

**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**

**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 DOMÍNIO 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<sup>a</sup>. 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<sup>a</sup>. 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 Infraware. Em outras palavras, adaptar a arquitetura do módulo responsável pelo fornecimento de serviços da Infraware (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 Infraware 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.

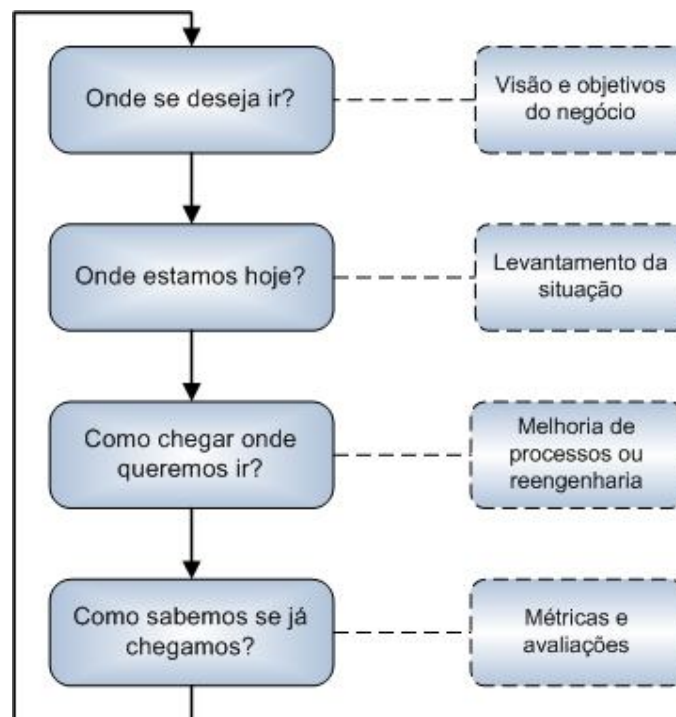


Figura 2-2 - Modelo simples de melhoria de processos [Bon et al, 2006]

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.

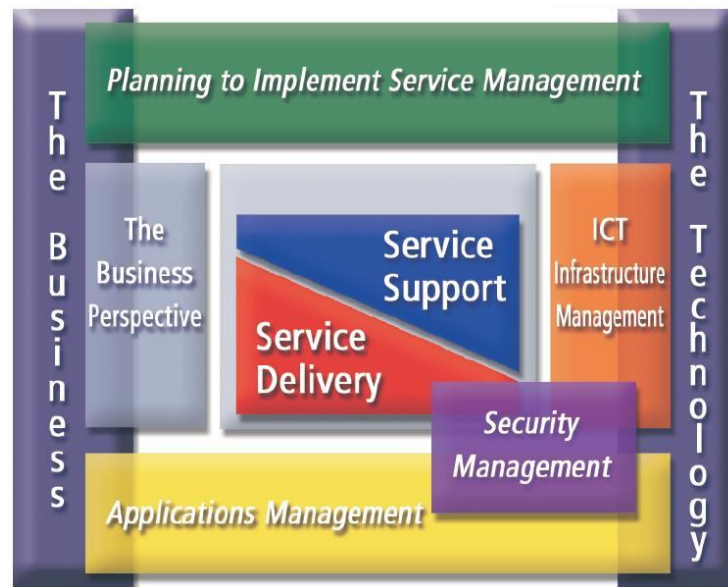


Figura 2-3 - Framework de publicações do ITIL [Macfarlane & Rudd, 2005]

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.

