentar melhorar o processo de fornecimento de serviço como um todo.

5.6. Considerações Finais

A divisão em camadas da Infracore trouxe uma reestruturação da arquitetura conceitual do Módulo de Serviço. Esta nova arquitetura melhora a arquitetura anterior, pois permite a descentralização das tarefas internas entre os componentes do módulo, redistribui algumas tarefas para a camada Gerencial e, principalmente, permite acomodar com melhor as funcionalidades descritas no ITIL.

A primeira melhoria está diretamente relacionada ao fim da sobrecarga sobre o Coordenador para a realização das tarefas do Módulo de Serviço. Com isso, há a possibilidade de vários componentes estarem executando suas tarefas simultaneamente, por exemplo, o Publicador atualiza a descrição de um serviço, enquanto que o Descobridor realiza a descoberta de um serviço já requisitado, e o Coletor de Informações Contextuais pode estar aguardando uma informação contextual que foi pedida para a Camada de Sensoriamento. A segunda melhoria está ligada ao fato de todas as solicitações vindas de elementos externos à Infracore ter que passar pela Camada Gerencial antes de chegar ao Módulo de Serviço. Com isso, a Camada Gerencial abstrai a existência desse módulo para os elementos externos. Além do mais, como todas as solicitações passam pela Camada Gerencial, ela acaba realizando uma filtragem das solicitações que são encaminhadas ao Módulo de Serviço. A terceira vantagem diz respeito à

possibilidade de se implementar com mais facilidade as funcionalidades que existem no domínio de Gerenciamento de Nível de Serviço.

É importante destacar que a nova arquitetura conceitual do Módulo de Serviço, apresentada neste estudo, é uma proposta inicial que, para assegurar a sua validade, ela deve ser implementada e avaliada através de testes e análise de desempenho, sendo essa uma proposta para trabalhos futuros.

6. Estudo de Caso: Aplicação do Processo de Gerenciamento de Nível de Serviço na Plataforma Infraware

6.1. Apresentação

Este capítulo apresenta um estudo de caso de aplicação do processo de Gerenciamento de Nível de Serviço na plataforma Infraware, utilizando as ontologias de domínio definidas pelo Capítulo 4 como base formal de inferência e compartilhamento de informação para a execução do processo de execução de serviços. Com isso, é possível produzir importantes informações a respeito da qualidade dos serviços oferecidos pela plataforma, quem fornece esses serviços, para quem é oferecido e etc.

Ao longo do capítulo, pretende-se demonstrar como os conceitos modelados pelas ontologias podem ser implementados e aplicados em um sistema computacional que provê um conjunto de serviços comuns a diversas aplicações, e que pode ser potencialmente usado em domínios e ambientes heterogêneos, com características diferentes entre si, o que exige ainda mais capacidade de gerenciamento com vistas à transparência e qualidade no fornecimento de seus serviços.

Uma importante questão discutida no capítulo refere-se à importância de se escolher uma linguagem apropriada para a implementação dos modelos. Essa, além de expressiva, deve ser computável, entendível por máquina e decidível. A partir da execução dos modelos implementados nesta linguagem, pode-se gerar uma série de importantes inferências a partir dos dados existentes, o que pode ser um grande aliado do ponto de vista gerencial.

6.2. Linguagens de Implementação do Modelo

O desenvolvimento dos modelos do domínio de Gerenciamento de Nível de Serviço (SLM), feitos no Capítulo 4 permitiu formalizar os conceitos desse domínio. Em verdade, uma das motivações que levaram a criação deste modelo usando a UFO foi justamente a possibilidade de expressar os conceitos ligados ao SLM da forma mais fiel possível. Ter a capacidade de expressar os conceitos é o primeiro passo para a validação deste modelo.

O passo seguinte é implementar esse modelo através de uma linguagem que permita essa implementação. Dessa forma, a adoção de uma linguagem adequada para representar esses conceitos, que possa ser entendível quando lida por máquinas de inferência e que possa ser computável no sentido de gerar novas informações em um tempo aceitável, se torna fundamental. Porém, para que se consiga realizar inferências e gerar novas informações, é preciso escolher uma linguagem formal que possua sintaxe e semântica não-ambígua e formulações exatas.

Existem algumas linguagens que são usadas para descrever e implementar ontologias, sendo que grande parte se tornaram padrões adotados pelo *World-Wide Web Consortium (W3C)*. Dentre as linguagens de representação destacam-se a RDF (*Resource Description Framework*) [RDF, 2004], a DAML+OIL [DAML, 2001] e a OWL (*Web Ontology Language*) [OWL, 2004]. Essas linguagens expressam os conceitos e os relacionamentos da ontologia através de triplas e elas foram projetadas para que houvesse a possibilidade de realizar processamento semântico de informação na Web. A mais utilizada entre elas é a OWL, devido a sua maior expressividade.

Para atender aos diversos propósitos e requisitos, a linguagem OWL é subdividida em três sub-linguagens, que possuem diferentes níveis de expressividade. São elas:

- **OWL Lite:** é a mais simples das sub-linguagens oferecendo suporte para classificações hierárquicas e restrições simples. Apesar de dar suporte a restrições de cardinalidade, ela aceita apenas com cardinalidade 0 ou 1. A OWL Lite é a mais fácil de se oferecer ferramentas de suporte e é uma boa alternativa para a construção de taxonomias. Ela possui menor complexidade formal que a OWL DL.
- **OWL DL:** fornece maior grau de expressividade do que a OWL Lite e, ao mesmo tempo, consegue garantir que as conclusões de seus processamentos sejam computáveis (completude) e que essas computações terminem em um tempo finito (decidíveis). Em OWL DL, todos os construtores da linguagem são permitidos, mas existem restrições com relação ao uso desses construtores. Essa sub-linguagem recebeu o nome de DL devido à correspondência com a Lógica Descritiva (*Description Logic*).
- **OWL Full:** provê a máxima expressividade e liberdade sintática que a OWL pode oferecer, permitindo até inserir novos termos ao vocabulário pré-definido, porém não

há garantia de computabilidade e decidibilidade. Assim, máquinas de inferência podem perder a capacidade de gerar conclusões ou de realizar computações em tempo finito.

Neste trabalho, a linguagem escolhida para a implementação dos modelos desenvolvidos no Capítulo 4 é a OWL DL. A adoção dessa linguagem se deve ao fato da plataforma Infraware ter sido projetada para as tecnologias da Web Semântica. Isso significa que toda implementação feita nessa linguagem pode ser estendida ou reutilizada em outras plataformas e aplicações semelhantes à Infraware. Outra justificativa para a adoção da OWL DL é que se deseja ter uma linguagem que tenha a característica de ser computável e decidível sem que se sacrifique o poder de expressividade. Afinal de contas, é preciso fornecer uma resposta rápida e satisfatória às requisições dos clientes.

Além da OWL DL, usada para a implementação do modelo, este trabalho faz uso da linguagem *Semantic Web Rule Language* (SWRL) [Horrocks, 2003]. SWRL é usada para a formalização das regras, usando lógica de primeira ordem, existentes no modelo. Em outras palavras, a linguagem SWRL é importante para a representação dos axiomas e das regras definidas nos modelos ontológicos construídos no Capítulo 4.

A implementação das ontologias em OWL DL e a criação dos axiomas em SRWL foram feitas usando a ferramenta Protégé [Protégé, 2008] que, além de ser um editor de ontologias, consegue integrar essas duas linguagens em um mesmo ambiente. Para a melhor organização, compreensão e visualização das ontologias construídas, as mesmas são divididas em quatro módulos, como mostra a Tabela 6.1 - Modelos de Implementação em OWL a seguir.

Tabela 6.1 - Modelos de Implementação em OWL

Modelo em OWL	Descrição
UFO_Ont.owl	Representa uma implementação simplificada dos conceitos da UFO-A, B e C utilizados como base para a construção dos modelos de Gerenciamento de Nível de Serviço. Tais conceitos são representados pelos retângulos brancos nas Figuras 3-7, 3-8 e 3-9.
SLM_Ont.owl	Representa a implementação dos conceitos do fragmento principal da ontologia ITIL de Gerenciamento de Nível de Serviço. Tais conceitos são representados pelos retângulos de cor mais escura na Figura 4-2.
Req_Ont.owl	Representa a implementação dos conceitos do fragmento de requisição de serviço da ontologia ITIL de Gerenciamento de

	Nível de Serviço. Tais conceitos são representados pelos retângulos de cor mais escura na Figura 4-3.
ServiceLvl_Ont.owl	Representa a implementação dos conceitos do fragmento de nível de serviço da ontologia ITIL de Gerenciamento de Nível de Serviço. Tais conceitos são representados pelos retângulos de cor mais escura na Figura 4-6.

A Figura 6-1 ilustra uma parte da representação gráfica do modelo de Gerenciamento de Serviço ITIL no *Protégé*, que é usado para gerar um dos códigos em OWL da Tabela 6.1, em especial do código do modelo *SLM_Ont.owl*. É importante destacar neste momento que, devido ao fato das linguagens OWL e SWRL possuírem capacidades de expressão menores que a lógica de primeira ordem, as ontologias de fundamentação utilizadas para a construção dos modelos de domínio do Capítulo 4, e a transformação dos modelos em modelos de implementação não é totalmente fiel. Em especial, muitos dos axiomas do Capítulo 4 que apresentam o uso de negações e o quantificador existencial não puderam ser representados da maneira como foram formalizados pela SWRL. Grande parte dos axiomas foram adaptados, durante sua implementação em SRWL para que pudessem inferir as informações necessárias. Outra observação a ser feita é a limitação do uso de restrição de cardinalidade em relações explicitamente transitivas [OWL, 2004]. Por isso, algumas escolhas foram feitas na construção dos modelos de implementação em virtude das limitações dessas linguagens e algumas das questões de competência da ontologia não foram implementadas, como as relacionadas ao nível de serviço.

