



Ana Luiza Ávila Cerqueira

**Integração de Ontologia com Modelagem de Processo: Um
Método para Facilitar a Elicitação de Requisitos**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para
obtenção do título de Mestre pelo Programa de Pós-
Graduação em Informática da PUC-Rio.

Orientador: Julio Cesar Sampaio do Prado Leite

Rio de Janeiro, abril de 2007



Ana Luiza Ávila Cerqueira

**Integração de Ontologia com Modelagem de Processo:
Um Método para Facilitar a Elicitação de Requisitos**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Julio Cesar Sampaio do Prado Leite

Orientador
Departamento de Informática - PUC-Rio

Prof. Marco Antonio Casanova

Departamento de Informática - PUC-Rio

Prof. Carlos José Pereira Lucena

Departamento de Informática - PUC-Rio

Prof. José Eugenio Leal

Coordenador Setorial de Pós-Graduação do Centro Técnico Científico
PUC-Rio

Rio de Janeiro, abril de 2007

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Ana Luiza Ávila Cerqueira

Graduou-se em Ciência de Computação pela Unifacs - BA. Durante o mestrado foi bolsista da Vice Reitoria Acadêmica, desenvolvendo trabalho na área de engenharia de software.

Ficha Catalográfica

Cerqueira, Ana Luiza Ávila

Integração de ontologia com modelagem de processo : um método para facilitar a elicitação de requisitos / Ana Luiza Ávila Cerqueira ; orientador: Julio César Sampaio do Prado Leite. – 2007.

239 f. : il. ; 30 cm

Dissertação (Mestrado em Informática)–Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2006.

Inclui bibliografia

1. Informática – Teses. 2. Ontologia. 3. Modelagem de Processo de Negocio. 4. Requisitos. 5. Sistema de Informação. I. Leite, Julio César Sampaio do Prado. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Agradecimentos

Ao meu orientador Professor Julio Cesar Sampaio do Prado Leite pelo estímulo, confiança e conselhos na realização deste trabalho.

À minha família, em especial, meus pais, irmãos e avós, pelo apoio, força e dedicação.

Aos amigos Claudia Cappelli, Daniele Reis, Pedro Asti, Rodrigo Spinola, e em especial ao meu namorado Daniel Prado pelo apoio em todos os momentos do mestrado.

A Fapesb, pelos auxílios concedidos, sem os quais este trabalho não poderia ter sido realizado.

Aos Professores da Comissão Examinadora.

Aos meus colegas da PUC-Rio.

A todos os professores e funcionários do Departamento pelos ensinamentos e ajuda recebida.

Resumo

Cerqueira, Ana Luiza; Leite, Julio **Integração de Ontologia com Modelagem de Processo: Um Método para Facilitar a Elicitação de Requisitos**. Rio de Janeiro, 2007. 239p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A definição dos requisitos do software, onde os engenheiros interagem com os clientes para conhecerem as atividades da organização e entenderem suas necessidades, é um processo chave para o sucesso dos sistemas de informação. Portanto, requisitos definidos de maneira clara e de acordo com as necessidades dos clientes são condições fundamentais para o desenvolvimento de sistemas de informações úteis para a organização. Nessa dissertação propomos um método que visa auxiliar o engenheiro de requisitos na tarefa de elicitação de requisitos mais aderentes ao negócio e com isto conseguir desenvolver melhores sistemas de informação para as organizações. O método proposto parte do princípio que, integrando conhecimentos adquiridos com o modelo de processo do negócio, e o modelo ontológico do negócio, podemos produzir uma sólida fonte de conhecimentos. O uso dessa fonte de conhecimentos na elicitação dos requisitos do sistema de informação da organização objetiva prover aos engenheiros de requisitos uma sólida base para a definição de requisitos aderentes ao negócio da organização. A integração da perspectiva ontológica com a perspectiva de processos é detalhada e exemplificada através de um estudo de caso.

Palavras-chave

Ontologia, Modelagem de Processo de Negócio, Requisitos, sistema de Informação

Abstract

Cerqueira, Ana Luiza; Leite,Julio The integration of business process modeling and business ontology. An method for facilitade requirements elicitation . Rio de Janeiro, 2007. 239p. MSc Dissertation - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Requirements definition, where engineers interact with clients to better know the organization's activities and understand their needs, is a key process to the success of an information system. As such, requirements defined with clarity and according to the client's needs are fundamental to the development of effective information systems for the organization. This dissertation proposes a method to help requirements engineers in the task of eliciting requirements. These requirements must be in accordance with the business as to improve the quality of the organization's information systems. The proposed method integrates process modeling and business ontology. The work assumes that the integration of these perspectives provides a solid source of organization knowledge. Using this knowledge source, requirements engineers are empowered to define requirements that better fits the organization business. The integration of process modeling and business ontology is described in detail and exemplified by a case study.

Keywords

Ontology, Business Process Management, Requirements, Information Systems

Sumário

1 . Introdução	12
1.1. Objetivos do Estudo	13
1.2. Contextualização do Problema	13
1.3. Proposta de Trabalho	15
1.4. Organização da Dissertação	17
2 . Modelagem de Processo de Negócio	18
2.1. Definição e Propósitos	18
2.2. Métodos de Modelagem	21
2.3. Meta-modelos	28
2.4. Formas de Representação	34
2.5. Ferramentas	40
2.6. Conclusão	44
3 . Ontologias	45
3.1. Definição e Objetivos	45
3.2. Métodos para Construção de Ontologia	47
3.3. Meta-modelos de Ontologia	53
3.4. Linguagens	60
3.5. Ferramentas	62
3.6. Conclusão	63
4 . Método Proposto	64
4.1. Conhecendo o Udl	64
4.2. Produzindo o Conhecimento	65
4.2.1. Formas de Obtenção do Conhecimento	65
4.2.2. Iniciando a Construção dos Modelos	67
4.2.3. Construção do Modelo de Processo	67
4.2.4. Construção da Ontologia	72
4.2.5. Validação dos Modelos Construídos	79
4.3. Integrando o Conhecimento	79
4.3.1. Embasamento Teórico	80

4.3.2. Preparação para o Método de Integração dos Conhecimentos	81
4.3.3. Método de Integração dos Conhecimentos	83
4.4. Exibindo o Conhecimento Integrado	95
4.5. Verificando os Modelos Gerados	97
4.6. Conclusão	97
5 . Estudo de Caso	100
5.1. Ambiente do Estudo de Caso	100
5.2. Processo de Construção do Estudo de Caso	100
5.3. Aplicação do Método Proposto no Estudo de Caso	101
5.3.1. Conhecendo o Udl	102
5.3.2. Produzindo o Conhecimento	102
5.3.3. Integrando o Conhecimento	134
5.3.4. Exibindo o Conhecimento Integrado	138
5.3.5. Verificando os Modelos Gerados	144
5.3.6. Conclusão	145
6 . Considerações Finais	146
6.1. Lições Aprendidas	146
6.2. Trabalhos Relacionados	147
6.3. Contribuições	149
6.4. Trabalhos Futuros	151
7 . Bibliografia	153
Anexo 1 – Léxico	158
Anexo 2 – Método de Integração de Conhecimento (Resultados)	163

Lista de Figuras

Figura 1 - Diagrama SADT com as etapas da proposta de trabalho.	16
Figura 2 - Perguntas que auxiliam a construção da visão da organização [Mac Knight 04].	20
Figura 3 - Método para modelar processo [De Bortoli 00].	25
Figura 4 - Elementos do primeiro nível hierárquico do <i>Usual Process</i> (BpProcesso) - Processo [Fiorini 01].	30
Figura 5 - Elementos do segundo nível hierárquico do <i>Usual Process</i> – Atividade Macro [Fiorini 01].	31
Figura 6 - Potencial dos diagramas da UML para modelos de processo - [Cesare 02].	36
Figura 7 - Elementos o BPMN [retirado de White 04].	39
Figura 9 - Notação para os principais elementos do <i>usual process</i> .	71
Figura 10 – Conjunto de informação do meta modelo <i>Frame Ontology</i> .	75
Figura 11 - Informações não identificadas pelas heurísticas.	78
Figura 12 - Ligações entre os modelos organizacionais no EM [Bubenko 94].	80
Figura 13 - Visualização do processo – Ferramenta SRP.	84
Figura 14 - Árvore do processo – Ferramenta SRP.	84
Figura 15 - Visualização da Atividade Macro – Ferramenta SRP.	84
Figura 16 - Atividade detalhada na arvore do processo - Ferramenta SRP.	86
Figura 17 – Item Funções: corresponde ao responsável por executar a atividade macro - Ferramenta SRP.	87
Figura 18 - Tabela auxiliar Função exibindo a descrição da função – Ferramenta SRP.	87
Figura 19 - Entradas das atividades – Ferramenta SRP.	89
Figura 20 - Classes e “Slots” – Ferramenta Protege.	90
Figura 21 - Descrição e relacionamento – Ferramenta Protege.	91
Figura 30 - Classes da Ontologia.	129
Figura 31 - Ontologia da linguagem – Ferramenta Protege.	135
Figura 32 - Navegação pela ontologia da Linguagem – Ferramenta Protege.	136
Figura 33 - Exemplo da Compreensão Inter-relacionada das Entradas.	137
Figura 35 – Documento IEEE preenchidos com resultados do método.	144
Figura 34 – <i>Hierarquia das classes</i> .	238

Lista de Tabelas

Tabela 1 - Características dos métodos para construção de ontologia [adaptado de Perez 04].	53
Tabela 2 – Lista de propriedades da OWL <i>Ontology</i> [retirado de Perez 04].	59
Tabela 3 - Mapeamento da correspondência entre as informações resultantes da aplicação do método e os itens do documento do IEEE.	96
Tabela 4 – Resultado do passo 1 do método de construção de ontologia.	109
Tabela 5 – Resultado do passo 2 do método de construção de ontologia.	110
Tabela 6 - Resultado do passo 3 do método de construção de ontologia.	121
Tabela 7 - Resultado do passo 4 do método de construção de ontologia.	123
Tabela 8 - Resultado do passo 5 do método de construção de ontologia – Classes.	125
Tabela 9 - Resultado do passo 5 do método de construção de ontologia – Propriedades.	126
Tabela 10 - Resultado do passo 6 do método de construção de ontologia.	127
Tabela 11 – Resultados das heurísticas de hierarquia.	131
Tabela 12 – Resultado das heurísticas de relacionamento “Parte-De”.	132
Tabela 13 – Resultado do Passo 1 do método de integração de conhecimento.	165
Tabela 14 - Resultado do Passo 1.1 do método de integração de conhecimento.	167
Tabela 15 - Resultado do Passo 2 do método de integração de conhecimento.	173
Tabela 16 - Resultado do Passo 3 do método de integração de conhecimento.	184
Tabela 17 - Resultado do Passo 4 do método de integração de conhecimento.	199
Tabela 18 - Resultado do Passo 5 do método de integração de conhecimento.	210
Tabela 19 - Resultado do Passo 6 do método de integração de conhecimento.	224
Tabela 20 - Resultado do Passo 7 do método de integração de conhecimento.	231

Tabela 21 - Resultado do Passo 8 do método de integração de conhecimento.