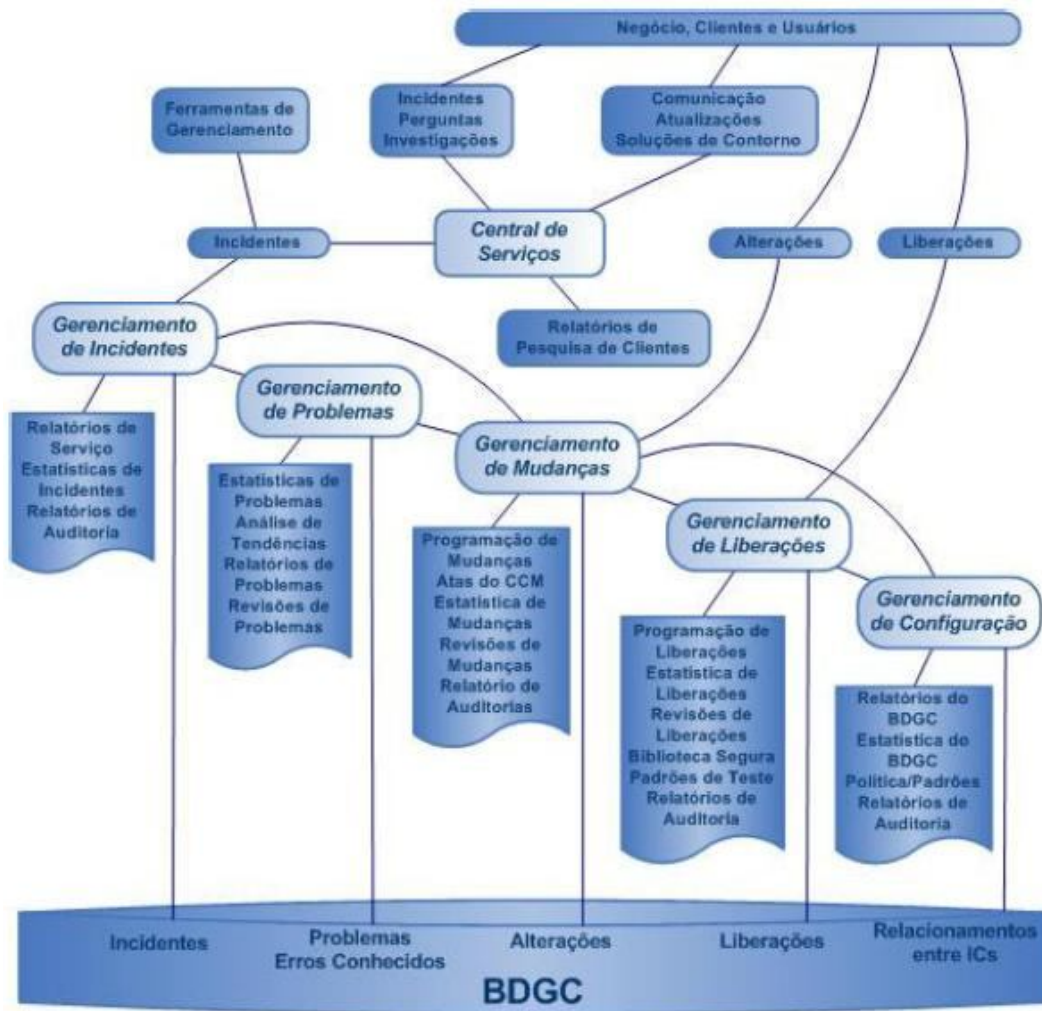


Figura 2-4 - Processos de suporte a serviço do ITIL [Macfarlane & Rudd, 2005]

- 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.