Dentre as principais escolhas realizadas para a construção do modelo de implementação, encontram-se o não uso das restrições de cardinalidade nos relacionamentos de composição das Figura 4-1 - Base da ontologia do domínio de Gerenciamento de Nível de Serviço e Figura 4-2 - Fragmento principal da ontologia de Gerenciamento de Nível de Serviço. Por exemplo, as composições entre Ação (*Action*) e Ação Complexa (*Complex Action*), entre Organização de TI (*IT Organization*) e Departamento de TI (*IT Department*), e entre Provedor Contratado (*Hired ITSP*) e Provedor Interno (*Internal ITSP*) foram representadas sem a restrição de cardinalidade “2..*”. A relação de causalidade “é causado por” (*caused-by*) tiveram de ser especificadas para cada categoria de entidades envolvidas, por exemplo, a relação de causalidade entre uma ação e um compromisso foi representada pela propriedade *isCausedBy_Action_Commitment*, enquanto a relação de causalidade entre um acordo e uma ITSE foi representada pela propriedade *isCausedBy_ITSE_Agreement*. Isso foi feito também para outras relações do mesmo tipo. Além disso, algumas relações com multiplicidade maior

que um nos dois sentidos, por exemplo, a relação */requests* da Figura 4-3 foi representada por duas propriedades em OWL, uma para cada sentido, formando, no exemplo, as propriedades *requests* e *isRequestedBy*. O mesmo ocorreu para outras relações semelhantes.

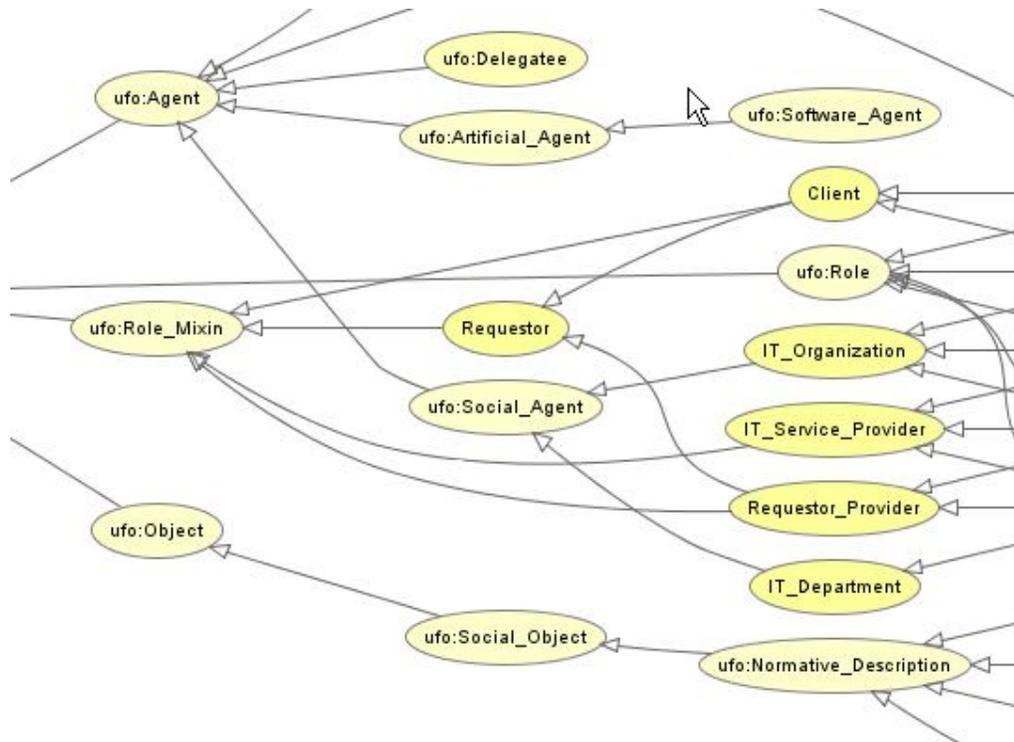


Figura 6-1 - Modelo do Gerenciamento de Serviço em OWL construído no Protégé

Como já mencionado, as expressões em SWRL dos axiomas descritos no Capítulo 4 não utilizam negações e o quantificador existencial. Por isso, nenhum axioma com essas características pôde ser representado integralmente, sendo representados sem o quantificador existencial. Isso provoca uma perda de expressividade, mas os resultados produzidos a partir disso atenderam satisfatoriamente a necessidade do cenário apresentado no estudo de caso. Além disso, os axiomas que utilizam o operador lógico “OU” foram desmembrados em duas expressões, cada uma formalizando um elemento relacionado ao operador.

6.3. Cenário de Aplicação

Esta seção descreve um cenário de aplicação, como prova de conceito da ontologia, que ilustra o uso da plataforma no fornecimento e execução de serviços a uma aplicação. Este cenário está relacionado ao projeto *TeleCardio – Telecardiologia a Serviço do Paciente em Ambientes Hospitalares e Residenciais* [Castro, 2007]. O projeto TeleCardio explora a

aplicabilidade da Informática na área da Telemedicina visando a melhoria da qualidade dos serviços voltados a pacientes cardíacos crônicos em regime domiciliar ou hospitalizados em Unidades de Saúde públicas ou privadas. Vislumbra-se com este projeto a possibilidade de desenvolver aplicações de telecardiologia, móveis e sensíveis a contexto, em diferentes cenários de funcionamento. Uma dessas aplicações desenvolvidas é um sistema de acompanhamento e monitoramento da atividade cardíaca de pacientes que possuem doenças crônicas no coração.

A Figura 6-2 mostra a arquitetura do sistema, que é composto basicamente por uma *Unidade Remota*, que pode ser o próprio domicílio do paciente ou alguma Unidade de Saúde onde ele esteja internado, e a *Central de Monitoramento*, a qual pode estar localizada no próprio hospital. A Unidade Remota e a Central de Monitoramento utilizam a Internet como meio para a troca de dados. Na Unidade Remota, são identificados os seguintes elementos:

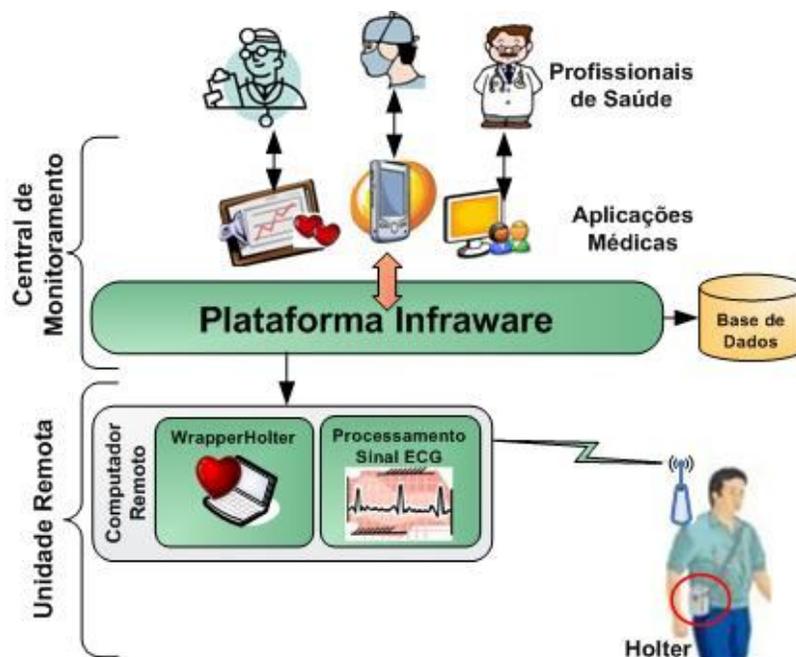


Figura 6-2 - Interação entre os atores do sistema TeleCardio [Andreão et al, 2006]

a) Dispositivo portátil: Dispositivo eletrônico de bolso que acompanha o paciente, dando-lhe mobilidade na realização das tarefas cotidianas. Ele integra um aparelho de medida de eletrocardiograma ambulatorial ou *Holter*, um botão de alarme e um dispositivo para transmissão de dados em rádio frequência para o computador remoto.

b) Computador Remoto: Responsável pela recepção, armazenamento e tratamento das informações transmitidas pelo dispositivo portátil. Através de um programa de comunicação, os dados são em seguida enviados à plataforma Infraware. O módulo de Acesso e Integração de Dados da plataforma realiza, então, a aquisição e o armazenamento dos dados vitais do paciente assistido pelo serviço.

A Central de Monitoramento, por sua vez, é composta pelos seguintes elementos:

a) Servidor de Monitoramento: Unidade funcional implementada a partir da plataforma Infraware e de uma aplicação sensível a contexto de telecardiologia [Andreão et al, 2006] para o gerenciamento das informações e acompanhamento remoto do estado clínico dos pacientes.

b) Banco de Dados Contextual: O Servidor de Monitoramento trabalha sobre uma base dados contextual contendo informações de cada paciente. Exemplos de informações incluem exames médicos, dados pessoais, histórico clínico, médico responsável e outros. Essas informações e os sinais vitais enviados pelo computador remoto, via *WrapperHolter*, compõem o Prontuário Eletrônico do Paciente (PEP). Esse prontuário fica a disposição do médico responsável para consultas e alterações.

A Figura 6-3 mostra uma parte da interface web da Aplicação TeleCardio usada para acompanhar e monitorar a atividade cardíaca dos pacientes cardíacos crônicos. Nela, é possível analisar os sinais do eletrocardiograma (ECG) de um paciente e verificar suas condições gerais de saúde do coração, inclusive com indicativos de situações mais alarmantes, como infarto ou isquemia silenciosa.

A partir do ambiente descrito, o seguinte cenário de monitoramento domiciliar para a prova de conceito é utilizado.



Figura 6-3 - Interface da Aplicação TeleCardio

6.3.1. Descrição do Cenário de Monitoramento Domiciliar

O paciente é mantido em seu domicílio, onde seu sinal ECG (eletrocardiograma) é adquirido através de um aparelho *holter* e transmitido continuamente a um computador remoto que, por sua vez, envia os dados para a Central de Monitoramento. O especialista médico de plantão em uma Unidade de Saúde ou mesmo o médico particular do paciente em sua residência ou em seu local de trabalho, pode interagir com o sistema, acompanhando o paciente a partir de um computador *desktop*, um *notebook* ou um dispositivo móvel como o PDA. O paciente, sentindo qualquer desconforto pode solicitar um serviço de atendimento médico domiciliar. Neste caso, é enviada uma ambulância à sua residência para prestar os devidos socorros. Caso o paciente sinta algo mais grave ele pode solicitar um serviço de remoção clínica. Neste caso, também é enviado uma ambulância à sua residência, que faz a remoção deste paciente até um hospital especializado.

Para dar suporte aos pacientes com problemas cardíacos, a Infracore disponibiliza para eles uma série de serviços que auxiliam os pacientes em suas necessidades, Neste cenário são ilustrados os serviços de marcação de consultas com cardiologista e de remoção clínica. Além

do mais a Infracore disponibiliza os serviços anunciados por provedores de serviço externos como o serviço de remoção do Serviço de Atendimento Móvel de Urgência (SAMU), usado neste exemplo.