236

1. Introdução

A utilização de sistemas de informação nas organizações tem auxiliado as empresas no mercado competitivo dos dias atuais, na medida em que, facilitando o trabalho diário com a automatização de algumas atividades das empresas, levam as mesmas a aumentar a eficiência, agilidade e confiabilidade dos seus resultados. [Mac Knight 04]

Os sistemas de informação têm a finalidade de auxiliar a organização e os trabalhadores na execução de suas tarefas de maneira mais otimizada para obtenção de melhores resultados [Mac Knight 04]. Contudo, para alcançar sua finalidade, é necessário que os sistemas de informação estejam aderentes ao negócio da organização. Segundo [Leite 01], a deficiência na produção de *software* aderente ao negócio está na falta de atenção para a tarefa de elicitar, definir e acompanhar a evolução dos requisitos durante o processo de construção de *software*.

A fase de elicitação de requisitos é o momento onde o engenheiro conhece o Udl (Universo de Informações) [Leite 01] e as necessidades do cliente. O Udl é o contexto geral no qual o *software* deverá ser desenvolvido e operado, incluindo todas as fontes de informação e todas as pessoas relacionadas ao *software*. [Leite 07] Baseando-se neste conhecimento adquirido e nas necessidades do cliente, o engenheiro inicia a definição dos requisitos que devem ser implementados para a criação do sistema de informação. Portanto, esta fase é de grande importância e deve ter uma atenção especial, pois é a partir dos insumos gerados nesta fase que se baseará a construção do sistema de informação.

1.1. Objetivos do Estudo

Uma grande dificuldade, hoje em dia, para as empresas que desenvolvem *software* é a tarefa de elicitación dos requisitos com seus clientes, principalmente pelo fato de que esta tarefa demanda muito tempo, o que gera um aumento de gastos e, conseqüentemente, influencia no preço final do *software* [Monteiro 04]. A elicitación é tarefa difícil de ser executada quando o cliente não conhece a fundo seus processos de trabalho, pois gera confusões no entendimento do negócio, tendo como conseqüência serviços ou *softwares* de baixa qualidade.

A descoberta dos requisitos de um sistema é uma das mais importantes tarefas do desenvolvimento de *software*. Se os requisitos não estiverem alinhados às necessidades do negócio, levarão à construção de um sistema que não atenderá às expectativas dos seus usuários. [Mac Knight 04]

Para que uma organização possa obter cada vez mais *softwares* aderentes aos requisitos elicitados, é necessário um melhor entendimento dos conceitos, definições, regras e modelos que estão por trás dos mesmos. Quanto mais conhecimento existir sobre o Udl do negócio para o qual se quer elicitare requisitos, mais qualidade terá o resultado obtido na produção do *software*. [Monteiro 04]

1.2. Contextualização do Problema

As empresas, devido ao seu grande volume de informação, necessitam de *softwares* de qualidade para que, através destes, possam disponibilizar informação de qualidade, para auxiliar nas tomadas de decisão e nas operações diárias de uma empresa. Construir um *software* de qualidade é garantir que o mesmo seja aderente às necessidades do negócio, além de possuir informações coerentes com o negócio, e consistentes entre si. [Leite 01]

O desenvolvimento destes *softwares* para empresas, os sistemas de informação, devido à grande quantidade de informação e o fato das informações serem disponibilizadas através das pessoas da empresa, acabam dificultando e aumentando o trabalho do engenheiro de requisitos, na medida em que as informações estão “espalhadas” pelos departamentos. As pessoas da empresa,

muitas vezes, podem ter interpretações diferentes sobre a mesma informação, e a falta de atenção a estas diferentes interpretações podem causar erros no entendimento dos requisitos do sistema de informação. [Cota 04]

As técnicas de elicitação de requisitos têm como finalidade facilitar a vida do engenheiro de requisitos na primeira das quatro áreas de conhecimento (elicitação, modelagem, análise e gerência) em que se divide a engenharia de requisitos [Neto 00]. Porém, as técnicas de elicitação têm problemas, e um deles é a dificuldade de uniformização do conhecimento, devido às dificuldades do usuário em definir suas necessidades, o que torna uma tarefa difícil e demorada a busca do entendimento das necessidades do negócio e, por conseqüência, a compreensão dos requisitos do sistema de informação. [Goguen 93].

Entendendo melhor o negócio e compreendendo o “conhecimento” que transita nos processos da empresa, é possível ajudar na elicitação dos requisitos mais aderentes às necessidades do negócio, para, com isto, ter a possibilidade de construir um melhor *software* para apoiar o processo de negócio da empresa. Para entender melhor o negócio da empresa é recomendável que seus processos de negócio sejam bem entendidos e modelados. Para compreender o conhecimento que transita nestes processos, é necessário um modelo que represente o conhecimento deste Udl (Universo de Informações). Acreditamos que a modelagem de processos de negócio, juntamente com a criação de uma ontologia, pode prover estes modelos que auxiliariam numa uniformização inicial do conhecimento. Com esta uniformização poderemos facilitar o trabalho dos engenheiros de requisitos na busca pelos requisitos do sistema de informação, que são necessários para apoiar os processos de negócio da mesma. Além disto, com a uniformização do conhecimento através destes modelos, é mais fácil garantir a aderência dos requisitos às necessidades do negócio, minimizando erros no entendimento do negócio.

Resumidamente, podemos dizer que o problema é a dificuldade, a demora, e as distorções no entendimento das informações na fase de elicitação de requisitos, através na busca e na definição das necessidades do negócio. A modelagem de negócio juntamente com a ontologia procuram atacar este problema de contextualização e da uniformização da informação buscando facilitar o trabalho do engenheiro de requisitos.

1.3. Proposta de Trabalho

A proposta de trabalho é estudar as áreas de modelagem de processos de negócio e ontologias, além de propor uma maneira de juntar estas áreas de forma a criar um “método” que facilite o processo de elicitação de requisitos. O que minimizaria os problemas de distorção no entendimento das necessidades e requisitos dos sistemas de informação e otimizaria o tempo da elicitação.

A idéia é estudar a modelagem de processo de negócio com o foco na obtenção de um modelo que represente, de maneira clara e objetiva, o processo de negócio da empresa. O estudo da ontologia visa encontrar uma representação para o conhecimento que envolve o negócio. Com estes dois modelos construídos, propomos uma maneira de extrair e unir estes conhecimentos de forma a facilitar o trabalho na elicitação de requisitos.

Um estudo de caso será utilizado para demonstrar a junção destes modelos e os resultados adquiridos com a aplicação do método. Estes resultados servirão para mostrar a utilidade prática da aplicação do método num contexto do mundo real, bem como serão fonte de “*feedback*”, que podem ser utilizados no aperfeiçoamento do método.

Abaixo detalhamos os conceitos principais envolvidos na proposta e suas breves descrições.

1. Modelagem de Processo de Negócio

A modelagem de processo de negócio visa estabelecer práticas de mapeamento de processos, em que, através da identificação de papéis, responsabilidades, produtos, atividades, tarefas, casos de uso e ferramentas para a execução, deve se tornar possível identificar o conjunto de informações que é útil para a compreensão e identificação das necessidades de negócio.

2. Ontologia

Para o conhecimento e aprendizado de um Udl (Universo de Informações), se faz necessário o mapeamento dos conceitos utilizados neste Udl de forma explícita. Uma Ontologia ajuda a melhor compreender uma área de conhecimento, permite um consenso entre grupos sobre um determinado Udl e também deixa explícito, em forma de conhecimento formal, informações que poderão ser divididas com qualquer outro grupo.

Uma ontologia é uma maneira de se conceitualizar, de forma explícita e formal, os conceitos e restrições relacionados a um Udl de interesse [Noy 01]. Assim, ontologia é um modelo abstrato com o intuito de representar conceitos de um determinado Udl de forma clara e objetiva.

Este trabalho seguiu as seguintes etapas mostradas na Figura 1:

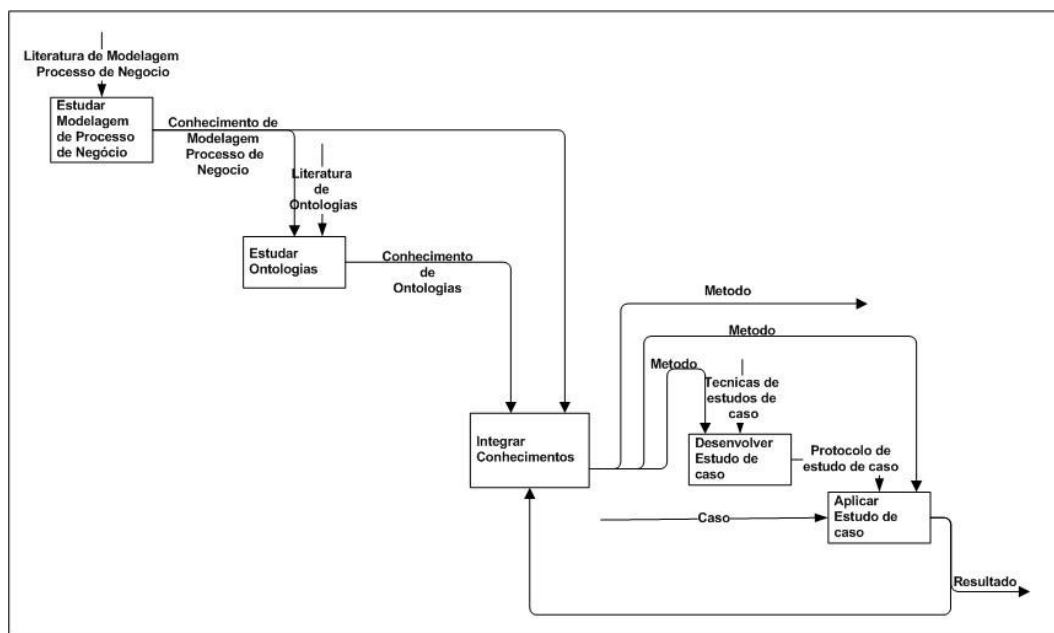


Figura 1 - Diagrama SADT com as etapas da proposta de trabalho.