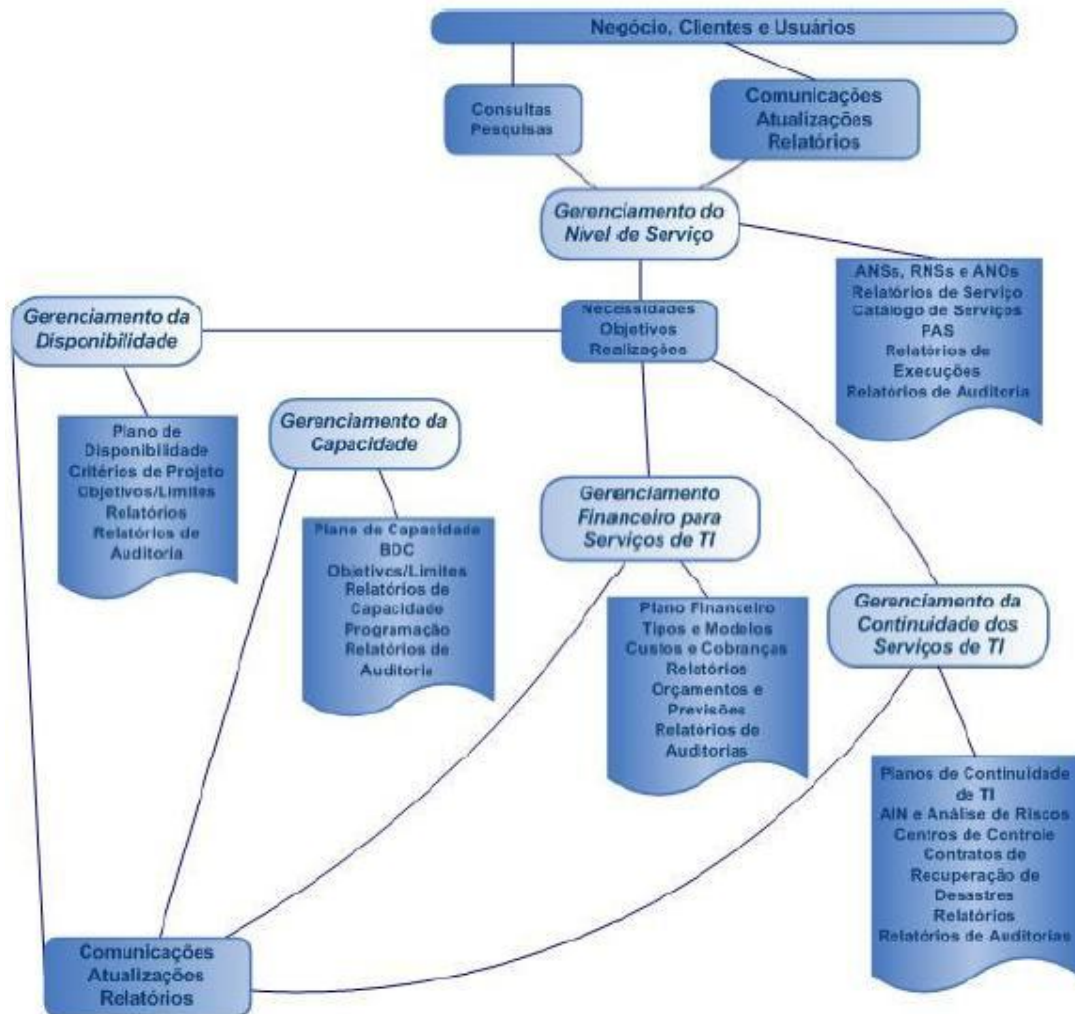


Figura 2-5 - Processos de entrega de serviço do ITIL [Macfarlane & Rudd, 2005]

- 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.

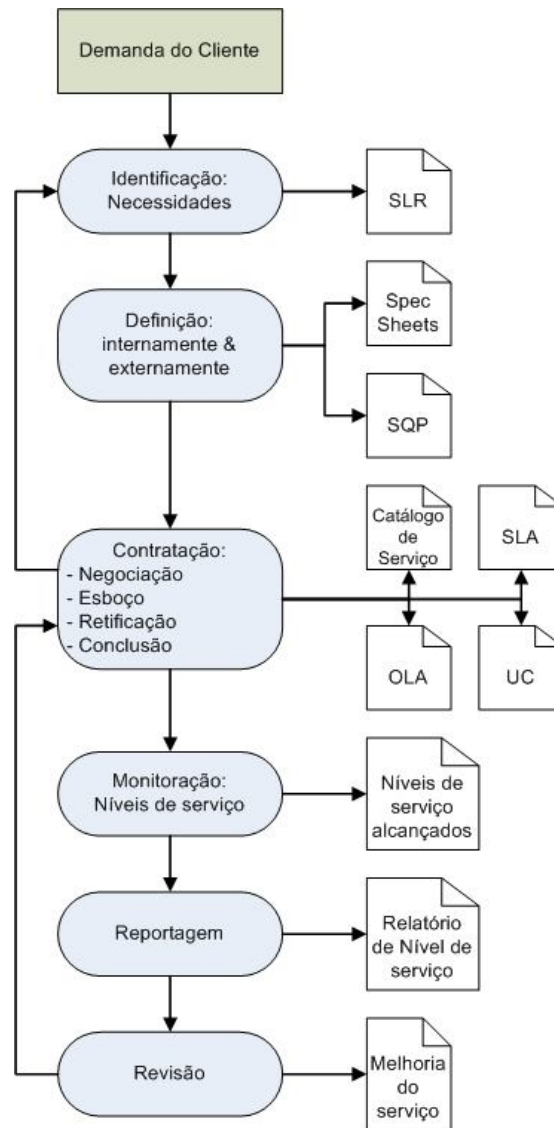


Figura 2-6 - Processo de Gerenciamento de Nível de Serviço [Bon et al, 2006]

## 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.

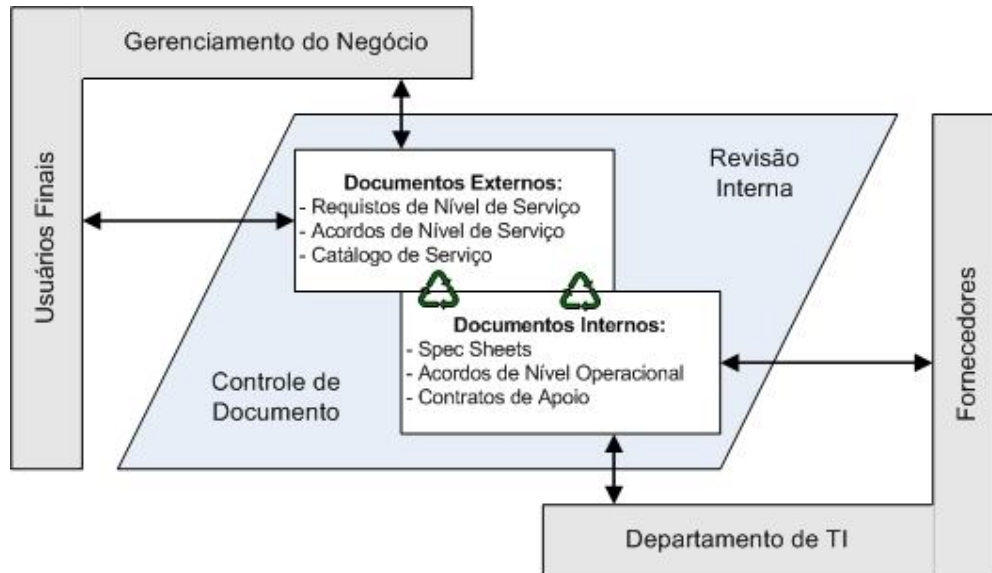


Figura 2-7 - Fase de Especificação [Bon et al, 2006]

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.