Quando um paciente (cliente) tem uma necessidade, ele consulta a plataforma Infracore, através da Central de Serviços Gerencial, sobre a existência de serviços que possa satisfazer sua necessidade, seja de marcação de consultas, de atendimento domiciliar ou de remoção. Esta consulta consiste em analisar o Catálogo de Serviços existente na plataforma ou diretamente fazer um pedido à plataforma para saber se existe um determinado serviço que atenda ao paciente (cliente). Ambos podem ser feitos através de uma aplicação, que aqui é chamada de Aplicação TeleCardio. Para realizar este pedido, o cliente utiliza um documento chamado SLR para pedir o serviço. Este documento descreve todas as necessidades do cliente com relação ao serviço que ele deseja. Se não houver nenhum serviço que atenda totalmente as necessidades do paciente, a plataforma retorna alguns serviços semelhantes e que atendem parcialmente suas necessidades.

A partir daí o paciente entra em processo de negociação na qual ele analisa as descrições dos serviços retornados, ou seja, a forma como são oferecidos os serviços e escolhe aquele que mais lhe agrada. Quando o paciente escolhe um serviço, ele realiza uma requisição deste serviço à Infracore, e o resultado dessa requisição é um contrato entre a plataforma Infracore e o cliente. Este contrato é o SLA entre as partes citadas, e dado que existe este contrato, o provedor de serviço que foi contratado vai realizar uma execução do serviço solicitado para atender a requisição do cliente. Na diversos casos, a complexidade da execução do serviço (ITSE) é tanta que acaba gerando acordos entre o módulo de execução do serviço (Módulo de Serviço) e outros provedores de serviço, dado que o Módulo de Serviço por si só não consegue realizar toda a ITSE. Esses acordos podem ser internos, ou seja, entre a plataforma e os módulos da Infracore, externos, entre o Módulo de Serviço e outras entidades externas à plataforma. Os acordos internos são os OLAs e os acordos externos são os UCs.

Após a especificação do SLA e dos OLAs e UCs, o Módulo de Serviço possui todas as informações necessárias para executar o serviço, uma vez que ele conhece o cliente que requisitou o serviço, o(s) provedor(es) que vai(vão) realizá-lo, os acordos estabelecidos entre os provedores e o cliente, as delegações que são transferidas e as execuções que serão executadas por cada provedor. A partir daí a Infracore, via Módulo de Serviço, executa todas

as ações necessárias para a correta entrega de informações e resultados solicitados pelo cliente. Além disso, a plataforma se organiza para monitorar a execução desse serviço e levantar indicadores de desempenho e de qualidade das ações executadas e resultados entregues ao cliente.

O resultado final esperado por este cenário é poder identificar o cliente que requisitou o serviço, o(s) provedor(es) que vai(vão) realizá-lo, os acordos estabelecidos entre os provedores e o cliente, as delegações que são transferidas e as execuções que serão executadas por cada provedor.

6.4. Execução das Atividades de Gerenciamento de Serviço

Neste estudo de caso, são realizadas as sub-atividades de Identificação e Definição de Negociação referentes ao processo de Gerenciamento de Nível de Serviço do ITIL. Neste cenário, todas as atividades são de responsabilidade do Módulo de Serviço e da Camada Gerencial da Infracore, e os resultados inferidos nestas atividades são importantes para que seja sempre executado o serviço certo, isto é, o que atende a necessidade do cliente e que seja de qualidade. Para a realização dessas atividades, partiu-se de um conjunto básico de informações, e a partir das regras e de algumas intervenções manuais foram feitas várias inferências e gerados diversos resultados importantes para o cenário do projeto TeleCardio.

6.4.1. Sub-atividade de Identificação

Para identificar os serviços adequados às necessidades dos clientes, primeiro é preciso conhecer os serviços que cada provedor pode fornecer além de conhecer as necessidades de cada cliente, para que, aí sim, possa ser feito o casamento entre as informações. No primeiro caso, ou seja, para conhecer os serviços que cada provedor pode fornecer e realizar é preciso investigar as descrições de serviços existentes no catálogo deste provedor para saber se ele pode realizar o serviço que o cliente precisa. Para o segundo caso, basta conhecer as necessidades que caracterizam cada cliente.

Para o cenário de aplicação deste trabalho, definiu-se uma série de serviços (IT Service), ITSEs (IT Service Execution) que são instâncias desses serviços, e descrições para estes serviços (IT Service Description) existentes no catálogo (IT Service Catalogue) de cada provedor (IT Service Provider). Além disso, definiram-se alguns clientes (Clients) e

necessidades (Need) que caracterizam esses clientes. No Anexo A se encontram as instâncias criadas inicialmente para o modelo.

6.4.2. Sub-atividades de Definição e Negociação

Identificada a necessidade do cliente e o serviço que pode atendê-la, o próximo passo é traduzir essas necessidades para uma forma que seja compreensível para organização que vai fornecer o serviço, e definir inicialmente um provedor responsável que vai realizar a execução de tal serviço.

Durante a etapa de definição, é preciso verificar se a execução do serviço pode ser realizada por um único provedor ou se é necessário delegar parte da execução do serviço a outros provedores capacitados para isso. É preciso também definir quais partes são delegadas a quais provedores.

A etapa seguinte é a de negociação, que caminha mais ou menos junto com a de definição. Nesta etapa o provedor analisa as necessidades do cliente ao mesmo tempo em que o cliente analisa as condições que o provedor oferece para fornecer o serviço. Se ambas as partes aceitarem as condições propostas, estabelece-se um acordo entre eles do tipo SLA. Além do mais, já que partes de uma execução de serviço pode ser delegada a outro provedor, é de suma importância que também se estabeleça acordos entre os provedores para que haja um comprometimento de ambas as partes para a realização do serviço.

Nesta implementação algumas instâncias (informações) foram criadas inicialmente para que a partir delas sejam inferidas informações que estabeleçam os provedores responsáveis por uma execução e/ou partes desta execução, além dos acordos firmados entre eles e as delegações associadas a estes acordos.

6.5. Implementação e Inferência de Informações Gerenciais

Os modelos instanciados no Anexo A permitem a realização de inferências que são importantes para o estudo de caso em questão, pois simulam as atividades de identificação, definição e negociação do Gerenciamento de Nível de Serviço ITIL. Elas são importantes também para a prova de conceito, ou seja, a validação de parte do modelo desenvolvido no capítulo 4 (Figura 4-2 e Figura 4-3), e conseqüentemente, para atingir um dos objetivos deste trabalho.

Todas as inferências foram executadas na própria ferramenta *Protégé* com o auxílio do sistema de inferência de regras chamado *Jess* [Jess, 2008]. O objetivo desta implementação é mostrar que dada uma necessidade de um cliente e uma requisição de serviço feita por ele, quais são os serviços que podem atender as necessidades do cliente, o provedor contratado pelo cliente para realizar o serviço, a execução de serviço gerada a partir da requisição, as delegações de serviços existentes e os acordos firmados, tanto entre cliente e provedor quanto entre provedores.

Toda a implementação foi conceitualmente dividida em três partes, e essas partes não abrangem todos os modelos construídos, apenas os dois primeiros modelos, representados pela Figura 4-2 - Fragmento principal da ontologia de Gerenciamento de Nível de Serviço e Figura 4-3- Fragmento de requisição de serviço da ontologia de Gerenciamento de Nível de Serviço. Cada parte infere determinadas informações gerenciais que são utilizadas pela parte da implementação subsequente. É importante deixar claro neste momento que, para que algumas dessas informações fossem inferidas, outras tiveram que ser inseridas manualmente no *Protégé*, devido às limitações existentes na linguagem (OWL e SWRL) e no sistema de inferência. Assim, pode-se dizer que esta implementação não é totalmente perfeita, dada a falta de expressividades das linguagens OWL e SWRL, o que impossibilitou uma transcrição totalmente fiel dos modelos e axiomas para as linguagens. Além do mais, a implementação apenas simula as ações que vão ocorrer no Módulo de Serviço para que ele consiga fornecer os serviços que são requisitados pelos clientes, dado que o esse módulo não foi realmente implementado.

A primeira parte da implementação gera como informação os serviços que podem ser requisitados por um cliente de acordo com um pedido seu, através do documento de *Service Level Requirements* (SLR), motivado por uma necessidade sua. Esse resultado é mostrado na Figura 6-4 e também no Anexo A.

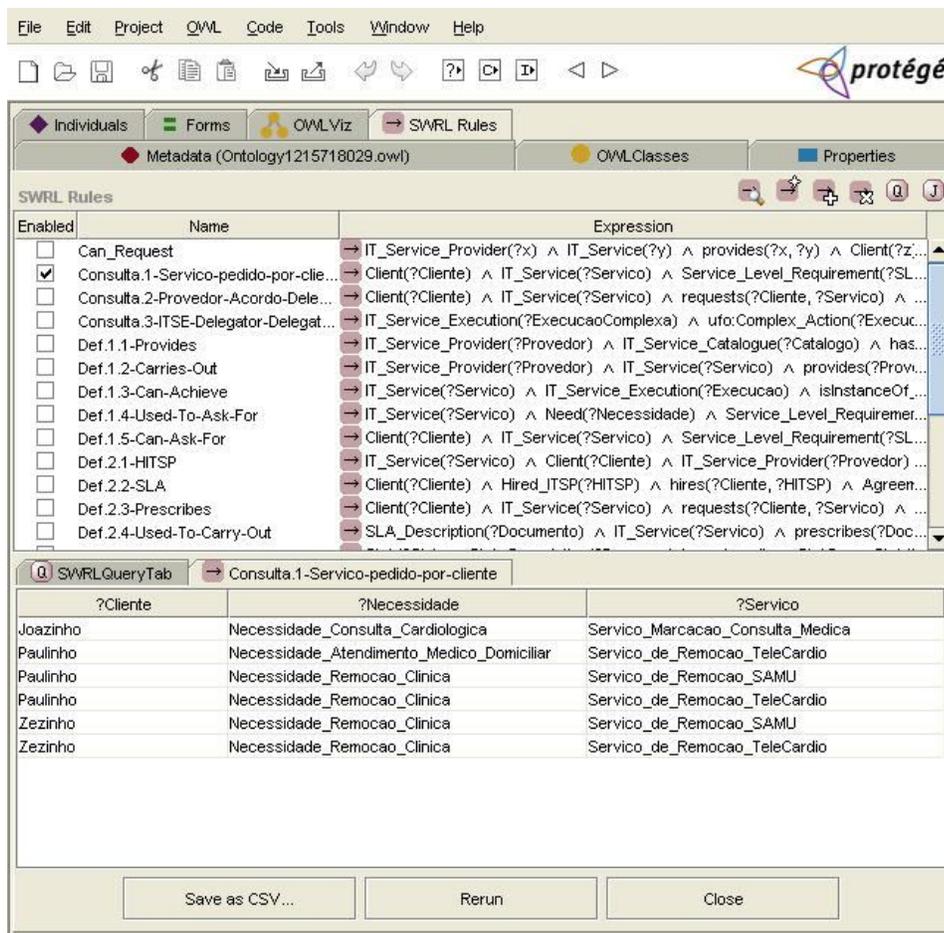


Figura 6-4 - Serviços que podem ser requisitados por um cliente

Para a produção deste resultado, o sistema teve que inferir os serviços fornecidos por cada provedor de serviço de TI (ITSP), as execuções de serviço (ITSE) realizadas por um ITSP, as necessidades que podem ser satisfeitas por cada serviço, o documento (SLR) usado para pedir um serviço, e por fim os serviços que podem ser requisitados por um cliente dado uma necessidade sua. Todas essas inferências são feitas a partir das regras *Def.1.1* à *Def.1.5*, mostradas no Anexo A.