1.4. Organização da Dissertação

Esta dissertação apresenta, nos capítulos 2 e 3, um estudo teórico sobre os principais conceitos utilizados nesta dissertação, modelagem de processo de negócio e ontologias, respectivamente. Neste estudo, apresentamos: definições e objetivos dos conceitos, métodos para construção dos modelos, ferramentas de apoio ao desenvolvimento e linguagens de representação.

O quarto capítulo é considerado o central desta dissertação, pois é onde apresentaremos o método proposto para integrar os conhecimentos obtidos na modelagem de processo de negócio e na ontologia do Udl do negócio.

O quinto capítulo apresenta um estudo de caso, realizado num ambiente do mundo real, que tem o intuito de mostrar a aplicação do método proposto numa organização, bem como expor as contribuições da aplicação do mesmo.

No sexto capítulo, exibiremos as conclusões deste trabalho, além das contribuições esperadas e os trabalhos futuros.

2. Modelagem de Processo de Negócio

Este capítulo apresenta um estudo sobre modelagem de processo de negócio. Este estudo exibirá a definição de modelagem de processo de negócio e seus propósitos, bem como uma pesquisa em torno de alguns métodos existentes para construção de modelos de processos. Abordaremos também os meta-modelos para processos de negócio, linguagens para modelagem do processo de negócio e ferramentas utilizadas para apoiar a construção dos modelos.

2.1. Definição e Propósitos

A modelagem de processo de negócio é um conjunto de atividades ordenadas, ou seja, é uma ordenação do trabalho, com entradas e saídas bem definidas; com atores responsáveis pela execução de tarefas, com objetivo de modelar como é executado o “trabalho” da organização.

Para [Sharp 00], modelagem de processo é:

“Collection of interrelated work tasks, initiated in response to an event, that achieves a specific result for the customer of the process”

Segundo este autor a modelagem de processo de negócio é uma maneira de organizar o trabalho e os recursos, sejam eles pessoas, equipamentos ou informações, no sentido de atingir os objetivos da organização.

Esta visão de organização do trabalho para atingir os objetivos é a mudança de um paradigma histórico, em que as empresas organizavam o trabalho e os recursos em funções e especialidades [Sharp 00]. Antigamente, o negócio das empresas era visto de forma vertical, ou seja, separados por funcionalidades, áreas ou departamentos, cada qual com suas atividades e objetivos. Nos dias de hoje a visão do negócio foi alterada para uma visão horizontal, uma visão de processos, onde as atividades de um mesmo processo

podem ser realizadas em diversas áreas ou departamentos buscando atingir os objetivos da empresa.

Para atingir seus objetivos, é necessário que as organizações conheçam com mais precisão seu negócio, para entender quais são e como são executadas suas principais funções; e, desta forma, conseguir fornecer melhores resultados, definindo processos de trabalho mais eficientes. [Mac Knight 04]

O propósito da modelagem é fornecer uma visão do negócio da organização, que muitas vezes é difícil de “enxergar” sem apoio de um modelo, devido a sua complexidade. Assim, o modelo de negócio serve para que a empresa possa melhor compreender seu negócio através de uma abstração da realidade.

Para [Cruz 04], o propósito da modelagem de processo é fornecer uma perspectiva simplificada da estrutura do negócio, como um meio para atingir o fim, permitindo comunicar, documentar e entender as atividades da organização. Assim, a modelagem pode ser utilizada para melhoria e reengenharia dos processos.

Abaixo listamos alguns dos propósitos da modelagem de processo segundo [Paim 02]:

- Representar ou entender como uma organização funciona (ou alguma parte dela).
- Utilizar/explicitar o conhecimento adquirido e a experiência para usos futuros.
- Racionalizar e assegurar o fluxo de informações.
- Projetar ou reprojeter e especificar uma parte da organização (aspecto funcional, comportamental, informacional, organizacional ou estrutural).
- Analisar alguns aspectos da organização (análise organizacional, qualitativa e outras).
- Simular o comportamento de algumas partes da organização.

Concluindo, para [Mac Knight 04] os objetivos do modelo de processo de negócio se resumem em seis perguntas sobre a organização (5W1H), cujas respostas produzem grandes insumos, que auxiliam na construção do modelo da organização e seu negócio:

- O que é feito?
- Quem faz?
- Quando?
- Onde?
- Por quê?
- Como?

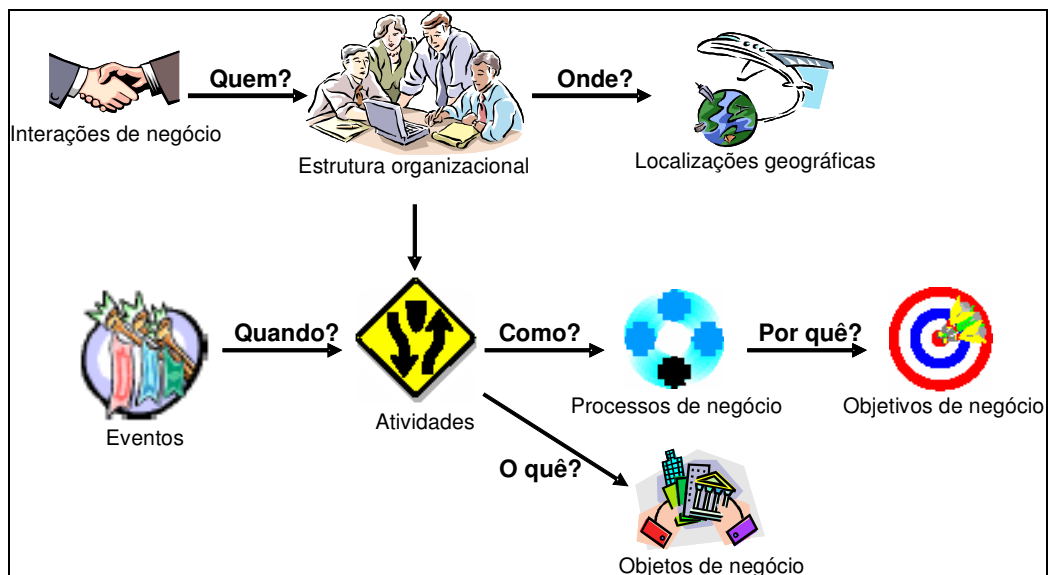


Figura 2 - Perguntas que auxiliam a construção da visão da organização [Mac Knight 04].

Nas próximas seções deste capítulo, apresentaremos métodos para construção de modelos de processo, meta-modelos - no qual o modelo de processo deve basear-se -, linguagens para representação do processo, e por fim, ferramentas que apóiam a construção dos modelos.

2.2. Métodos de Modelagem

A definição de um método é a maneira como deve ser feita, ou executada, uma atividade; é um passo-a-passo. Segundo [Paim 02], “*um método pode ou não ter uma fundamentação teórica. Em geral, um método é o resultado apurado das melhores práticas em um domínio particular de uma dada atividade.*”.

Abaixo citamos alguns dos métodos para modelagem de processos pesquisados:

- **Método Orientado à Workflow** [Sharp 00]

Este método está dividido em 4 fases:

A. Emoldurar o Processo

Esta fase é muito importante, pois é a fase na qual estruturamos os processos do negócio, que serão mapeados.

Esta fase inclui as seguintes atividades:

- Identificar o conjunto de processos, que inclui tanto os processos alvos, que são o foco da modelagem, quanto os processos que os rodeiam. Esta atividade ajuda a clarificar os processos que estão dentro e fora do escopo da modelagem.
- Identificar os processos alvo e seus limites.
- Revisar ou documentar a missão, estratégia e objetivos da organização.
- Mapear uma prévia estimada do processo.
- Mapear uma prévia do processo aperfeiçoado.
- Desenvolver um glossário de termos do Udl.
- Opcionalmente, iniciar uma documentação sobre as observações da cultura, competências.

B. Entender o Processo Atual (AS-IS)

Esta é uma fase basicamente de documentação do processo atual, com vistas a entender as atividades do processo e de que forma estas estão atingindo, ou não, os objetivos da organização.

Esta fase possui as seguintes atividades:

- Mapear o fluxo de trabalho do processo corrente. Visando identificar “quem faz o que”, o autor sugere utilizar os fluxogramas organizacionais.
- Investigar os “outros objetivos” da organização como, por exemplo, utilização de algum sistema já existente na organização, políticas a seguir, metas de medições para as atividades.
- Mapear uma avaliação final dos “outros objetivos” da organização, como suas regras, políticas, sistemas, motivações.
- Documentar aspectos importantes de cultura, competências e sistemas de gerência.

C. Projetar o Novo Processo (TO-BE)

Esta fase está subdividida em dois estágios:

C.1. Caracterizar o Novo Processo (TO-BE)

Este estágio objetiva definir o conjunto de melhorias a serem implementadas para atingir os objetivos da organização.

Este estágio é de grande importância, no sentido de evitar inconsistência no projeto do fluxo de trabalho do novo processo, porque, neste momento, é realizada a análise dos objetivos da empresa em conjunto com os “outros objetivos”, mapeados na fase anterior, para definir o conjunto de melhorias que deverão ser implementadas no novo processo.

Abaixo as atividades realizadas neste estágio:

- Decidir a direção do novo processo (Continuar como está? Realizar melhorias? Refazer o processo?).
- Desenvolver idéias de novas funcionalidades para o novo processo, baseadas nos objetivos em conjunto com os “outros objetivos”.

- Montar uma matriz com os objetivos e “outros objetivos”, e mapear as idéias das novas funcionalidades que o atendem.
- Avaliando a matriz, selecionar novas funcionalidades chaves para o novo processo.

C.2. Projetar o *Workflow* do Novo Processo (TO-BE)

Neste estágio é desenvolvido o *workflow* do novo processo.

D. Desenvolver os Cenários dos Casos de Uso

Esta fase tem como objetivo visualizar como os sistemas poderão suportar os processos.