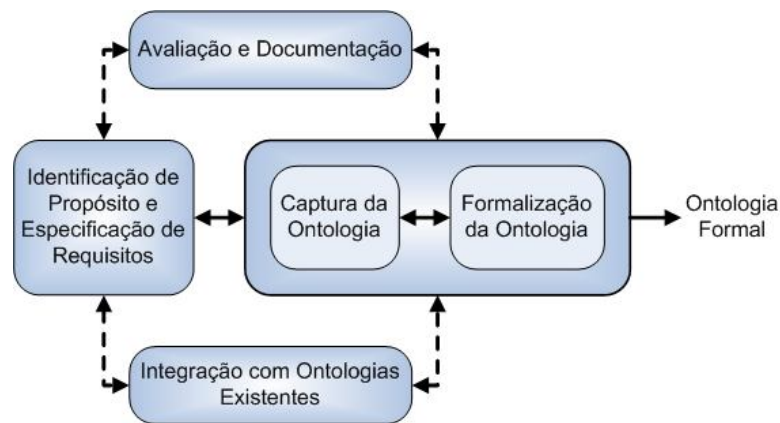


Figura 3-1 - Abordagem do SABiO para o desenvolvimento de ontologias [Falbo:1998]

### 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.

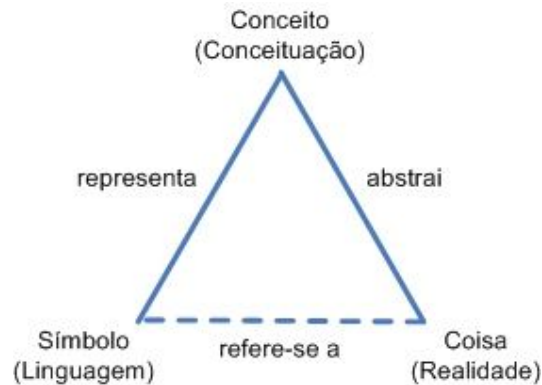


Figura 3-2 - Triângulo de Ullmann: relacionamento entre uma linguagem, uma conceituação e uma porção da realidade

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*.

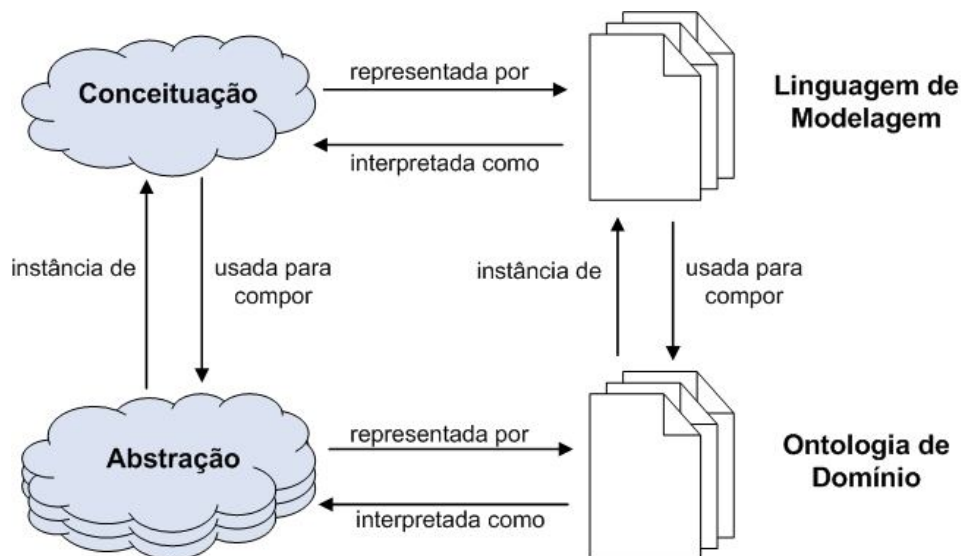


Figura 3-3 - Relacionamento entre conceituação, abstração, linguagem de modelagem e especificação [Guizzardi, 2005]

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.