A Figura 6-4 mostra que o cliente Joazinho necessita marcar uma consulta com um cardiologista e o serviço que pode ajudá-lo nisso é o *Servico_Marcacao_Consulta_Medica*.

Na segunda parte da implementação, dado que um cliente requisita um serviço (a partir daqueles que ele pode requisitar, inferidos na primeira parte da implementação), o sistema infere a delegação do serviço dado pelo cliente (delegador), o provedor de serviço contratado

para realizar o serviço (delegado), a execução do serviço que é realizada pelo provedor, e o SLA que é estabelecido entre o cliente e o provedor. Esse resultado é mostrado na Figura 6-5.

The screenshot shows the Protégé application interface. At the top, there's a menu bar (File, Edit, Project, OWL, Code, Tools, Window, Help) and a toolbar. Below that, a tabbed interface shows 'Metadata (Ontology1215718029.owl)', 'OWLClasses', 'Properties', 'Individuals', 'Forms', 'OWL Viz', and 'SWRL Rules'. The 'SWRL Rules' tab is active, displaying a table of rules:

Enabled	Name	Expression
<input type="checkbox"/>	Def.1.1-Provides	IT_Service_Provider(?Provedor) ^ IT_Service_Catalogue(?Catalogo) ^ has_ITSP_Catalogue(?Proved...
<input type="checkbox"/>	Def.1.2-Carries-Out	IT_Service_Provider(?Provedor) ^ IT_Service(?Servico) ^ provides(?Provedor, ?Servico) ^ IT_Ser...
<input type="checkbox"/>	Def.1.3-Can-Achieve	IT_Service(?Servico) ^ IT_Service_Execution(?Execucao) ^ isInstanceOf_ITSE_ITService(?Execuca...
<input type="checkbox"/>	Def.1.4-Used-To-Ask-For	IT_Service(?Servico) ^ Need(?Necessidade) ^ Service_Level_Requirement(?SLR) ^ describes_SL...
<input type="checkbox"/>	Def.1.5-Can-Ask-For	Client(?Cliente) ^ IT_Service(?Servico) ^ Service_Level_Requirement(?SLR) ^ Need(?Necessidad...
<input type="checkbox"/>	Def.2.1-HITSP	IT_Service(?Servico) ^ Client(?Cliente) ^ IT_Service_Provider(?Provedor) ^ provides(?Provedor, ?...
<input type="checkbox"/>	Def.2.2-SLA	Client(?Cliente) ^ Hired_ITSP(?HITSP) ^ hires(?Cliente, ?HITSP) ^ Agreement(?Acordo) ^ mediate...
<input type="checkbox"/>	Def.2.3-Prescribes	Client(?Cliente) ^ IT_Service(?Servico) ^ requests(?Cliente, ?Servico) ^ Service_Level_Requirem...
<input type="checkbox"/>	Def.2.4-Used-To-Carry-Out	SLA_Description(?Documento) ^ IT_Service(?Servico) ^ prescribes(?Documento, ?Servico) ^ IT_...
<input type="checkbox"/>	Def.2.5-Caused-By	SLA(?Sla) ^ SLA_Description(?Documento) ^ describes_SLADesc_SLA(?Documento, ?Sla) ^ IT_...
<input type="checkbox"/>	Def.2.6-ITSD	Client(?Cliente) ^ IT_Service_Provider(?Provedor) ^ hires(?Cliente, ?Provedor) ^ SLA(?Sla) ^ me...
<input type="checkbox"/>	Def.3.1-ITSP	IT_Service_Execution(?ExecucaoComplexa) ^ ufo:Complex_Action(?ExecucaoComplexa) ^ IT_Servi...
<input type="checkbox"/>	Def.3.2-ITSP-Service-Manager	IT_Department(?Departamento) ^ IT_Service_Execution(?Execucao) ^ carriesOut(?Departamento, ?I...
<input type="checkbox"/>	Def.3.3-Contributes-To	Agreement(?Acordo) ^ IT_Service_Execution(?Execucao) ^ isCausedBy_ITSE_Agreement(?Execuc...

Below the rules table, there's a 'SWRL Query Tab' showing two queries. The first query, 'Consulta.1-Servico-pedido-por-cliente', has a result table:

?Cliente	?Servico	?Delegacao	?Provedor	?Execucao	?Sla
Joazinho	Servico_Marcacao_Consulta_Medica	Delegacao_Marcacao_Consulta	Infraware	Exec_Consulta_Medica_1	Contrato_Servico_Marcacao_Consulta
Paulinho	Servico_de_Remocao_TeleCardio	Delegacao_Atendimento_Domiciliar	Infraware	Exec_Servico_Remocao_TC_1	Contrato_Servico_Atendimento_Domiciliar
Paulinho	Servico_de_Remocao_TeleCardio	Delegacao_Remocao_Clinica_2	Infraware	Exec_Servico_Remocao_TC_3	Contrato_Servico_Remocao_Clinica_2
Zezinho	Servico_de_Remocao_TeleCardio	Delegacao_Remocao_Clinica	Infraware	Exec_Servico_Remocao_TC_2	Contrato_Servico_Remocao_Clinica

At the bottom of the query results, there are buttons for 'Save as CSV...', 'Rerun', and 'Close'.

Figura 6-5 - Inferência do HITSP, do SLA existente, da ITSE e ITSD de um serviço requisitado

Para a produção deste resultado, o sistema teve que inferir o provedor de serviço que foi contratado (HITSP), o SLA e o documento que descreve esse SLA, a execução de serviço (ITSE) causada por esse SLA, e por fim a delegação (ITSD) feita pelo cliente ao provedor e quem faz o papel de delegador e delegado. Todas essas inferências são feitas a partir das regras *Def.2.1* à *Def.2.6*, mostradas no Anexo A.

A Figura 6-5 mostra, por exemplo, que Zezinho, ao requisitar o *Servico_Remocao_Telecardio* delega uma remoção clínica à Infraware, e esta por sua vez realiza a execução do serviço de remoção do Telecardio número 2, que é causada pelo *Contrato_Servico_Remocao_Clinica* estabelecido entre Zezinho e a Infraware.

A terceira e última parte da implementação infere os contratos de apoio e de nível operacional relacionados à execução de um serviço, que são estabelecidos entre os provedores de serviço. Infere também que é o delegador e o delegado de cada um desses contratos. O resultado destas inferências é exibido na Figura 6-6.

The screenshot shows the Protégé OWL editor interface. The top menu bar includes File, Edit, Project, OWL, Code, Tools, Window, and Help. The main window displays a list of SWRL rules under the 'SWRL Rules' tab. The rules are organized into columns: Enabled, Name, and Expression. Below the rules, there are two query tabs: 'Consulta 2-Providor-Acordo-Delegacao-de-um-Pedido' and 'Consulta 3-ITSE-Delegador-Delegatee-Acordo-Delegacao-de-uma-ITSE-Complexa'. The 'Consulta 3-ITSE-Delegador-Delegatee-Acordo-Delegacao-de-uma-ITSE-Complexa' tab is active, showing a table of inferred results. The table has six columns: ?ExecucaoComplexa, ?Execucao, ?Delegador, ?Delegado, ?Acordo, and ?Delegacao. The table contains several rows of data, including service identifiers, module names, and contract types.

?ExecucaoComplexa	?Execucao	?Delegador	?Delegado	?Acordo	?Delegacao
Exec_Servico_Remocao_TC_1	Exec_Servico_Localizacao	Modulo_Servico	Modulo_Acesso_Integracao_Dados	Contrato_Operacional_Servico_Localizacao	Delegacao_Exec_Localizacao
Exec_Servico_Remocao_TC_1	Exec_Servico_Remocao_SAMU	Modulo_Servico	SAMU	Contrato_Apoio_Servico_Remocao_SAMU	Delegacao_Exec_Remocao_SAMU
Exec_Servico_Remocao_TC_1	Exec_Servico_Rota	Modulo_Servico	Modulo_Interpretacao_Contexto	Contrato_Operacional_Servico_Rota	Delegacao_Exec_Rota
Exec_Servico_Remocao_TC_2	Exec_Servico_Localizacao	Modulo_Servico	Modulo_Acesso_Integracao_Dados	Contrato_Operacional_Servico_Localizacao	Delegacao_Exec_Localizacao
Exec_Servico_Remocao_TC_2	Exec_Servico_Rota	Modulo_Servico	Modulo_Interpretacao_Contexto	Contrato_Operacional_Servico_Rota	Delegacao_Exec_Rota
Exec_Servico_Remocao_TC_3	Exec_Servico_Localizacao	Modulo_Servico	Modulo_Acesso_Integracao_Dados	Contrato_Operacional_Servico_Localizacao	Delegacao_Exec_Localizacao
Exec_Servico_Remocao_TC_3	Exec_Servico_Rota	Modulo_Servico	Modulo_Interpretacao_Contexto	Contrato_Operacional_Servico_Rota	Delegacao_Exec_Rota

Figura 6-6 - Inferência dos IITSPs e EITSPs, dos OLAs e UCs e ITDSs associadas

Uma das primeiras ações para a produção deste resultado é atribuir ao Módulo de Serviço a tarefa de comandar a execução de cada serviço, já que é ele quem inicia a execução de todos os serviços. Para o cliente, existe uma única entidade que realiza a execução dos serviços que é a Infracore. Já para a plataforma, existem diversos módulos que cooperam entre si para a realização dos serviços, mas essa execução é sempre iniciada e comandada pelo Módulo de Serviço. É por isso que existe esta atribuição (Def.3.2). Para os clientes, os módulos internos da Infracore são transparentes e quando ele requisita um serviço à plataforma, na realidade ele está requisitando um serviço ao Módulo de Serviço.