As seguintes atividades são executadas:

- Identificar os cenários de casos de uso, analisando as atividades do *workflow* do novo processo.
 - Desenvolver os cenários dos casos de uso.
- **“Método de Modelagem para *Workflow*”** [De Bortoli 00]
O método de modelagem de processo da autora divide-se em 4 fases:

A identificação do Udl (universo de informações) [Leite 01].

A Elicitação, onde é realizada a aquisição do conhecimento da seguinte forma:

- Definir o contexto onde a aquisição de conhecimento vai ocorrer; o Udl com suas fontes de informação.
- Coleta de informações, realizada através de entrevistas, leitura de documentos.
- A partir desta coleta, identificar os objetos iniciais da modelagem: atores, documentos, atividades, tarefas, manuais e sistemas.

- Utilizar a técnica 5W1H (*Who?*, *When?*, *Where?*, *What?*, *Why?* e *How?*) para definir um conjunto específico de questões para cada objeto identificado.
- Buscar as respostas para o conjunto de questões definidos no passo anterior.
- Construir o LAL – Léxico Ampliado da Linguagem [Leite 93] do ambiente do negócio, para facilitar o entendimento e a comunicação com os atores do ambiente.

A modelagem, onde as representações textuais produzidas na elicitación são projetadas no fluxo de trabalho:

- Construir um Diagrama de Fluxo de Decisão da Atividade (DFDA) para cada atividade. Este diagrama representa um fluxo de trabalho de um processo, ou seja, o sequenciamento de suas atividades. No DFDA, as atividades são como processos, ou seja, para detalhar as atividades, construímos um novo DFDA para cada uma delas.

A validação, onde o fluxo de trabalho é validado com os atores do ambiente. Com a validação, novos conhecimentos podem ser adquiridos e devem estar refletidos no fluxo de trabalho.

A Figura abaixo resume o método proposto por [De Bortoli 00] :

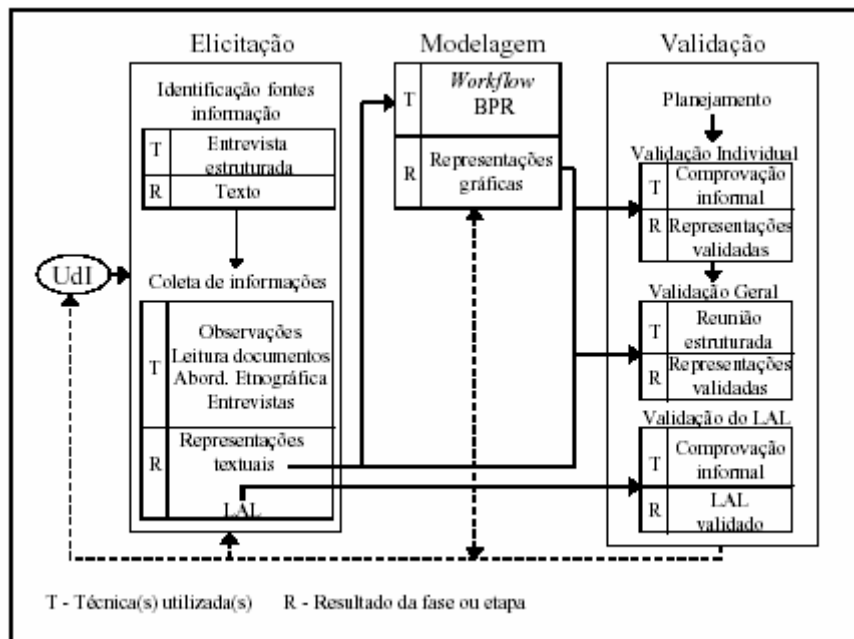


Figura 3 - Método para modelar processo [De Bortoli 00].

- **BPM Utilizado pelos Criadores do ARIS** [IDS Scherr 05]

Este método de BPM (*Business Process Modeling*), dependendo do projeto ou dos objetivos da empresa, pode ter passos modificados ou até mesmo omitidos. Abaixo apresentamos suas fases:

A. Estratégia de Processo de Negócio.

Esta é uma fase onde, baseados no entendimento do negócio, definimos o que é essencial para a organização. Para isto será desenvolvido um mapeamento dos processos da organização. Abaixo apresentamos as atividades desta fase:

- Realizar *workshops* para entender melhor o negócio e definir o escopo do modelo de processo. O autor sugere a construção de uma matriz com as informações dos produtos oferecidos pela organização, e seus respectivos mercados, e o que cada um de seus processos contribui no desenvolvimento do produto. A análise desta matriz deve ser feita com foco nos fatores

críticos de sucesso (satisfação do cliente, entrega no tempo, produtos de qualidade), comparando seu desempenho com o de outras empresas do mesmo ramo.

- Classificar os processos nas seguintes categorias: Processo de Gerenciamento, Processos Fundamentais e Processos de Suporte.
- Priorização dos processos, baseada na performance dos processos e nos fatores críticos de sucesso já analisados.
- Para os processos-foco da organização, descobertos através dos passos anteriores, determinar seus objetivos.
- Realizar a comunicação na organização dos objetivos dos processos e o que será afetado na trabalho dos funcionários.

B. Projeto de Processo de Negócio.

Esta é uma fase de documentação, na qual definimos o mapeamento dos processos com seus objetivos, analisando pontos para melhoria. Para documentação dos processos devem ser mapeados:

- Informações das atividades;
- Papéis envolvidos nas atividades;
- Entradas e Saídas das atividades;
- Sistemas envolvidos nas atividades;
- Objetivos;
- Atividade com potencial para melhoria e ações que devem ser realizadas;
- Definir medições para mensurar as melhorias que serão realizadas;
- Definir plano de implementação das melhorias.

C. Implementação de Processo de Negócio.

Nesta fase é realizada a implementação das melhorias planejadas no passo anterior, para estabelecer uma efetiva e eficiente organização do processo:

- Realizar adaptação da estrutura organizacional:
 - Definir um “dono” para cada processo. Esta pessoa será responsável por efetivar (implantar os objetivos) eficientemente (alcançando os objetivos) o processo de negócio.
 - Este “dono” deve estabelecer o foco do processo na organização, obter e desenvolver recursos para que o processo seja otimizado.

D. Controle de Processo de Negócio.

Esta atividade objetiva monitorar continuamente os processos da organização, oferecendo as medições dos processos, que poderão ser utilizadas para tornar a organização mais eficiente.

Após exibir a descrição, ou melhor, o passo-a-passo de alguns métodos para modelagem de processo, é importante salientar que, apesar de existirem métodos, “o mundo do processo de negócio” é cheio de políticas, conflitos de objetivos, velhos hábitos e obstáculos. Portanto, é importante que, independente do método selecionado, o engenheiro se aproxime bastante do ambiente da organização, para que possa conhecer bem estas variantes. [Sharp 00]

Analisando os métodos apresentados, notamos que o método proposto por [Sharp 00] tem como vantagem a simplicidade. O segundo método, apesar de também ser simples, não faz nenhuma alusão à elicitação dos objetivos da organização para auxiliar no mapeamento e entendimento dos processos de negócio da organização. Consideramos que este ponto poderá trazer prejuízos ao desenvolvimento do modelo de processos, visto que o engenheiro estará executando um trabalho mecânico e baseado nas informações que receber da organização além de não ter como analisar a coerência das informações e dos processos mapeados, visto que não tem mapeados os objetivos da organização.

O terceiro e último método apresentado é bastante rico em informações, porém, não é de fácil aplicação, ou melhor, é necessária uma infra-estrutura para sua aplicação, como pode notar-se na descrição de alguns de seus passos [IDS Scherr 05]. São necessários, por exemplo, resultados de medições de performance, pesquisa de *benchmarking*.

A escolha de um método deve ser baseada nos resultados que se deseja obter com a utilização do mesmo, e para propósito deste trabalho desejamos que o método seja de fácil aplicação e aprendizado, e produza modelos de fácil entendimento, para conseguirmos atingir um maior número de organizações. Logo, optamos pelo primeiro método para ser utilizado neste trabalho.

2.3. Meta-modelos

Os Meta-modelos são necessários no desenvolvimento de um modelo, pois guiam o seu desenvolvimento, mostrando as informações que devem ser especificadas para que o modelo esteja completo.

Para definição de processo existem diversos meta-modelos. Abaixo detalhamos dois desses:

- **O Meta-modelo para Fluxo de Trabalho** [Georgakopoulos 95]

Este meta-modelo [Georgakopoulos 95] tem o intuito de mapear os atores e suas atividades dentro do processo da organização. Possui os seguintes elementos, ou seja, informações que devem ser especificadas:

- Fluxo de Trabalho: São um conjunto de tarefas ordenadas. Os fluxos de trabalho podem ser utilizados dentro do fluxo de trabalho que está sendo modelado, com intuito de detalhar as atividades especificadas.
- Tarefas: São as operações ou a descrição de ações humanas.
- Objetos Manipulados: São documentos, imagens, dados.
- Papéis: São habilidades humanas ou serviços de sistemas necessários para um trabalho ou atividade.

- Atores: São pessoas ou sistemas de informações que satisfazem os papéis, executam as tarefas e interagem durante a execução do *workflow*.

- **Usual Process** [Fiorini 01]

O *usual process* é um tipo de processo encontrado na maioria dos processos existentes. Ele foi desenvolvido com base numa arquitetura para organização e descrição dos processos visando a reutilização de processo, ou seja, a utilização de um modelo de processos na criação de outros. Segundo [Fiorini 01], *usual process* é definido:

“É qualquer processo existente que nem é um process standard nem é um process pattern. Ele não é um process standard porque não tem a finalidade normativa. Não é um process pattern porque ele não necessariamente precisa ter sido testado (aplicado), em um considerável número de vezes, para solucionar um problema recorrente.”