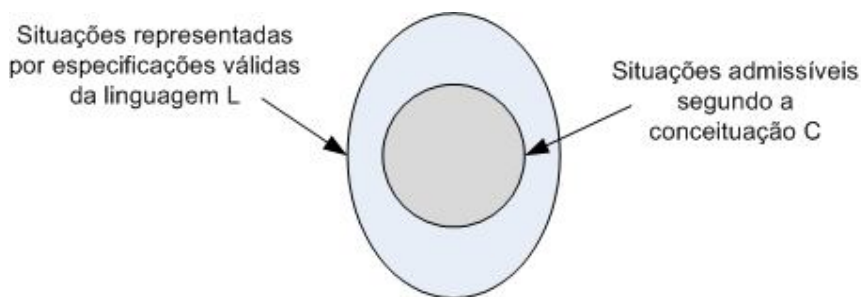


Figura 3-4 - Conseqüência de uma Linguagem de Modelagem imprecisa para a representação da conceituação de um domínio [Guizzardi, 2005]

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.



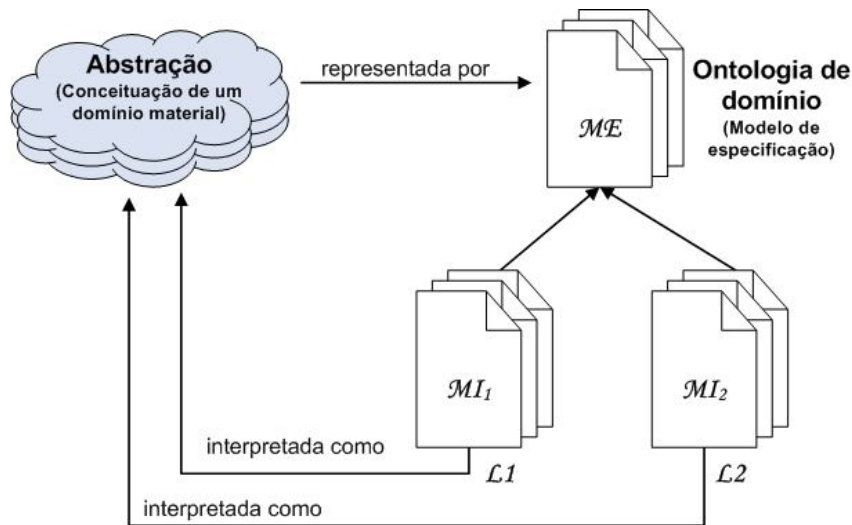


Figura 3-5 - Relacionamento entre ontologia de domínio e modelos de implementação [Guizzardi, 2005]