Para chegar às últimas conclusões, o sistema teve que inferir os provedores de serviços de TI internos (IITSP), ou seja, os módulos da Infracore que auxiliam na execução de um serviço. Infelizmente não foi possível inferir automaticamente os provedores de serviços externos

(EITSP), pois no Jess não é possível criar axiomas que fazem uso do operador de negação. O sistema teve que inferir também os OLAs e UCs que existem entre os provedores, as delegações (ITSD) associadas a esses acordos, e por fim o delegadores e delegados das ITSDs. Todas essas inferências são feitas a partir das regras *Def.3.1* à *Def.3.7*, mostradas no Anexo A.

A Figura 6-6 mostra, por exemplo, que a *Exec_Servico_Remocao_TC_1* é composta da *Exec_Servico_Localizacao*, *Exec_Servico_Rota* e *Exec_Servico_Remocao_SAMU*. Essas sub-execuções são realizadas por IITSPs (*Modulo_Acesso_Integracao_Dados* e *Modulo_Interpretacao_Contexto*) e EITSPs (*SAMU*), e portanto, delegadas pelo Módulo de Serviço a esses provedores, via contratos operacionais e de apoio.

6.6. Avaliação

O resultado final da implementação se mostrou acima das expectativas. Pôde-se demonstrar, ainda que de forma parcial (dada a falta de expressividades das linguagens OWL e SWRL, o que impossibilitou a implementação totalmente fiel dos modelos e axiomas), que a modelagem do domínio de Gerenciamento de Nível de Serviço usando ontologias de fundamentação pode trazer grandes benefícios para os sistemas computacionais que o implementarem, no que diz respeito à inferência automática de informações de gerência.

Tomando como exemplo a Infracore, a implementação dessas ontologias e regras no Módulo de Serviço, permitirão que esse módulo descubra quais são os serviços que podem atender as necessidades que um usuário possui, e que saiba também de antemão quais são os módulos internos ou provedores de serviços externos que ele precisará contar para executar os serviços e quais partes da execução cada um realizará. A vantagem disso é que será possível fazer uma previsão de capacidade e disponibilidade para a execução de algum serviço especial.

Caso haja um incidente em algum provedor durante a execução de algum serviço, o impacto sobre o serviço como um todo pode ser minimizado uma vez que se conhece a ITSE que estava sendo realizada. Assim o módulo de serviço pode descobrir outro serviço que realiza somente aquela execução e assim finalizar a execução do serviço que foi solicitado.

Assim, constata-se que uma infra-estrutura apropriada para o gerenciamento de serviços permite que sejam fornecidos serviços cada vez mais adequados às necessidades dos clientes e com qualidade cada vez maior. No caso da Infracore, em especial do Módulo de Serviço, isso

pode ser atingido mais facilmente através do modelo de gerenciamento de nível de serviço desenvolvido neste trabalho.

Em que pese os resultados positivos conseguidos com a implementação dos modelos conceituais apresentada neste capítulo superaram as expectativas, uma vez que houve algumas dificuldades para implementar e validar, através de testes, esses modelos. As dificuldades de implementação se concentraram na falta de expressividade da linguagem de implementação adotada, com dificuldades na transcrição das ontologias para o OWL e, em especial, a transcrição dos axiomas para regras SWRL.

As limitações da linguagem SWRL em não aceitar o operador de negação, disjunções e quantificadores existenciais, fez com que alguns axiomas não fossem implementados, e os que foram perderam parte da sua expressividade e se tornaram mais genéricos, abrangendo um conjunto maior do universo. Mesmo assim, os resultados obtidos mostraram que os benefícios gerados são capazes de motivar a implementação, superando as dificuldades.

7. Conclusões

Este trabalho teve como objetivo principal desenvolver modelos conceituais que representassem o domínio do processo de Gerenciamento de Nível de Serviço estabelecido pelas boas práticas ITIL. A construção dos modelos foi feita utilizando ontologias, mais especificamente ontologias de fundamentação da UFO, que deram o suporte teórico e semântico para a formalização dos conceitos existentes no domínio. Os modelos construídos foram aplicados no Módulo de Serviço da plataforma de serviços sensíveis ao contexto Infraware. Com isso, o Módulo de Serviço teve sua arquitetura conceitual reprojeta para poder incluir em sua implementação toda a conceituação dos modelos e para proporcionar um melhor gerenciamento e fornecimento dos serviços, com base no Gerenciamento de Nível de Serviço ITIL.

Como estudo de caso e prova de conceito, os modelos foram implementados em uma ferramenta de construção de ontologias e executados usando uma máquina de inferências. Essa execução simula algumas atividades do processo de Gerenciamento de Nível de Serviço realizado pelo Módulo de Serviço da Infraware. O cenário de aplicação envolve o fornecimento de serviços médicos de qualidade a uma aplicação de telemonitoramento de pacientes cardíacos, descrita no âmbito do projeto TeleCardio.

Este trabalho advoga que a construção de uma especificação formal, usando ontologias de fundamentação, do domínio do processo de Gerenciamento de Nível de Serviço, e como o modelo conceitual desenvolvido pode tornar mais clara e explícita a compreensão dos conceitos descritos, reduzindo interpretações ambíguas e inconsistentes feitas pelos agentes envolvidos (provedores de serviço e clientes). Através de uma especificação formal como a que foi desenvolvida aqui, é possível realizar raciocínio e inferências automáticas, produzindo informações importantes para o processo de gerenciamento. A implementação e os testes realizados permitiram validar os modelos concebidos, apesar das dificuldades encontradas como a falta de expressividade das linguagens utilizadas (OWL e SWRL). É importante destacar que o uso de ontologias de fundamentação na modelagem permitiu capturar com mais precisão os conceitos do domínio de Gerenciamento de Nível de Serviço, e que essa abordagem (uso ontologias de fundamentação como base para modelar os conceitos ITIL) é praticamente um campo aberto para a pesquisa. O trabalho aqui realizado constitui um passo inicial na direção da formalização do padrão ITIL.

O trabalho também apresentou uma nova proposta de arquitetura conceitual para o Módulo de Serviço da plataforma Infracore, que agora engloba as atividades do processo de Gerenciamento de Nível de Serviço do ITIL. A nova arquitetura proposta para o Módulo de Serviço descentraliza as funções, antes concentradas no Coordenador, de forma que os componentes possam realizar agora as suas tarefas de maneira mais independentes e autônomas, melhorando assim o fornecimento de serviço aos clientes. A incorporação dos conceitos e funções ITIL no Módulo de Serviço permitiu auxiliar no cumprimento do objetivo de fornecer serviços de qualidade, uma vez que esses conceitos e funções ajudam a orquestrar o comportamento e as ações que cada componente deve tomar durante a realização do serviço. Espera-se que a implementação futura deste módulo possa comprovar os resultados a respeito do fornecimento de serviços de maior qualidade, como demonstrado na simulação apresentada no estudo de caso.

Algumas questões não foram propositalmente exploradas no escopo deste trabalho, constituindo, assim, temas de estudos futuros. Dentre algumas propostas de novos trabalhos, destacam-se:

- **A continuidade do trabalho de modelagem do domínio do processo de Gerenciamento de Nível de Serviço.** Os conceitos apresentados e modelados neste trabalho abrangem uma grande parte do domínio de Gerenciamento de Nível de Serviço ITIL. Entretanto, este domínio ainda possui conceitos importantes para o gerenciamento de serviços como as atividades de Reportagem e, principalmente, de Monitoração, que não foram explorados nesta dissertação. Isso motiva o estudo e a modelagem dessas atividades em trabalhos futuros.
- **A modelagem dos demais processos ITIL de Gerenciamento de Serviços de TI.** O estudo e a modelagem do domínio de Gerenciamento de Nível de Serviço proposta nesta tese puderam demonstrar, ainda que de forma parcial, as vantagens e benefícios da modelagem de um dos domínios de Gerenciamento de Serviço de TI da biblioteca ITIL. A modelagem do domínio de Nível de Serviço permitiu a geração de informações gerenciais que auxiliam o sistema a fornecer serviços com qualidade. A modelagem de outros domínios dos processos ITIL, principalmente os da área de entrega de serviço (como o de Capacidade e Disponibilidade) certamente dará ao sistema melhores recursos para gerenciar o fornecimento dos seus serviços.

- **A implementação do Módulo de Serviço proposto.** O estudo de caso mostrou que a modelagem conceitual é importante e auxilia no fornecimento de serviços. A implementação do Módulo de Serviço, que agora integra conceitos modelados neste trabalho, permitiria uma avaliação mais precisa do modelo conceitual e das vantagens e benefícios que ele traz ao gerenciamento de serviços.