O *usual process* é organizado em três níveis hierárquicos (Processo, Atividade Macro e Atividade Detalhada) e têm como elementos, para cada nível, os itens exibidos no diagramas abaixo:

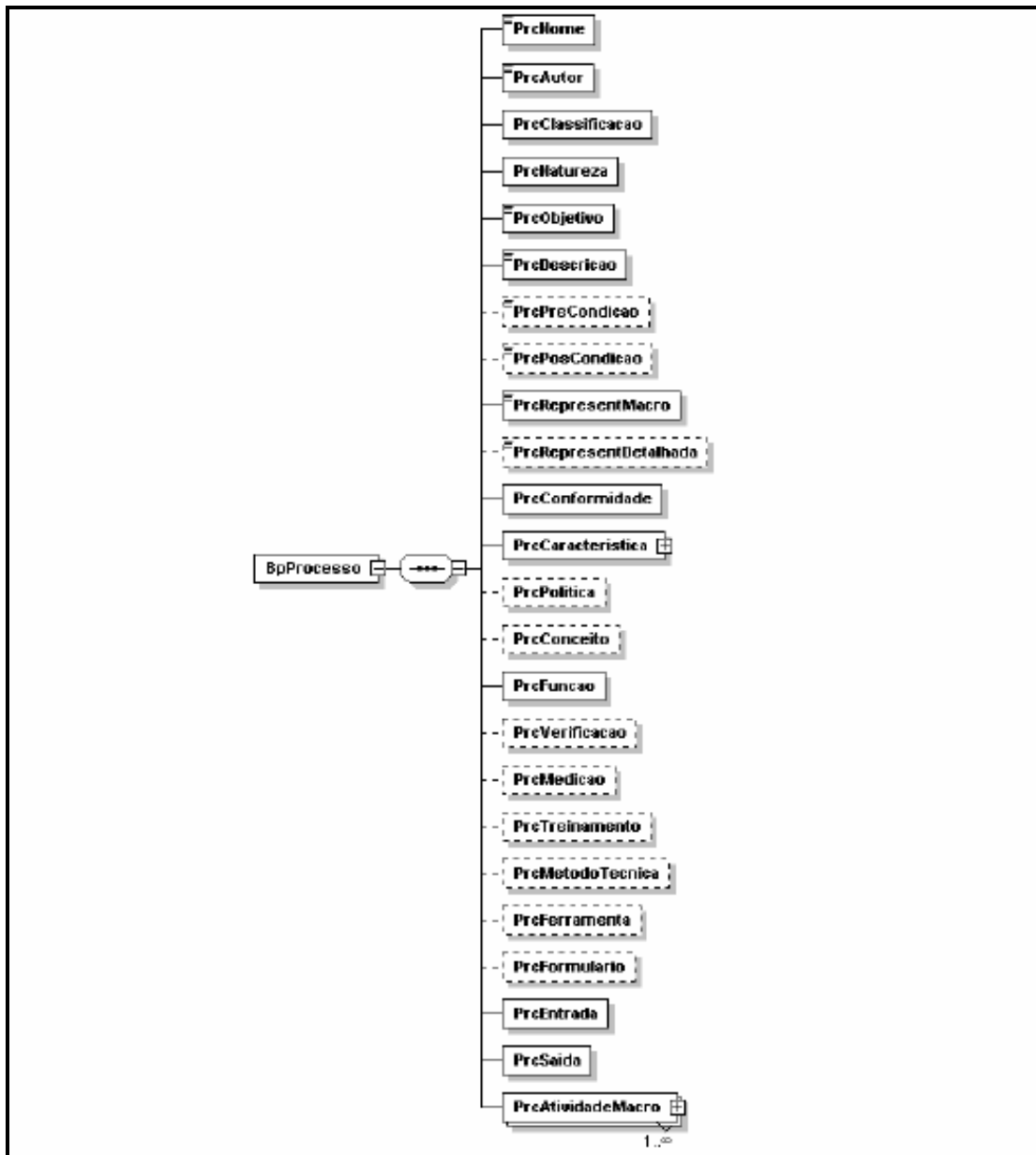


Figura 4 - Elementos do primeiro nível hierárquico do *Usual Process* (BpProcesso) - Processo [Fiorini 01].



Figura 5 - Elementos do segundo nível hierárquico do *Usual Process* – Atividade Macro [Fiorini 01].

Os elementos das atividades detalhadas são semelhantes aos da atividade macro, com exceção de elemento Atividade Detalhada, por este motivo não apresentamos a figura com seus elementos.

Alguns dos elementos do *usual process* fazem referências aos chamados elementos auxiliares. Estes elementos têm um conjunto próprio de elementos, e são independentes de um processo específico, ou seja, eles podem ser utilizados e/ou relacionados a qualquer processo. Abaixo temos uma lista destes elementos auxiliares com suas descrições [Fiorini 01]:

- Conceito: Lista termos específicos do Udl.
- Função: Atores do processo.
- Verificação: Revisões e auditorias realizadas no processo.
- Medição: Conjunto de definições, métodos e atividades utilizadas para medir um processo.
- Treinamento: Instruções especializadas para aprimorar o conhecimento dos envolvidos no processo.
- MétodoTécnica : Conjunto razoavelmente completo de regras e critérios, que estabelecem uma maneira precisa e repetível de executar uma tarefa e chegar a um resultado desejado.

- Ferramenta: Instrumento de apoio ao processo.
- Formulário: Estrutura de um documento em branco. São conhecidos como os *templates*.
- Política: São as diretrizes macro da organização referentes aos processos e seu uso.
- Artefato: Insumos que são fornecidos ao processo ou produtos que são produzidos pelo processo.

Abaixo exibiremos um resumo dos itens deste meta-modelo com suas descrições: [Fiorini 01]

Elemento Processo

- Nome: Identificação do processo.
- Autor: Criador do processo.
- Classificação: Podem ser processos fundamentais, de apoio ou organizacionais.
- Natureza: Propósito do processo dentro do banco de dados. Por exemplo, se é um processo criado para um projeto ou se é um *framework*.
- Objetivo: Ao que se destina o processo, sua finalidade.
- Descrição: Uma descrição genérica sobre o que o processo faz e como, um sumário sobre o processo.
- Pré-condição: Pré-requisitos para o processo iniciar.
- Pós-condição: O que deve estar realizado após a conclusão do processo.
- Representação Macro: Um diagrama ilustrando o processo do ponto de vista das suas atividades macro. A representação possui um atributo onde também se pode descrever como é a ordem; relacionamento entre as atividades.
- Representação Detalhada: Um diagrama ilustrando as atividades detalhadas do processo.
- Conformidade: Indicação se o processo está em conformidade com alguma norma ou modelo, como por exemplo, CMM ou ISO/IEC 9000.

- **Característica:** Refere-se aos atributos do processo, seu uso anterior e conhecimento necessário para melhor utilizá-lo. Possui os elementos “filho área”, “ciclo de vida”, “tipo de sistema”, “tamanho da organização”, “duração do projeto”, “tamanho da equipe”, “conhecimento requerido” e “localização da equipe usuária”.
- **Política/Conceito/Função/Verificação/Medição/Treinamento/Método Técnica/Ferramenta/Formulário/Entrada/Saída:** São referências aos elementos auxiliares dos processos.
- **Atividade Macro:** Conjunto de tarefas ou trabalho a ser realizado. Fornece uma visão de alto nível do que é realizado. O detalhamento das tarefas que compõem a atividade macro está descrito na atividade detalhada.

Elemento Atividade Macro

- **Nome:** Identificação da atividade. Seu nome está decomposto para que seja inserido na forma de facetas – ação e objeto. Sinônimos para a ação também são inseridos.
- **Descrição:** Texto sobre o que a atividade faz e como é realizada.
- **Pré-condição:** Pré-requisito para o início da atividade macro.
- **Entrada/Saída:** São referências para o elemento artefato.
- **Predecessor:** Atividade(s) que precede(m) a atividade em questão. Referência para as macro atividades do processo.
- **Pós-condição:** O que deve estar realizado após a conclusão da atividade macro.
- **Restrição:** Idem ao *usual process*.
- **RspNome:** Responsável pela execução da atividade. Referência para o elemento função.
- **MétodoTécnica:** Método/técnica utilizada pela atividade. Referência para o elemento método/técnica.
- **Ferramenta:** Aplicativo utilizado pela atividade ou que a implementa. Referência para o elemento ferramenta.
- **AtividadeDetalhada:** Procedimento detalhado da atividade macro.

Conforme anteriormente explicado, o elemento “atividade detalhada” possui os mesmos elementos da atividade macro, com exceção da “atividade detalhada”, por isso optou-se por não descrevê-los.

Analisando os meta-modelos apresentados, percebemos que o primeiro possui informações que permitem apresentar um modelo de processo que exprime as atividades que ocorrem no dia-a-dia das organizações, bem como suas entradas e saídas, ou seja, mostram uma boa abstração da realidade. Porém, segundo o próprio autor [Georgakopoulos 95], este meta-modelo não permite capturar os objetivos do processo, assim como as necessidades da organização.

O segundo meta-modelo, por sua vez, também apresenta facilidade na apresentação de uma abstração das atividades que são executadas numa organização e, além disso, conseguem capturar os objetivos e necessidades da empresa, tem como grande vantagem o fato de ter sido desenvolvido numa arquitetura que permite a reutilização de processos. Por este diferencial, optamos por selecioná-lo para ser utilizado neste trabalho.

2.4. Formas de Representação

As formas de representação do modelo de processo do negócio são a linguagens e/ou notações que são utilizadas na especificação do processo. Listamos algumas delas abaixo:

- **Linguagem Usual Process [Fiorini 01]**

A linguagem proposta pela autora é uma linguagem textual baseada no meta-modelo proposto pela mesma. Portanto, os elementos do meta-modelo com suas descrições, apresentados na seção anterior, é a própria linguagem que será utilizada na especificação do processo.

- **UML**

Linguagem que foi desenvolvida pela OMG (*Object Management Group*). É amplamente conhecida no mercado e conta com as vantagens de permitir uma uniformização de linguagem entre o engenheiro de sistema - que utiliza a UML pra modelar seu sistema - e o engenheiro de processo - que estará utilizando a mesma para

modelar o processo da organização. É uma linguagem visual, ajuda no foco ao cliente (organização), e ajuda a derivar melhor os requisitos do sistema. [Baker 01]

Esta linguagem possui um conjunto de diagramas para serem utilizados na representação dos modelos. [Cesare 02] realizou uma pesquisa visando demonstrar o potencial dos diagramas da UML para modelar processo de negócio. Para isto, relacionou cada um dos diagramas da UML com os principais conceitos que devem estar mapeados nos modelos de negócio. E foram marcadas com um ponto (●) as relações na qual o diagrama da UML demonstra potencial para representar o respectivo conceito no modelo de negócio. O resultado, com todos os relacionamentos, é demonstrado na Figura 6. Nesta figura também é mostrada uma breve descrição de cada um dos modelos da UML, onde pode ser encontrado o que cada um deles se propõe a apresentar.