### 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*.

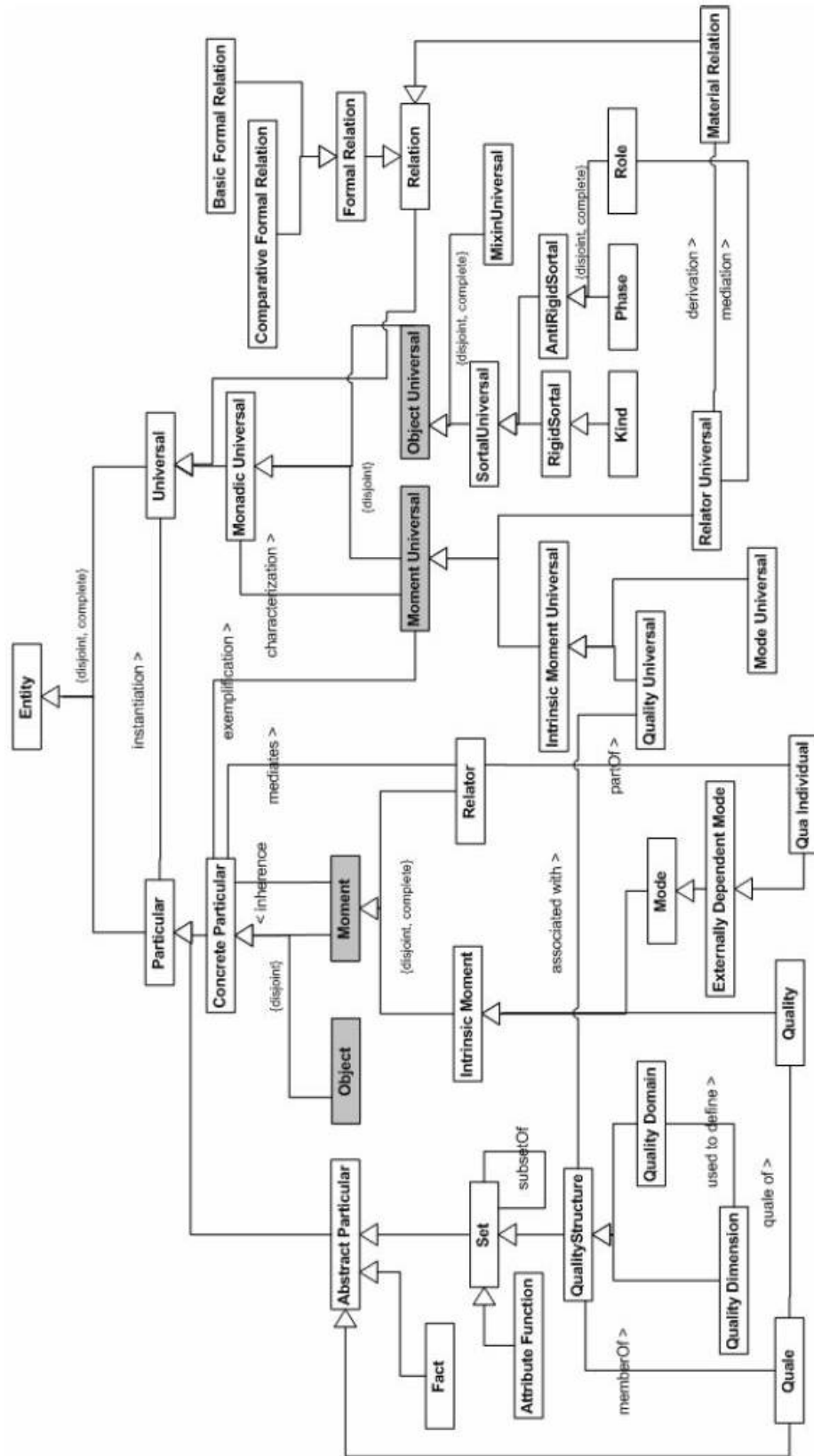


Figura 3-6 - Um fragmento da UFO-A [Guizzardi, 2008]

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.

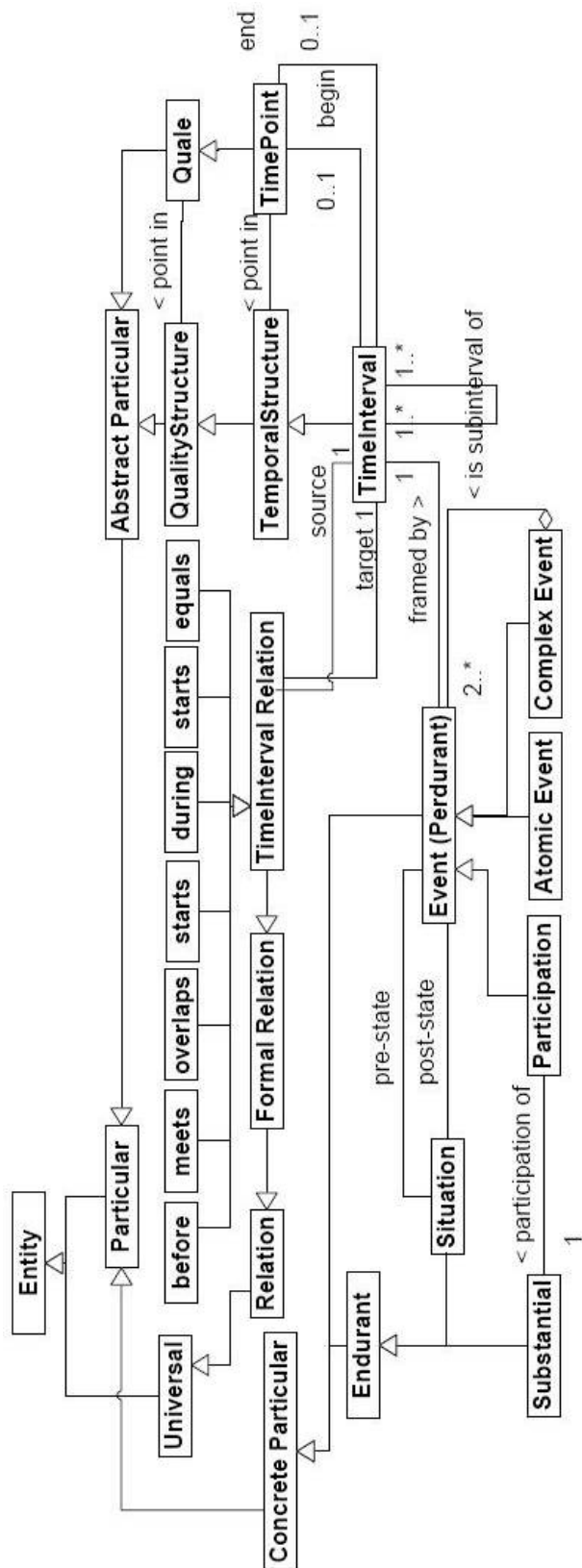


Figura 3-7 - Um fragmento da UFO-B [Guizzardi et al, 2008]