REFERÊNCIAS

- [Andreão et al, 2006] Andreao, R.V., Pereira Filho, J.G., Calvi, C.Z.: *TeleCardio: Telecardiologia a Serviço de Pacientes Hospitalizados em Domicílio*. In.: X Congresso Brasileiro de Informática em Saúde (CBIS 2006). Florianópolis - SC, 2006.
- [Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). *The Semantic Web*. In Scientific American, 284(5):34--43.
- [Bon, 2002] van Bon, J. (2002). *IT Service Management: An Introduction*. Van Haren Publishing. ISBN 978-9080671348.
- [Bon et al, 2006] van Bon, J., Pieper, M., van Der Veen, A. (2006). *Foundations of IT Service Management, Based on ITIL*. Van Haren Publishing, 2a. edição, 4a. impressão. ISBN 9077212582.
- [Bottazzi & Ferrario, 2008] Bottazzi, E., Ferrario, E. (2008). *Towards a DOLCE Ontology of Organizations*. In Journal of Business Process Integration and Management.
- [BSI 15000, 2002] BSI (2002). *BS 15000-1: Especificação para gestão de serviços*. Relatório técnico, BSI.
- [Calvi, 2007] Calvi, C. Z. (2007). *Gerenciamento de Serviços de TI e Modelagem do Processo de Configuração ITIL em uma Plataforma de Serviços Sensíveis ao Contexto*. Tese de Mestrado, Universidade Federal do Espírito Santo (UFES), Brasil.
- [Carmo, 2006] Carmo, R. R. M. (2006). *Um Protocolo de Descoberta de Serviço para Sistemas Sensíveis ao Contexto*. Tese de Mestrado, Universidade Federal do Espírito Santo (UFES), Brasil.
- [Castro, 2007] Castro, L. R. A. (2007). *Telecardio: Telecardiologia a serviços do paciente em ambientes hospitalares e residenciais*. Projeto de Graduação, Universidade Federal do Espírito Santo (UFES).
- [Chandrasekaran et al, 1999] Chandrasekaran, B., Josephson, J. R., Benjamins, V. R. (1999). *What Are Ontologies and Why do We Need Them?* In IEEE Intelligent Systems, 14(1):20--26.
- [Chen & Kotz, 2002] Chen, G., Kotz, D. (2002). *Solar: A Pervasive-computing Infrastructure for Context-aware Mobile Applications*. Relatório Técnico TR2002-421, Dartmouth College.
- [Chen, 2004] Chen, H. (2004). *An Intelligent Broker Architecture for Pervasive Context-aware Systems*. Tese de Doutorado, University of Maryland, EUA.
- [CobiT, 2008] CobiT (2008). Control Objectives for Information and Related Technology (CobiT). <http://www.isaca.org/cobit>. Acessado em 05 de Março de 2008.
- [Costa, 2006] Costa, A. C. M. (2006). *Um Algoritmo de Comparação Semântica de Serviços, Baseado em Ontologias e Sensível ao Contexto para a Plataforma Infracore*. Projeto de Graduação, Universidade Federal do Espírito Santo (UFES), Brasil.

- [DAML+OIL, 2001] DAML+OIL (2001). *DAML+OIL Reference Description*. <http://www.w3.org/TR/daml+oil-reference>. Acessado em 09 de Fevereiro de 2008.
- [DBMWare, 2005] de Farias, C. R. G., Filho, J. G. P., Barbosa, C., Leite, M. M., Barbosa, A. C. P. (2005). *DBMWare: Infra-estrutura de Suporte ao Desenvolvimento Baseado em Modelos de Aplicações Móveis Context-aware*. Projeto CTINFO/MCT/CNPq número 50.6284/2004-2, UNISANTOS/UFES/UFJF.
- [Dey, 2000] Dey, A. K. (2000). *Providing Architectural Support for Building Context-aware Applications*. Tese de Doutorado, Georgia Institute of Technology, EUA.
- [Dockhorn, 2003] Costa, P. D. (2003). *Towards a Service Platform for Context-aware Applications*. Tese de Mestrado, University of Twente, Holanda.
- [Dockhorn et al, 2006] Dockhorn, P., Guizzardi, G., Almeida, J. P. A., Pires, L. F., van Sinderen, M. (2006). *Situations on Conceptual Modeling of Context*. In 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06).
- [Family Locator, 2007] Sprint (2007). *Family Locator*. <http://www.sprint.com/familylocator/>. Acessado em 13 de Agosto de 2007.
- [Falbo, 1998] Falbo, R. A. (1998). *Integração de Conhecimento em um Ambiente de Desenvolvimento de Software*. Tese de Doutorado, Programa de Engenharia de Sistemas e Computação da COPPE/UFRJ.
- [Falbo et al, 1998] Falbo, R. A., Menezes, C. S., Rocha, A. R. (1998). *A Systematic Approach for Building Ontologies*. In Iberoamerican Conference on Artificial Intelligence (IBERAMIA), Lisboa, Portugal.
- [Falbo et al, 2002a] Falbo, R. A., Guizzardi, G., Duarte, K. C. (2002). *An Ontological Approach to Domain Engineering*. In Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering. SEKE'2002, pages 351--358.
- [Falbo et al, 2002b] Falbo, R. A., Guizzardi, G., Duarte, K. C., Natali, A. C. C. (2002). *Developing Software for and With Reuse: An ontological Approach*. In ACIS International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications (CSITeA-02), Foz do Iguaçu, Brasil.
- [Falbo, 2004] Falbo, R. A. (2004). *Experiences in Using a Method for Building Domain Ontologies*. In Proceedings of the 16th International Conference on Software Engineering and Knowledge Engineering, SEKE'2004, pages 474--477.
- [Gärdenfors, 2000] Gärdenfors, P. (2000). *Conceptual Spaces: the Geometry of Thought*. MIT Press.
- [Grembergen, 2003] Grembergen, W. V. (2003). *Strategies for Information Technology Governance*. Idea Group Publishing. ISBN 978-1591402848.

- [Shanahan et al, 2004] Shanahan, W. G. G. P., Brown, S. W., Boyer, R. T. (2004). *Activecampus - Experiments in Community-oriented Ubiquitous Computing*. In IEEE Computer Society, 37(10):73--81.
- [Gruber, 1995] Gruber, T. R. (1995). *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. In International Journal of Human-Computer Studies, 43(5--6):907--928.
- [Grüninger & Fox, 1995] Grüninger, M., Fox, M. S. (1995). *Methodology for the Design and Evaluation of Ontologies*. In Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI'95}.
- [Gu et al, 2004] Gu, T., Pung, H. K., Zhang, D. Q. (2004). *A Middleware for Building Context-aware Mobile Services*. In Proceedings of IEEE Vehicular Technology Conference, Milão, Itália.
- [Guarino, 1998] Guarino, N. (1998). *Formal Ontology and Information Systems*. In Guarino (ed.) Formal Ontology in Information Systems. Proceedings of the International Conference on Formal Ontology and Information Systems, FOIS'98, pages 3--15, Trento, Italy.
- [Guarino & Welty, 2004] Guarino, N., Welty, C. (2004). *The Handbook on Ontologies, chapter An Overview of OntoClean*, pages 151--172. Springer-Verlag.
- [Guizzardi et al., 2002] Guizzardi, G., Herre H., Wagner, W. (2002). *Towards Ontological Foundations for UML Conceptual Models*. In Proceeding of 1st International on Ontologies, Databases and Applications of Semantics (ODBASE 2002).
- [Guizzardi, 2005] Guizzardi, G. (2005). *Ontological Foundations for Structural Conceptual Models*. PhD thesis, University of Twente. Publicado como o livro "Ontological Foundations for Structural Conceptual Models", Telematica Instituut Fundamental Research Series No. 15.
- [Guizzardi, 2006a] Guizzardi, G. (2006). *The Role of Foundational Ontology for Conceptual Modeling and Domain Ontology Representation*. In 7th International Baltic Conference on Databases and Information Systems, Vilnius, Lituânia.
- [Guizzardi, 2006b] Guizzardi, R. S. S. (2006). *Agent-Oriented Constructivist Knowledge Management*. Tese de Doutorado, University of Twente, Holanda.
- [Guizzardi, 2007] Guizzardi, G. (2007). *Frontiers in Artificial Intelligence and Applications, Databases and Information Systems IV*, chapter On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. IOS Press. ISBN 978-1-58603-640-8.
- [Guizzardi & Wagner, 2008] Guizzardi, G., Wagner, G. (2008). *Theory and Application of Ontologies, chapter Using the Unified Foundational Ontology (UFO) as a Foundation for General Conceptual Modeling Languages*. Springer-Verlag.
- [Guizzardi & Guizzardi, 2008] Guizzardi, R. S. S., Guizzardi, G. (2008). *Social Modeling for Requirements Engineering, chapter Integrating Agent-Oriented Modeling Languages Using a Foundational Ontology*. Cooperative Information Systems Series. MIT Press. Em breve.

[Guizzardi et al, 2008] Guizzardi, G., Vasconcelos, J. O., Segrini, B., Falbo, R., Guizzardi, R. S. S. (2008). *Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The Case of the ODE Software Process Ontology*. In 1st Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS'2008), Recife, Brasil.

[Gurr, 1999] Gurr, C. A. (1999). *Effective Diagrammatic Communication: Syntactic, Semantic and Pragmatic Issues*. In Journal of Visual Languages and Computing, 10(4):317--342.

[Heller & Herre, 2004] Heller, B., Herre, H. (2004). *Ontological Categories in GOL*. In Axiomathes, 14(1--3):71--90.

[Horrocks, 2003] Horrocks, I. et al.: *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, DAML. Disponível em: <http://www.daml.org/2003/11/swrl/>. Acessado em 30 de Junho de 2008.

[HP, 2000] HP (2000). *The HP IT Service Management (ITSM) Reference Model - White Paper*. Relatório Técnico, Hewlett-Packard (HP).

[HP, 2003a] HP (2003). *The HP IT Service Management (ITSM) - Transforming it Organizations into Service Providers*. Relatório Técnico, Hewlett-Packard (HP).

[HP, 2003b] HP (2003). *The HP IT Service Management (ITSM) Reference Model*. Relatório Técnico, Hewlett-Packard (HP).

[iHospital, 2007] iHospital (2007). *Det Interaktive Hospital*. <http://www.ihospital.dk>. Acessado em 13 de Agosto de 2007.

[Infraware, 2005] Pereira Filho, J.G., et al: *Projeto INFRAWARE - Uma Plataforma para Desenvolvimento de Aplicações Móveis e Context-Aware*. Departamento de Informática/UFES, Financiamento: FAPES – Fundação de Apoio à Ciência e Tecnologia do Espírito Santo, processo nº 30897041/2005.

[ISO 20000-1, 2005] ISO (2005). *ISO/IEC 20000-1:2005. Information Technology -- Service Management -- Part 1: Specification*. Relatório Técnico, International Organization for Standardization (ISO).

[ISO 20000, 2008] ISO/IEC-20000 (2008). *ISO/IEC 20000 Certification Web Site*. <http://www.isoiec20000certification.com/>. Acessado em 06 de Março de 2008.

[ITIL, 2008] ITIL (2008). *IT Service Management*. Oficial site. <http://www.itil.org/en/itilv3-servicelifecycle/index.php>. Acessado em 09 de Março de 2008.