Section	Diagram	Description	Business Modelling Potential					
			R O L E S	A C T O R S	A C T I V I T Y	I N T E R A C T I O N S	P R O C E S S E S	E N T I T I E S
4.1	Use case	Models the functionality of the system as it is visible to external actors that stimulate this functionality.	●	●			●	
4.3	Class	Models the static part of a system in terms of classes, relationships amongst classes and their packaging.	●					●
	Object	Models the representation of instances of class models.	●					●
4.2	Collaboration	Models object interaction via message passing placing emphasis on the roles in the interaction and their links to each other.		●		●	●	●
	Sequence	Models object interaction via message passing placing emphasis on the sequence of interactions.		●		●	●	●
4.4	Statechart	Models the states and state transitions of an object of a given class.			●			●
	Activity	Models the flow of activities defined for any given procedure.			●		●	
	Component	Models the dependencies amongst software components.	Not considered					
	Deployment	Models the configuration of run-time processing elements and the software components, processes, and objects that live in them.	Not considered					

Figura 6 - Potencial dos diagramas da UML para modelos de processo - [Cesare 02].

- **BPMN [White 04]**

O BPMN (*Business Process Modeling Notation*) foi desenvolvido pela BPMI (*Business Process Management Initiative*), e tem como principal objetivo produzir uma linguagem de especificação de modelo de processo de negócio que seja facilmente compreendida por todos, desde os usuários do modelo, até os analistas e modeladores do processo.

O BPMN define um Diagrama de Processo de Negócio (BPD) através de mapeamento dos fluxos dos processos, que exibem uma rede de objetos que representam as atividades e o fluxo de controle das atividades para mostrar a ordem de performance das atividades.

O BPD possui um conjunto de elementos gráficos que facilitam o desenvolvimento do diagrama. Estes elementos subdividem-se em 4 categorias:

- **Objetos de Fluxo**

São os elementos:

- *Eventos*, que são representados graficamente por um círculo e, no diagrama, simbolizam o acontecimento de evento durante o curso do processo de negócio. Estes eventos afetam o fluxo do processo e têm causa e impacto.

- *Atividades*, que são representados graficamente por um retângulo com bordas arredondadas e, no diagrama, simbolizam a performance de trabalho das organizações. Existem duas modalidades de atividades: as tarefas e os sub-processos. As tarefas são atividades atômicas, enquanto os sub-processos são uma composição de tarefas para execução de uma atividade.

- *“Portas” ou Gateway*, que são representadas graficamente por um losango e, no diagrama, são utilizados para controlar a divergência e convergência do fluxo de seqüência das atividades, ou seja o fluxo de decisões.

- **Objetos Conectores**

São os elementos:

- *Fluxo de Seqüência*, que são representados graficamente por uma seta com linha sólida e seta pontiaguda, no diagrama são utilizados para mostrar a ordem das atividades no processo.

- *Fluxo de Mensagem*, que são representados graficamente por uma seta com linha tracejada e seta pontiaguda vazada (sem preenchimento), no diagrama são utilizados para mostrar o fluxo de mensagens trocadas entre dois participantes do processo.

- *Associação*, que são representados graficamente por uma seta com linha pontilhada e seta pontiaguda aberta, no diagrama são utilizadas para mostrar a associação de dados, textos ou artefatos com os objetos do fluxo, por exemplo, para mostrar as entradas e saídas das atividades.

- Raias

São os elementos:

- *“Blocos” ou Pool*, que são representados graficamente por retângulo, no diagrama são utilizados para mostrar as atividades de um participante do processo.

- *“Divisões” ou Lane*, que são representados graficamente por um retângulo com divisões, no diagrama são utilizados para particionar o elemento “bloco” e, desta forma, organizar e categorizar as atividades exibidas no “bloco”.

- Artefatos

São os elementos:

- *Objetos de dados*, que são representados graficamente por uma “folha”, no diagrama são utilizados para mostrar os dados que são requeridos e produzidos pelas atividades.

- *Grupo*, que é representado graficamente por um retângulo com bordas pontilhadas e arredondadas, no diagrama é utilizado para documentação de análises propostas, mas não afeta o fluxo do processo.

- *Anotação*, que é representada graficamente por um traço com o retângulo aberto, no diagrama é utilizada para prover informação adicional ao leitor do diagrama, ou seja, uma observação.

Abaixo apresentamos uma figura com todos os elementos citados acima e sua respectiva representação gráfica:

Conjunto	Elementos	Representação Gráfica
Objetos de dados	Eventos	
	Atividades	
	Gateway	
Objetos Conectores	Fluxo de Sequência	
	Fluxo de Mensagem	
	Associação	
Raias	Pool	
	Lane	
Artefatos	Objeto de dados	
	Grupo	
	Anotação	

Figura 7 - Elementos o BPMN [retirado de White 04].

Realizando uma análise nas linguagens, podemos observar que a primeira linguagem, apesar de ser textual e mesmo não possuindo uma notação própria para representá-la graficamente, tem como grande vantagem o fato de ter sido desenvolvida para suportar o meta-modelo do *usual process*. Este conta com um diferencial em relação aos outros por possuir a característica da reutilização.

A segunda linguagem mostrada tem modelos com potenciais diferenciados para modelagem de processo, ou seja, cada modelo atende a alguns dos principais conceitos, como mostrado na Figura 6. Portanto, é necessário construir um conjunto de modelos com objetivos diferentes para conseguir expressar um modelo de processos que atenda aos seus principais conceitos.

A terceira linguagem tem como principal trunfo a facilidade de compreensão de suas notações nas representações do processo, contudo, não está completamente adaptada ao meta-modelo selecionado.

Pelo fato de termos selecionado, para fins de utilização neste trabalho, o meta-modelo *usual process*, por ser baseado numa arquitetura que permite a reutilização de processo, optamos também por selecionar a própria linguagem do *usual process* para representação do modelo de processos, vista sua completa adaptabilidade a este meta-modelo.

2.5. Ferramentas

As ferramentas apóiam o levantamento e a modelagem de processos e, segundo [Bastos 00], podem ser classificadas em três tipos:

- **Ferramentas de desenho gráfico, sem referencial metodológico e não baseadas em banco de dados**

São ferramentas que disponibilizam ao usuário um meio de criar um diagrama a partir de objetos pré-formatados ou disponibilizados nesta. Este tipo de ferramenta não tem método de modelagem associado, deixando o usuário livre para criar qualquer tipo de objeto ou modelo. Exemplos deste tipo de ferramenta: *MS Power Point, Corel Draw*.

- **Ferramentas com referências metodológicas, não baseadas em banco de dados**

São ferramentas que disponibilizam ao usuário objetos para o uso dentro da metodologia, ou seja, mesmo tendo os objetos e modelos compatíveis com metodologias, a ferramenta não força o processo de modelagem. Porém, utilizando os objetos dentro da metodologia, a ferramenta exige maior padronização do usuário em respeitar as regras do modelo; além do fato que, ao utilizar uma metodologia, temos uma maior garantia para obtenção dos resultados. Exemplos deste tipo de ferramenta: *FlowChater, Visio*.

- **Ferramentas com referências metodológicas, baseadas em banco de dados**

Estas ferramentas possuem banco de dados associados, que permitem que os objetos e informações modeladas sejam armazenados de forma organizada, com consistência e unicidade. Por possuírem metodologias associadas, também garantem uma melhor obtenção de resultados.

Exemplos deste tipo de ferramenta: *Aris*, *System Architect*.

O autor [Bastos 00] realizou uma pesquisa sobre as ferramentas atuais de modelagem de processos. Nesta pesquisa, mostrou as características de algumas das ferramentas atuais. Abaixo iremos apresentar algumas de suas conclusões sobre as ferramentas ARIS e Visio:

- **ARIS**

A *Aris Toolset* foi desenvolvida pela empresa alemã IDS Prof. Scheer GmbH (www.ids-scheer.de) e utiliza a metodologia *Business Process Improvement*, porém também pode utilizar outras metodologia e modelos, como UML(*Unified Modeling Language*) e BPM (*Business Process Modeling*).

O autor [Bastos 00] destaca seus principais pontos fortes:

- “A visualização/navegação entre os modelos (permite interrelacionamentos entre objetos e modelos facilitando a navegação em uma interface amigável)”.
- “A flexibilidade associada às possibilidades de organização de modelos e à vasta quantidade de modelos e metodologias disponíveis”.
- “Permite ao usuário a elaboração de análises e geração de relatórios a partir dos modelos”.

▪ **VISIO**

O Visio foi desenvolvida pela americana *Visio Corporation* (www.visio.com). É uma ferramenta de diagramação e desenho com figuras “*templates*” pré-disponíveis ao usuário. Possui três versões (*Professional*, *Technical* e *Standard*) que se diferenciam pela presença/ausência de alguns diagramas e símbolos.

É simples e de fácil utilização. Não possui banco de dados, porém tem modelos para diferentes metodologias, com objetivos diversos.

O autor [Bastos 00] destaca seus principais pontos fortes:

- “Interface amigável e de fácil utilização”.
- “*Templates* com objetos para diversos tipos de diagramas: fluxogramas, organogramas, modelos IDEF (metodologia de diagramação de sistemas), UML, TQM(*Total Quality Management*) e ISO(*Organization for International Standarlization*), diagramas de redes (IBM, HP, 3COM), etc.”

▪ **SRP**

O SRP – Sistema de Reutilização de Processos é uma ferramenta de cadastro e reutilização de processos de negócio desenvolvida para este trabalho. Seus requisitos para desenvolvimento foram baseados na reengenharia do Sistema de Reutilização de Processos de *Software* – RPS, desenvolvido pela [Fiorini 01] e atende às características do meta-modelo desenvolvido pela autora, o *usual process*.

A ferramenta SRP tem uma diferença de abordagem em relação ao RPS, pois o primeiro é mais genérico, ou seja, ele tem como objetivo atender aos processos de negócio em geral, enquanto o segundo visa apenas os processos de *software*.

O SRP possui banco de dados associado e não tem metodologia para desenvolvimento de modelo de processo associada, porém, pode ser adaptável à qualquer metodologia, desde que esta metodologia produza modelos que sejam compatíveis com o meta-modelo no qual esta ferramenta se baseia.