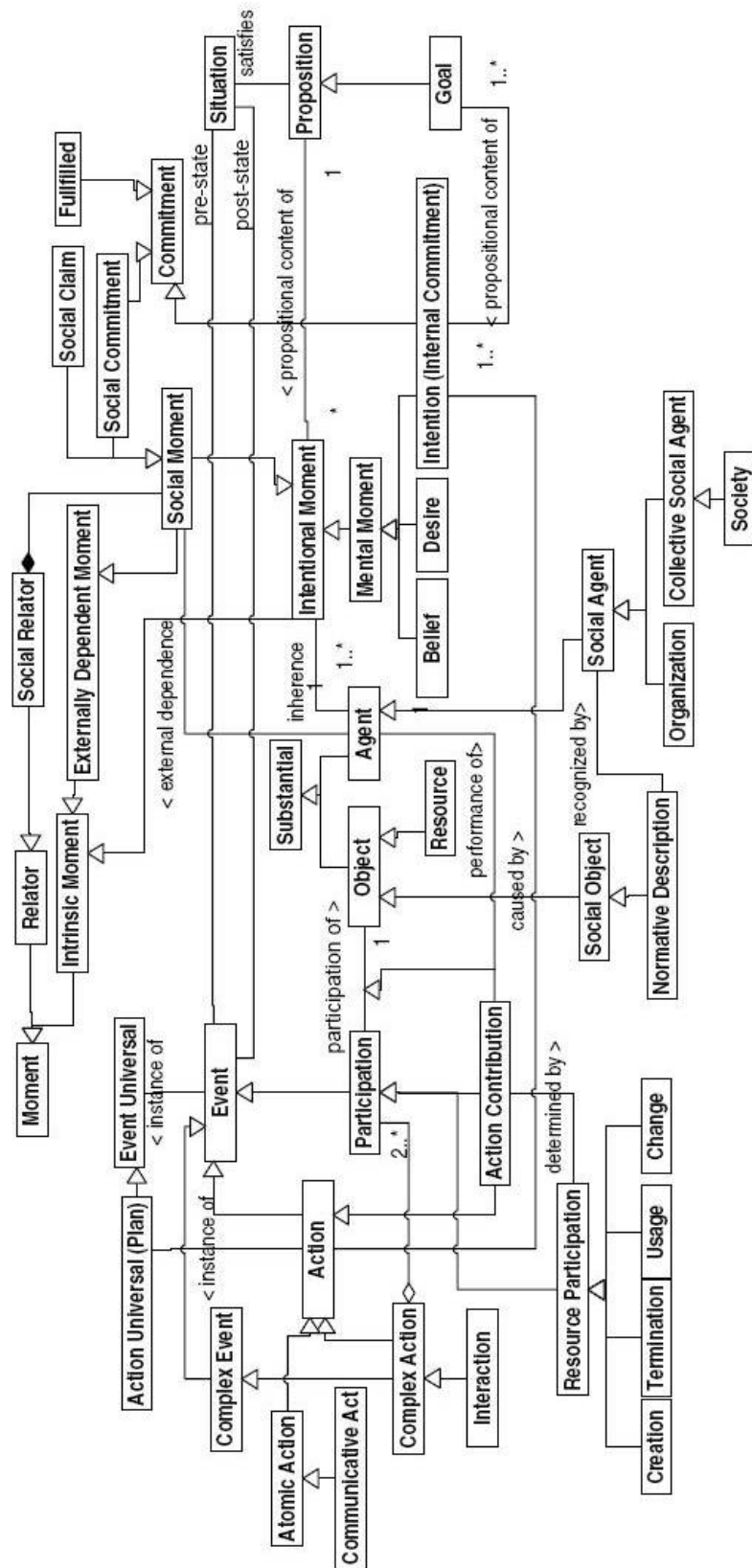


Figura 3-8 - Um fragmento da UFO-C [Guizzardi et al, 2008]

*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*.

Uma primeira diferença que existe, está no fato de uma *dependency* ser uma *formal relation* e uma *delegation* uma *material relation*. Uma *dependency* conecta dois *agents*, chamados de *depender* e *dependee*, e um *dependum* (objeto de dependência), cuja natureza define o tipo de dependência entre os *agents*. Seja *G* um *goal* do *agent A*. Caso *A* não seja capaz de alcançá-lo, mas exista um *agent B* que seja capaz, então tem-se uma relação de dependência entre *A* (*depender*) e *B* (*dependee*) com relação a *G*. O fato de uma *agent* não conseguir concluir seu *goal* significa que, ou tal *agent* não possui a capacidade de alcançá-lo, ou que para alcançá-lo ele tenha que interferir e contradizer suas outras *intentions*. É por isso que o *agent* decide delegar a cumprimento desse *goal* a outro *agent*.