[JESS, 2008] Jess (2008). *Jess: the Rule Engine for the Java™ Platform*. Disponível em: <http://herzberg.ca.sandia.gov/>. Acessado em 04 de Julho de 2008.

[Kordel, 2004] Kordel, L. (2004). *IT Governance Hands-on: Using CobiT to Implement IT Governance*. In Information Systems Control Journal , 2.

[Krogstie, 2000] Krogstie, J. (2000). *Evaluating UML: A Practical Application of a Framework for the Understanding of Quality in Requirements Specifications and Conceptual Modeling*. In Norwegian Informatics Conference (NIK).

[Kumar et al, 2007] Kumar, B. P., Selvam, J., Meenakshi, V. S., Kanthi, K., Suseela, A. L., Kumar, V. L. (2007). *Business Decision Making, Management and Information Technology*. In ACM Ubiquity}, 8(8).

[Macfarlane & Rudd, 2005] Macfarlane, I., Rudd, C. (2005). *Gerenciamento de Serviços de TI*. New Millenium Editora e Serviços Gráficos Ltda.

[Mealy, 1967] Mealy, H. G. (1967). *Another Look at Data*. In Proceedings of the Fall Joint Computer Conference, volume~31, pages 525--534, Anaheim, Califórnia. Washington, DC: Thompson Books, Londres: Academic Press.

[Merriam-Webster, 2008] Merriam-Webster (2008). *Dicionário merriam-webster online*. <http://www.merriam-webster.com/dictionary/ontology>. Acessado em 08 de Fevereiro de 2008.

[MOF, 2001] MOF (2001). *Microsoft Operations Framework - White Paper*. Relatório Técnico, Microsoft.

[OMG, 2007] UML, O. (2007). *OMG Unified Modeling Language, Superstructure, v2.1.2*. <http://www.omg.org/docs/formal/07-11-02.pdf>. Acessado em 09 de Fevereiro de 2008.

[OWL, 2004] OWL (2004). *OWL Web Ontology Language Reference*. <http://www.w3.org/TR/2004/REC-owl-ref-20040210>. Acessado em 09 de Fevereiro de 2008.

[Pereira Filho et al, 2006] Filho, J. G. P., Pessoa, R. M., Calvi, C. Z., de Oliveira, N. Q., do Carmo, R. R. M., Barbosa, A. C. P., de Farias, C. R. G., Leite, M. M. (2006). *Infraware: Um middleware de Suporte a Aplicações Móveis Sensíveis ao Contexto*. In 24º Simpósio Brasileiro de Redes de Computadores (SBRC 2006)}, Curitiba, Paraná, Brasil.

[Pessoa, 2006] Pessoa, R. M. (2006). *Infraware: Um Middleware de Suporte à Aplicações Sensíveis ao Contexto*. Tese de Mestrado, Universidade Federal do Espírito Santo (UFES), Brasil.

[Protégé, 2008] Protégé (2008). *The Protégé Ontology Editor and Knowledge Acquisition System*. Disponível em <http://protege.stanford.edu/>. Acessado em 30 de Junho de 2008.

[RDF, 2004] RDF (2004). *RDF primer*. <http://www.w3.org/TR/rdf-primer/>. Acessado em 09 de Fevereiro de 2008.

[RDFS, 2004] RDFS (2004). *RDF Vocabulary Description Language 1.0: RDF Schema*. <http://www.w3.org/TR/rdf-schema/>. Acessado em 09 de Fevereiro de 2008.

[Rios, 2003] Rios, D. (2003). *Using Ontologies in Context-aware Services Platforms*. Tese de Mestrado, University of Twente, Holanda.

- [Rocha & Endler, 2006] de Rocha, R. C. A., Endler, M. (2006). *Supporting Context Management for Heterogeneous and Evolving Ubiquitous Environments*. In IEEE Distributed Systems Online}, 7(4).
- [Rudd & Hodgkiss, 2004] Rudd, C., Hodgkiss, G. (2004). *An Introductory Overview of ITIL*. itSMF Ltd, 1a. Edição.
- [Sallé, 2004] Sallé, M. (2004). *IT Service Management and IT governance: Review, Comparative Analysis and Their Impact on Utility Computing*. Relatório Técnico, Hewlett-Packard Company.
- [Santos, 2004] Santos, L. O. B. (2004). *Semantic Services Support for Context-aware Platforms*. Tese de Mestrado, Universidade Federal do Espírito Santo (UFES), Brasil.
- [Schneider, 2002] Schneider, L. (2002). *Formalised Elementary Formal Ontology*. Relatório Técnico, ISIB-CNR.
- [Smith & Welty, 2001] Smith, B., Welty, C. (2001). *Ontology: Towards a new synthesis*. In Chris Welty and Barry Smith, eds., *Formal Ontology in Information Systems*. Pp. I ii-x. Ogunquit, Maine: ACM Press, 2001.
- [TeleCardio, 2005] Pereira Filho, J.G., et al: *Projeto Telecardio - Telecardiologia a Serviço do Paciente em Ambientes Hospitalares e Residenciais*. DI/DEE/UFES, Financiamento: FAPES - Fundação de Apoio à Ciência e Tecnologia do Espírito Santo, processo nº 31024866/2005.
- [Ullmann, 1972] Ullmann, S. (1972). *Semantics: An Introduction to the Science of Meaning*. Basil Blackwell, Oxford.
- [Uschold & King, 1995] Uschold, M., King, M. (1995). *Towards a Methodology for Building Ontologies*. In Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI'95.
- [Weiser, 1991] Weiser, M. (1991). *The Computer of the Twenty-first Century*. In Scientific American, 265(3):94--104.
- [Welty & Guarino, 2001] Welty, C., Guarino, N. (2001). *Supporting Ontological Analysis of Taxonomic Relationships*. In Data and Knowledge Engineering}, 39(1):51--74.

Anexo A: Regras implementadas em SWRL e executadas pelo Jess

Def.1.1 – Provides

IT_Service_Provider(?Provedor) ^ IT_Service_Catalogue(?Catalogo) ^
has_ITSP_Catalogue(?Provedor,?Catalogo) ^ IT_Service_Description(?Descricao_Servico) ^
has_Catalogue_ServiceDescription(?Catalogo,?Descricao_Servico) ^ IT_Service(?Servico) ^
describes_ServiceDescription_Service(?Descricao_Servico,?Servico) →
provides(?Provedor,?Servico)

Def.1.2 – Carries-Out

IT_Service_Provider(?Provedor) ^ IT_Service(?Servico) ^ provides(?Provedor,?Servico) ^
IT_Service_Execution(?Execucao) ^ isInstanceOf_ITSE_ITService(?Execucao,?Servico) →
carriesOut(?Provedor,?Execucao)

Def.1.3 – Can-Achieve

IT_Service(?Servico) ^ IT_Service_Execution(?Execucao) ^
isInstanceOf_ITSE_ITService(?Execucao,?Servico) ^ ufo:Situation(?Situacao) ^
isPostState_Situation_ITSE(?Situacao,?Execucao) ^ ufo:Proposition(?Proposicao) ^
ufo:satisfies(?Situacao,?Proposicao) ^ Need(?Necessidade) ^
ufo:isPropositionalContent(?Proposicao,?Necessidade) →
canAchieve(?Servico,?Necessidade)

Def.1.4. – Used-To-Ask-For

IT_Service(?Servico) ^ Need(?Necessidade) ^ Service_Level_Requirement(?SLR) ^
describes_SLR_Need(?SLR,?Necessidade) ^ canAchieve(?Servico,?Necessidade) →
usedToAskFor(?SLR,?Servico)

Def.1.5 – Can-Ask-For

Client(?Cliente) ^ IT_Service(?Servico) ^ Service_Level_Requirement(?SLR) ^
Need(?Necessidade) ^ describes(?SLR,?Necessidade) ^

characterizes_Need_Requestor(?Necessidade,?Cliente) ^ usedToAskFor(?SLR,?Servico) → askFor(?Cliente,?Servico)

Consulta.1 – Servico-Pedido-Por-Cliente

Client(?Cliente) ^ IT_Service(?Servico) ^ Service_Level_Requirement(?SLR) ^ Need(?Necessidade) ^ describes(?SLR, ?Necessidade) ^ characterizes_Need_Requestor(?Necessidade, ?Cliente) ^ usedToAskFor(?SLR, ?Servico) → query:select(?Cliente, ?Necessidade, ?Servico) ^ query:orderBy(?Cliente)

Def.2.1 – HITSP

IT_Service(?Servico) ^ Client(?Cliente) ^ IT_Service_Provider(?Provedor) ^ provides(?Provedor, ?Servico) ^ requests(?Cliente,?Servico) → hires(?Cliente,?Provedor) ^ Hired_ITSP(?Provedor)

Def.2.2 – SLA

Client(?Cliente) ^ Hired_ITSP(?HITSP) ^ hires(?Cliente, ?HITSP) ^ Agreement(?Acordo) ^ mediates_Agreement_ITSP(?Acordo,?HITSP) ^ mediates_Agreement_Requestor(?Acordo,?Cliente) ^ ufo:Normative_Description(?Documento) ^ describes(?Documento,?Acordo) → SLA(?Acordo) ^ SLA_Description(?Documento) ^ isDescribed_SLA_SLADesc(?Acordo,?Documento)

Def.2.3. – Prescribes

Client(?Cliente) ^ IT_Service(?Servico) ^ requests(?Cliente, ?Servico) ^ Service_Level_Requirement(?SLR) ^ usedToAskFor(?SLR, ?Servico) ^ SLA_Description(?Documento) ^ SLA(?Sla) ^ describes_SLADesc_SLA(?Documento, ?Sla) ^ mediates_Agreement_Requestor(?Sla, ?Cliente) → prescribes(?Documento,?Servico)

Def.2.4 – Used-To-Carry-Out

SLA_Description(?Documento) ^ IT_Service(?Servico) ^ prescribes(?Documento, ?Servico) ^ IT_Service_Execution(?Execucao) ^ isInstanceOf_ITSE_ITService(?Execucao,?Servico) → usedToCarryOut(?Documento,?Execucao)

Def.2.5 – Caused-By

SLA(?Sla) ^ SLA_Description(?Documento) ^ describes_SLADesc_SLA(?Documento, ?Sla)
^ IT_Service_Execution(?Execucao) ^ usedToCarryOut(?Documento, ?Execucao) →
isCausedBy_ITSE_Agreement(?Execucao, ?Sla)