Esta ferramenta, conforme a linguagem do meta-modelo, é textual e não possui uma notação própria, mas é flexível para aceitar qualquer notação criada pelo usuário, desde que siga às características do meta-modelo, ou seja, temos na ferramenta um local onde é inserido, para cada processo, sua representação gráfica, que pode ser desenvolvida em qualquer notação pelo usuário, desde que esta notação esteja em conformidade com o meta-modelo no qual a ferramenta se baseia.

Esta ferramenta também possibilita a reutilização, total ou parcial, dos processos armazenados no banco de dados, na criação de novos processos de negócio. Possibilitando, desta forma, que empresas de ramos de negócio semelhante, quando modelarem seus processos, possam utilizar, como base, outros processos semelhantes ao da sua organização, ou mesmo adaptar partes de processos que já são utilizados, na prática, por outras empresas.

O escopo desta ferramenta abrange basicamente duas funcionalidades: o cadastro de processos e a reutilização dos mesmos.

Concluimos que uma análise de ferramentas de apoio à modelagem de processos deve considerar as particularidades de cada ferramenta e, principalmente, o contexto onde se pretende utilizá-lo [Bastos 00]. Logo, levando em consideração o contexto, a ferramenta mais adequada para ser utilizada é o SRP, visto ser uma ferramenta adaptada ao meta-modelo e à linguagem selecionados anteriormente, e que possibilita a reutilização de processos. Porém, apesar desta ser considerada a escolha mais adequada no contexto, atentamos para o fato de que a falta de uma notação própria é um fator que poderá prejudicar a uniformidade de entendimento do modelo. Quando não existe uma notação própria deixamos os usuários livres para criar suas notações e, portanto, teremos vários símbolos de notação significando o mesmo item do meta-modelo, o que pode tornar confusa a leitura de um modelo para o usuário. Em vista disso, proporemos no capítulo 4 uma notação para este meta-modelo, e selecionamos a ferramenta *Visio* para prover os objetos que utilizaremos como símbolos na notação. Esta escolha se baseou no fato do *Visio* ser uma

ferramenta de diagramação e desenho, que é exatamente o que necessitamos para complementar a linguagem do *usual process*.

2.6. Conclusão

Neste capítulo, vimos pontos importantes que auxiliam no entendimento sobre modelagem de processo de negócio, como seus objetivos, métodos para construção de modelos de processo de negócio, meta-modelos - que guiam a construção do modelo -, ferramentas e linguagens - que auxiliam no desenvolvimento do modelo.

Podemos notar que a modelagem de processo para organização é um instrumento poderoso, na medida em que, provendo um melhor entendimento de como a organização funciona, torna-se mais eficiente o controle, coordenação e a tomada de decisões sobre a operação e organização da empresa.

Os benefícios que esta modelagem pode trazer para a organização são desde um material rico para análise de estruturação das atividades, até uma documentação do negócio para fins de auditoria da organização, além de permitir uma uniformização de entendimento do negócio entre todos envolvidos na organização.

No próximo capítulo apresentaremos uma pesquisa em torno da ontologia, que tem por objetivo modelar o conhecimento do Udl da organização. Lembramos que nosso objetivo é unir modelos de processo e ontologia de maneira a facilitar a tarefa de elicitação de requisitos para sistemas de informação.

3. Ontologias

Este capítulo tem a finalidade de apresentar um estudo sobre ontologia. Neste estudo, apresentaremos a definição de ontologia e seus objetivos, bem como alguns dos métodos existentes para construção de ontologias, seus meta-modelos, linguagens e ferramentas utilizadas para apoiar a construção das ontologias.

3.1. Definição e Objetivos

A ontologia é definida por diversos autores na literatura, abaixo apresentamos algumas destas definições:

Para [Noy 01], ontologia é definida como:

“a formal explicit description of concepts in a domain of discourse (classes (sometimes called concepts)), properties of each concept describing various features and attributes of the concept (slots (sometimes called roles or properties)), and restrictions on slots (facets (sometimes called role restrictions)). An ontology together with a set of individual instances of classes constitutes a knowledge base. In reality, there is a fine line where the ontology ends and the knowledge base begins.”

Em [Oliveira 99], encontramos o seguinte conjunto de definições para ontologias:

“(a) Ontologia é uma especificação explícita de uma conceituação [Gruber 93,95].

(b) Ontologia é uma descrição parcial e explícita de uma conceituação [Guarino 95].

(c) *Ontologia é uma teoria sobre um domínio que especifica um vocabulário de entidades, classes, propriedades, predicados e funções; e um conjunto de relações que, necessariamente, amarram esses vocabulários [Farquhar 99].*”

Basicamente, podemos notar que ontologias são especificações formais de Universos de Informações, que tem, segundo [Macedo 03], como características importantes:

- Expressar o consenso do conhecimento de uma comunidade de pessoas em um Udl;
- servir como uma referência de termos definidos;
- fornecer uma linguagem suficientemente expressiva para uma comunicação eficaz entre as pessoas, ou seja, onde o entendimento seja único.

A ontologia, para área de informática, tem como objetivo facilitar o compartilhamento e reutilização das informações [Breitman 04], além de definir uma especificação conceitual para “os conhecimentos” de um determinado Udl.

Para [Noy 01], o resumo dos objetivos da ontologia são expressos nos seguintes itens:

- Compartilhar o entendimento comum da informação entre as pessoas e os agentes de *software*.
- Disponibilizar a reutilização do domínio da informação.
- Deixar a compreensão do Udl explícita.
- Separar o conhecimento do Udl do conhecimento operacional.
- Analisar o conhecimento do Udl.

Como vimos nas definições apresentadas, a ontologia é uma especificação formal de conceitos e termos do universo de informações, e objetiva utilizar esta especificação para compartilhar um entendimento único do Udl.

Para auxiliar na criação desta especificação, existem métodos que norteiam o desenvolvimento da ontologia, meta-modelos que auxiliam como guia das informações que devem estar contidas na ontologia, ferramentas para apoiar o desenvolvimento da ontologia e linguagens para auxiliar na representação da

mesma. Nas seções seguintes, apresentaremos uma pesquisa sobre cada um dos itens citados acima como facilitadores na construção de uma ontologia.

Antes de apresentar os itens facilitadores do desenvolvimento de ontologias, enfatizaremos alguns pontos que devem ser considerados na criação da ontologia [Noy 01]:

- Não existe uma maneira correta para modelar um Udl, sempre existirão várias alternativas. A melhor solução depende da aplicação da modelagem.
- Desenvolvimento de ontologia é um processo iterativo.
- Os conceitos, numa ontologia, têm que ser os mais semelhantes possíveis com os objetos e relacionamentos do Udl modelado.

3.2. Métodos para Construção de Ontologia

Os métodos para construção da ontologia nos mostram o passo-a-passo que deve ser executado para desenvolver uma ontologia de Udl. Abaixo apresentamos alguns deles:

- **A Simple Knowledge-Engineering Methodology [Noy 01]**

A autora propõe o seguinte método para construção de uma ontologia de Udl:

A. Determinar o Udl e o Escopo da Ontologia

Para auxiliar nesta tarefa a autora propõe que sejam respondidas as seguintes questões:

- O que o Udl que ontologia irá cobrir?
- Para que iremos usar a ontologia?
- Para que tipo de questões as informações da ontologia deveram prover respostas?
- Quem irá usar e manter a ontologia?

B. Considerar a Reutilização de Ontologias Existentes

Ao iniciar o desenvolvimento da ontologia, deve-se considerar a possibilidade de alguém já ter desenvolvido alguma ontologia que possamos utilizar de fonte para refinar e estender seus conceitos no desenvolvimento da ontologia de Udl.

C. Enumerar Termos Importantes da Ontologia

Deve-se montar uma lista com termos que são usualmente utilizados em sentenças dos usuários do Udl. A partir destes termos, devemos formular perguntas sobre os mesmos, que auxiliem na derivação de novos termos. Por exemplo: quais são as propriedades dos termos? O que estes termos expressam no Udl?

Estes termos servirão de base para a definição das classes da ontologia do Udl.

D. Definir as Classes e a Hierarquia das Mesmas

Para a autora, existem três abordagens para definir a hierarquia de classes:

- **Top-down:** São definidos os termos do Udl, que são mais genéricos, e a partir destes termos são realizadas especializações. Por exemplo: um termo bem genérico do Udl de vinhos é o próprio vinho, que pode ser especializado em vinho branco, tinto e rose.
- **Bottom-up:** São definidos os termos mais específicos do Udl e, em seguida, são realizados agrupamentos destes termos a partir de características comuns, obtendo-se, desta forma, uma generalização dos mesmos.
- **Combination:** É uma combinação das duas abordagens acima, onde são definidos termos considerados mais destacados dentro do Udl e, a partir dos mesmos, são realizadas especializações ou generalizações, conforme for mais apropriado.

E. Definir as Propriedades das Classes – “Slots”

Deve-se, para cada classe, descrever sua estrutura interna de conceitos, ou seja, suas propriedades. Por exemplo: cor e sabor são propriedades da classe vinho.

As propriedades das classes são classificadas como:

- Propriedades Intrínsecas: São propriedades da estrutura interna da classe. Por exemplo: sabor do vinho.
- Propriedades Extrínsecas: São propriedades da estrutura externa da classe. Por exemplo: nome do vinho.
- Propriedades Partes De: São propriedades relacionadas à organização da classe. Por exemplo: no Udl de automóveis, o volante é parte do carro.
- Relacionamentos com outras classes: São propriedades que exibem os relacionamentos com outras classes. Por exemplo: fabricante de vinhos tem uma relação entre o vinho e a vinicultura.

F. Definir as Facetas das Propriedades das Classes

As propriedades das classes possuem facetas, ou seja, particularidades que definem, de forma mais específica, as propriedades, ou “*slots*”. Como exemplos de facetas temos a definição do tipo de valor da propriedade, valores permitidos para propriedade, número de valores permitidos (cardinalidade). Abaixo listaremos facetas mais comuns citadas pela autora:

- Cardinalidade

A cardinalidade permite definir a quantidade de valores que o “*slot*” pode possuir. Podendo ser expressa por valores definidos, ou através dos termos cardinalidade singular ou cardinalidade múltipla, onde não especificamos exatamente a quantidade de valores do “*slot*”, apenas informamos que pode ter apenas 1 valor, ou múltiplos valores.