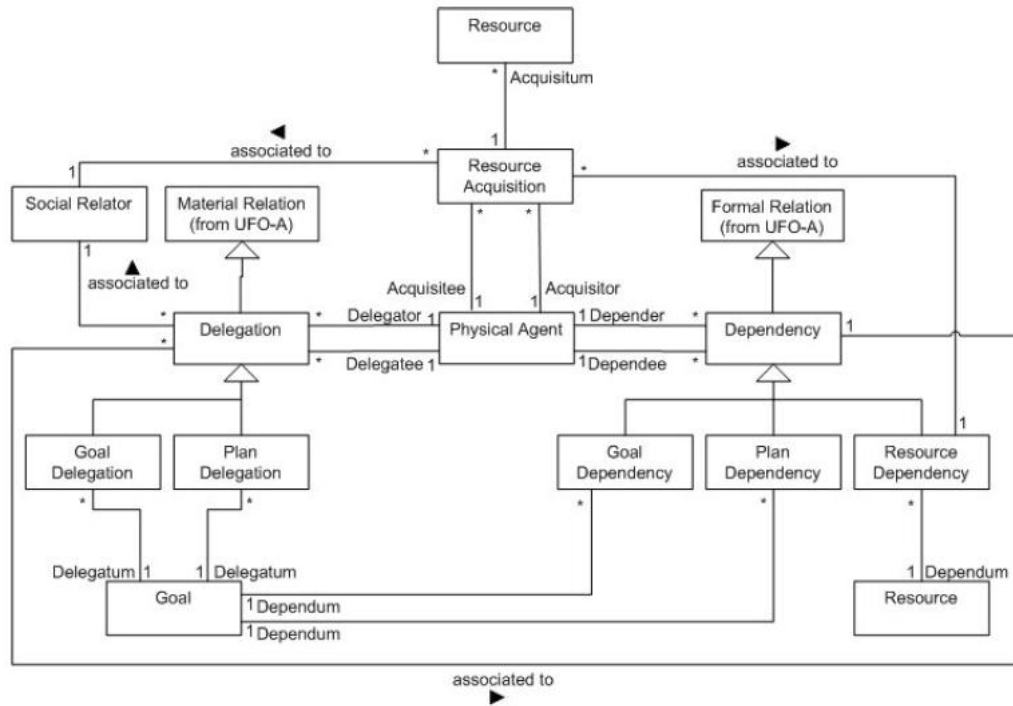


Figura 3-9 - Distinção entre as relações de dependência, delegação e aquisição [Guizzardi & Guizzardi, 2008]

Uma *delegation* é associada a uma dependência, mas por se tratar de uma *material relation*, ela possui um significado mais amplo do que apenas conectar entidades. Uma *delegation* conecta dois *agents* (*delegator* e *delegatee*) a um *goal* (*delegatum*), e a fundamentação dessa relação material é um *social relator*, isto é, os compromissos e reivindicações estabelecidas entre as partes envolvidas nesta *delegation*. Ou seja, quando um *agent A* delega um *goal G* a um *agent B*, além do fato de *A* depender de *B* a respeito de *G*, *B* compromete-se a realizar

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.

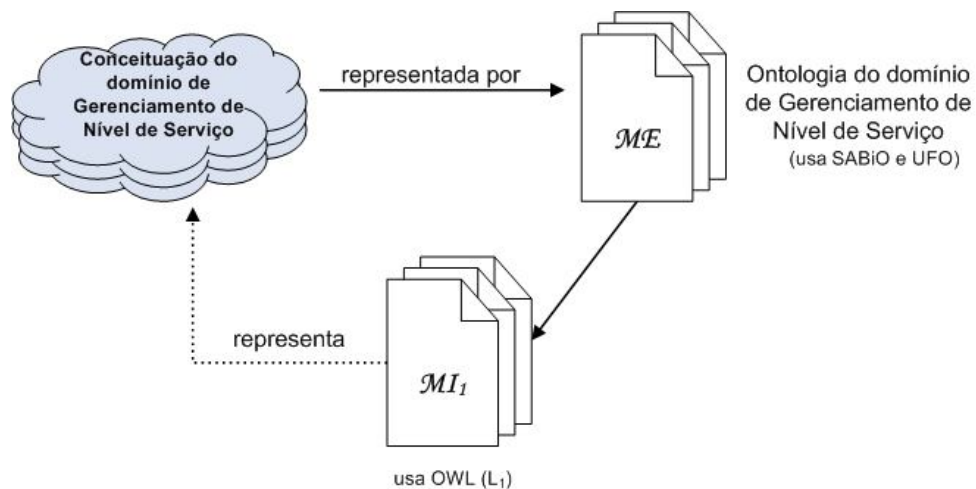


Figura 3-10 - Desenvolvimento e implementação do domínio de Gerenciamento de Nível de Serviço usando ontologias