Def.2.6 – ITSD

Client(?Cliente) ^ IT_Service_Provider(?Provedor) ^ hires(?Cliente, ?Provedor) ^ SLA(?Sla)
^ mediates_Agreement_Requestor(?Sla, ?Cliente) ^
mediates_Agreement_ITSP(?Sla, ?Provedor) ^ ufo:Plan_Delegation(?Delegacao) ^
ufo:isAssociatedTo(?Delegacao, ?Sla) → IT_Service_Delegation(?Delegacao) ^
ufo:isDelegatorOf(?Cliente, ?Delegacao) ^ ufo:isDelegateeOf(?Provedor, ?Delegacao) ^
ufo:isAssociatedTo_Delegation_SocialRelator(?Delegacao, ?Sla)

Consulta.2 – Provedor-Acordo-Delegacao-de-um-Pedido

Client(?Cliente) ^ IT_Service(?Servico) ^ requests(?Cliente, ?Servico) ^
Hired_ITSP(?Provedor) ^ hires(?Cliente, ?Provedor) ^ Agreement(?Sla) ^
mediates_Agreement_Requestor(?Sla, ?Cliente) ^
mediates_Agreement_ITSP(?Sla, ?Provedor) ^ IT_Service_Execution(?Execucao) ^
isInstanceOf_ITSE_ITService(?Execucao, ?Servico) ^
isCausedBy_ITSE_Agreement(?Execucao, ?Sla) ^ IT_Service_Delegation(?Delegacao) ^
ufo:isAssociatedTo_Delegation_SocialRelator(?Delegacao, ?Sla) ^
ufo:hasDelegator(?Delegacao, ?Cliente) ^ ufo:hasDelegatee(?Delegacao, ?Provedor) →
query:select(?Cliente, ?Servico, ?Delegacao, ?Provedor, ?Execucao, ?Sla) ^
query:orderBy(?Cliente, ?Execucao)

Def.3.1 – IITSP

IT_Service_Execution(?ExecucaoComplexa) ^ ufo:Complex_Action(?ExecucaoComplexa) ^
IT_Service_Provider(?Organizacao) ^ carriesOut(?Organizacao, ?ExecucaoComplexa) ^
IT_Service_Execution(?Execucao) ^
ufo:isComposedOf_CAction_Action(?ExecucaoComplexa, ?Execucao) ^
IT_Service_Provider(?Departamento) ^ carriesOut(?Departamento, ?Execucao) ^
ufo:isComposedOf_SocialAgent_Agent(?Organizacao, ?Departamento) →
Internal_ITSP(?Departamento)

Def.3.2 – IITSP-Service-Manager

IT_Department(?Departamento) ^ IT_Service_Execution(?Execucao) ^
carriesOut(?Departamento, ?Execucao) ^ Hired_ITSP(?Provedor) ^
carriesOut(?Provedor,?Execucao) → Internal_ITSP(?Departamento)

Def.3.3 – Contributes-To

Agreement(?Acordo) ^ IT_Service_Execution(?Execucao) ^
isCausedBy_ITSE_Agreement(?Execucao, ?Acordo) ^ SLA(?Sla) ^
IT_Service_Execution(?ExecucaoComplexa) ^
isCausedBy_ITSE_Agreement(?ExecucaoComplexa, ?Sla) ^
ufo:isComposedOf_CAction_Action(?ExecucaoComplexa, ?Execucao) →
contributesTo(?Acordo, ?Sla)

Def.3.4 – OLA

Agreement(?Acordo) ^ IT_Service_Execution(?Execucao) ^
isCausedBy_ITSE_Agreement(?Execucao, ?Acordo) ^ Internal_ITSP(?IITSP) ^
carriesOut(?IITSP, ?Execucao) ^ SLA(?Sla) ^ IT_Service_Execution(?ExecucaoComplexa)
^ isCausedBy_ITSE_Agreement(?ExecucaoComplexa, ?Sla) ^
Internal_ITSP(?ProvedorInterno) ^ carriesOut(?ProvedorInterno, ?ExecucaoComplexa) ^
ufo:isComposedOf_CAction_Action(?ExecucaoComplexa, ?Execucao) ^
contributesTo(?Acordo,?Sla) → OLA(?Acordo) ^ contributesTo_OLA_SLA(?Acordo,?Sla) ^
Requestor_Provider(?ProvedorInterno) ^
mediates_Agreement_Requestor(?Acordo,?ProvedorInterno) ^
mediates_Agreement_ITSP(?Acordo, ?IITSP)

Def.3.5 – UC

Agreement(?Acordo) ^ IT_Service_Execution(?Execucao) ^
isCausedBy_ITSE_Agreement(?Execucao, ?Acordo) ^ External_ITSP(?EITSP) ^
carriesOut(?EITSP, ?Execucao) ^ SLA(?Sla) ^ IT_Service_Execution(?ExecucaoComplexa)
^ isCausedBy_ITSE_Agreement(?ExecucaoComplexa, ?Sla) ^
Internal_ITSP(?ProvedorInterno) ^ carriesOut(?ProvedorInterno, ?ExecucaoComplexa) ^
ufo:isComposedOf_CAction_Action(?ExecucaoComplexa, ?Execucao) ^

$\text{contributesTo}(\text{?Acordo}, \text{?Sla}) \rightarrow \text{UC}(\text{?Acordo}) \wedge \text{contributesTo_UC_SLA}(\text{?Acordo}, \text{?Sla}) \wedge$
 $\text{Requestor_Provider}(\text{?ProvedorInterno}) \wedge$
 $\text{mediates_Agreement_Requestor}(\text{?Acordo}, \text{?ProvedorInterno}) \wedge$
 $\text{mediates_Agreement_ITSP}(\text{?Acordo}, \text{?EITSP})$

Def.3.6 – ITSD-OLA

$\text{OLA}(\text{?Ola}) \wedge \text{Internal_ITSP}(\text{?ProvedorInterno}) \wedge \text{Internal_ITSP}(\text{?IITSP}) \wedge$
 $\text{mediates_Agreement_Requestor}(\text{?Ola}, \text{?ProvedorInterno}) \wedge$
 $\text{mediates_Agreement_ITSP}(\text{?Ola}, \text{?IITSP}) \wedge \text{IT_Service_Execution}(\text{?ExecucaoComplexa}) \wedge$
 $\text{carriesOut}(\text{?ProvedorInterno}, \text{?ExecucaoComplexa}) \wedge \text{IT_Service_Execution}(\text{?Execucao}) \wedge$
 $\text{carriesOut}(\text{?IITSP}, \text{?Execucao}) \wedge \text{ufo:Plan_Delegation}(\text{?Delegacao}) \wedge$
 $\text{ufo:isAssociatedTo}(\text{?Delegacao}, \text{?Ola}) \rightarrow \text{IT_Service_Delegation}(\text{?Delegacao}) \wedge$
 $\text{ufo:isDelegatorOf}(\text{?ProvedorInterno}, \text{?Delegacao}) \wedge \text{ufo:isDelegateeOf}(\text{?IITSP}, \text{?Delegacao}) \wedge$
 $\text{ufo:isAssociatedTo_Delegation_SocialRelator}(\text{?Delegacao}, \text{?Ola})$

Def.3.7 – ITSD-UC

$\text{UC}(\text{?Uc}) \wedge \text{Internal_ITSP}(\text{?ProvedorInterno}) \wedge \text{External_ITSP}(\text{?EITSP}) \wedge$
 $\text{mediates_Agreement_Requestor}(\text{?Uc}, \text{?ProvedorInterno}) \wedge$
 $\text{mediates_Agreement_ITSP}(\text{?Uc}, \text{?EITSP}) \wedge \text{IT_Service_Execution}(\text{?ExecucaoComplexa}) \wedge$
 $\text{carriesOut}(\text{?ProvedorInterno}, \text{?ExecucaoComplexa}) \wedge \text{IT_Service_Execution}(\text{?Execucao}) \wedge$
 $\text{carriesOut}(\text{?EITSP}, \text{?Execucao}) \wedge \text{ufo:Plan_Delegation}(\text{?Delegacao}) \wedge$
 $\text{ufo:isAssociatedTo}(\text{?Delegacao}, \text{?Uc}) \rightarrow \text{IT_Service_Delegation}(\text{?Delegacao}) \wedge$
 $\text{ufo:isDelegatorOf}(\text{?ProvedorInterno}, \text{?Delegacao}) \wedge \text{ufo:isDelegateeOf}(\text{?EITSP}, \text{?Delegacao}) \wedge$
 $\text{ufo:isAssociatedTo_Delegation_SocialRelator}(\text{?Delegacao}, \text{?Uc})$

Consulta.3 – ITSE-Delegator-Delegatee-Acordo-Delegacao-de-uma-ITSE_Complexa

$\text{IT_Service_Execution}(\text{?ExecucaoComplexa}) \wedge \text{ufo:Complex_Action}(\text{?ExecucaoComplexa}) \wedge$
 $\text{IT_Service_Execution}(\text{?Execucao}) \wedge$
 $\text{ufo:isComposedOf_CAction_Action}(\text{?ExecucaoComplexa}, \text{?Execucao}) \wedge$
 $\text{Agreement}(\text{?Acordo}) \wedge \text{isCausedBy_ITSE_Agreement}(\text{?Execucao}, \text{?Acordo}) \wedge$
 $\text{IT_Service_Delegation}(\text{?Delegacao}) \wedge \text{ufo:isAssociatedTo}(\text{?Delegacao}, \text{?Acordo}) \wedge$
 $\text{IT_Service_Provider}(\text{?Delegador}) \wedge \text{IT_Service_Provider}(\text{?Delegado}) \wedge$
 $\text{ufo:isDelegatorOf}(\text{?Delegador}, \text{?Delegacao}) \wedge \text{ufo:isDelegateeOf}(\text{?Delegado}, \text{?Delegacao}) \rightarrow$

```
query:select(?ExecucaoComplexa,?Execucao,?Delegador,?Delegado,?Acordo,?Delegacao) ^
query:orderBy(?ExecucaoComplexa, ?Execucao)
```

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

ANDRÉ CYPRIANO MONTEIRO COSTA

**MODELAGEM DO PROCESSO
DE GERENCIAMENTO DE NÍVEL
DE SERVIÇO DO PADRÃO ITIL**

**UMA ABORDAGEM USANDO ONTOLOGIAS DE FUNDAMENTAÇÃO
E SUA APLICAÇÃO NA PLATAFORMA INFRAWARE**

**VITÓRIA
2008**

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)