- Tipo de valor

É o tipo de valor que o “*slot*” pode ter. Abaixo alguns destes valores:

- *String*: Utilizado nos “*slots*” do tipo texto. Exemplo: “*slot*” nome (classe Vinho).
- *Number*: Utilizado nos “*slots*” do tipo numérico. Exemplo: “*slot*” preço (classe Vinho).
- *Boolean*: Utilizado nos “*slots*” do tipo “*flag*”, onde o tipo de valor deve ser “verdadeiro” ou “falso”. Exemplo: “*slot*” gasoso (classe Vinho).
- *Enumerated*: Utilizado nos “*slots*” que permitem ter uma lista de valores. Exemplo: “*slot*” Sabor (classe Vinho), que tem como lista de valores, forte, moderado e delicado.
- *Instance*: Utilizado nos “*slots*” que representam relacionamento com outras classes. Exemplo: “*slot*” produzido (classe Vinho) representando o relacionamento com a classe “Produtor de Vinho”.

G. Criar Instâncias

Este é o último passo do método, onde são criadas as instâncias das classes definidas nos passos anteriores.

- **Processo de Semi-Automático para Construção de Ontologia [Breitman 04]**

A Figura 8 é um quadro com a descrição passo-a-passo do método, proposta por [Breitman 04] para construção de uma ontologia de Udl. Este método tem como ponto de partida o léxico do Udl [Leite 00], pois é a partir deste artefato que o método é executado, ou seja, o léxico é a entrada deste método.

O léxico, formalmente chamado LAL (Léxico Ampliado da Linguagem), é o conjunto de símbolos usados no contexto do Udl que são descritos na sua forma denotativa e conotativa, ou seja, cada conceito do contexto é representado por um símbolo, que é descrito de forma a retratar dois aspectos: a noção (denotação) e o impacto (conotação). A noção é o

significado do símbolo, enquanto o impacto é a influência do símbolo no contexto [Felicíssimo 04]. Os símbolos do léxico também são classificados em 4 categorias: sujeito, objeto, verbo e estado. [Leite 99]

As descrições dos símbolos seguem o princípio de circularidade, que prega a utilização dos outros símbolos do léxico na descrição dos símbolos e do vocabulário mínimo, que diz que as partes das descrições, que não são outros símbolos do léxico, devem pertencer a um subconjunto reduzido de palavras com significado bem definido. [Leite 99]

1. *Listar os termos alfabeticamente de acordo com seu tipo (verbo, objeto, sujeito ou estado)*
2. *Fazer 3 listas: conceito (classe) (C), propriedade (R) e axiomas (AO). Na lista de classes cada entrada terá um nome, descrição (linguagem natural) e uma lista contendo zero ou mais rel (função que relaciona o conceito em questão a outros, de maneira não taxonômica). As entradas na lista de axiomas terão nomes (labels) somente.*
3. *Utilizando a lista de símbolos do léxico classificados como sujeito ou objeto, para cada termo:*
 - 3.1 *Adicione uma nova classe a lista de classes. O nome da classe é o símbolo do léxico propriamente dito. A descrição da classe é a noção do termo.*
 - 3.1.1 *Para cada impacto,*
 - 3.1.1.1 *Checar se já faz parte da lista de propriedades da ontologia.*
 - 3.1.1.2 *Caso não faça parte da lista (a propriedade ainda não existe), adicione uma nova propriedade na lista (de propriedades). O nome da propriedade deve ser baseado no verbo utilizado para descrever o impacto*
 - 3.1.1.2.1 *Verificar consistência.*
 - 3.1.1.3 *Na lista de classes adicione uma nova rel para a classe em questão. A rel é formada pela classe + a propriedade (definida em 3.1.1.1) + a classe relacionada (esta classe é o objeto direto/indireto do verbo utilizado no impacto do símbolo do léxico. Usualmente é um termo do próprio léxico e aparece sublinhado).*
 - 3.1.1.4 *Checar se existem indicativos de negação no vocabulário mínimo que relacionem duas ou mais classes. Verificar se estas classes possuem um relacionamento do tipo disjuntas (exemplo macho e fêmea).*
 - 3.1.1.4.1 *Se verdadeiro, adicionar o disjoint a lista de axiomas.*
 - 3.2 *Verificar consistência.*
 4. *Utilizando a lista de símbolos classificados como tipo verbo, para cada termo:*
 - 4.1. *Checar se já faz parte da lista de propriedades da ontologia.*
 - 4.1.1. *Caso não faça parte da lista (a propriedade não existe), adicione uma nova propriedade na lista (de propriedades). O nome da propriedade é o símbolo do léxico propriamente dito.*
 - 4.1.1.1. *Verificar consistência.*
 5. *Utilizando a lista de símbolos classificados como tipo estado, para cada termo:*
 - 5.1. *Para cada impacto*
 - 5.1.1 *Tentar identificar a importância relativa do termo para a ontologia. Esta estratégia é similar a utilização de questões de competência proposta em [Gruninger95]. Estas questões são obtidas através do rephraseamento dos impactos de cada símbolo em perguntas iniciadas por quando, onde, o quê, quem, porque, e como.*
 - 5.1.2 *Checar se existem indicativos de negação no vocabulário mínimo que relacionem duas ou mais classes. Verificar se estas classes possuem um relacionamento do tipo disjuntivo (exemplo macho e fêmea)*
 - 5.1.2.1 *Se verdadeiro, adicionar o disjoint a lista de axiomas.*
 - 5.1.3 *Caso o termo seja central a ontologia, classifique-o como classe (C)).*
 - 5.1.4 *Caso contrário (o termo não é central para a ontologia) classifique-o como propriedade (R).*
 - 5.1.5 *Verificar consistência.*
 6. *Quando todos os termos tiverem sido adicionados à ontologia,*
 - 6.1 *Checar se existe conjuntos de conceitos que compartilham rel idênticos*
 - 6.1.1 *Para cada conjunto de conceito que compartilha rel, construir uma lista de conceitos separados*
 - 6.1.2 *Buscar na ontologia conceitos que fazem referência a todos os membros desta lista*
 - 6.1.2.1 *Se não forem encontrados, busca na noção e no impacto de cada membro da lista de conceitos tentando identificar um termo comum do vocabulário mínimo*
 - 6.1.3 *Construir uma hierarquia de conceitos onde todos os membros da lista de conceitos é um sub-conceito do conceito encontrado em*
 - 6.1.4 *Verificar consistência*

Figura 8 - Processo Semi-automático para construção de ontologia. [Breitman

Existem, na literatura, outros métodos utilizados para construção de ontologias. Abaixo apresentamos um quadro que mostra algumas de suas principais características segundo [Perez 04]:

Características	Cyc	Uschold & King	Gruninger & Fox	Kactus	Methontology	Sensus	On-To-Knowledge
Ciclo de Vida	Desenvolvimento de Prototipos	Não proposta	Desenvolvimento de Prototipo ou	Desenvolvimento de Prototipos	Desenvolvimento de Prototipos	Não proposta	Incremental e ciclica com desenvolvimento
Estratégia de identificação de conceitos	Não especificada	Middle-out	Middle-out	Top-down	Middle-out	Não especificada	Top-Down Botton-up Middle-out
Ferramentas de apoio	Cyc Tools	Não especificada	Não especificada	Não especificada	ODE WedODE OntoEdit Protege-2000	Não especificada	OntoEdit com o plugins
Aceitação por organizações externas	Não conhecida	Não conhecida	Não conhecida	Não conhecida	FIPA (www.fipa.org)	Não conhecida	VU Amsterdam, CogniT, Administrator, SwissLife
Ontologias criadas	Cyc Tools	Enterprise Ontology	TOVE	Electrical network ontologies	Chemicals OntoRoadMap Esperanto ontologies	Militar air campaing	Não citada

Tabela 1 - Características dos métodos para construção de ontologia [adaptado de Perez 04].

Analisando os métodos, podemos notar no primeiro método muita subjetividade nas descrições de suas etapas. Encontramos, na descrição, muito mais a explicitação do meta-modelo utilizado (no caso o *frame ontology*), do que o passo-a-passo que deve ser executado para construção da ontologia. Já o segundo método é bem mais específico na descrição do seu passo-a-passo, possibilitando uma execução bem mais objetiva do mesmo. E também conta com a vantagem de ser baseada num processo maduro para a construção de léxicos (entrada do método), que já foi validado em projetos reais [Breitman 05].

3.3. Meta-modelos de Ontologia

O meta-modelo é necessário, na construção da ontologia, porque auxilia a especificação das informações que devem estar presentes no modelo de ontologia. Seu objetivo é unificar a semântica utilizada na representação da ontologia [Perez 04]. Abaixo apresentamos alguns destes meta-modelos:

- **Frame Ontology [Gruber 93]:**

Este meta-modelo é constituído de uma coleção de convenções comuns para organização do conhecimento. Na descrição do método “*A Simple Knowledge-Engineering Methodology*” [Noy 01] já foram descritos os itens que compõem este meta-modelo. Abaixo apresentamos um resumo dos principais componentes deste meta-modelo segundo [Perez 04]:

- Classes: Uma coleção de indivíduos.
- Indivíduos ou Instâncias: Uma instância de uma classe.
- “*Slots*” ou Propriedades: São as características das classes
- Facetas: É a especificação dos “*slots*”. Exemplo: Tipo da classe (*String*, Numérico).

As principais taxonomias utilizadas para organizar as classes e instâncias são:

- Subclasse - of: É o relacionamento entre uma classe “filho” que herda as características da classe “pai”.
- Superclasse - of: É o relacionamento entre uma classe “pai” que provê características para a classe “filho”.
- *Disjoint*: É o relacionamento entre classes que são completamente opostas, ou seja, não possuem nada em comum.
- *Exhaustive*: São classes que não podem ser instanciadas.
- *Partition*: É o relacionamento “Parte de”, onde uma classe é parte de outra classe.
- *Instance- of*: É a instanciação de uma classe.

- **OIL Ontology**

Este meta-modelo tem como principais componentes [Perez 04]:

- **Classes**, que são agrupadas em 6 categorias, descritas abaixo. Este agrupamento de classes tem a finalidade de auxiliar e oferecer mais semântica na definição das classes, visto que as novas classes são definidas por uma, ou um conjunto, destas classes primitivas [Bechhofer01]:

- Classes para definir tipos concretos de expressões: São subclasses da classe *oil: ConcreteTypeExpression* . Estas classes definem expressões numéricas para os números, como, por exemplo: igual, maior que.
- Classes para definir expressões de classes: São subclasses da classe *oil: ClassExpression*. Estas classes podem ser formadas por expressões booleanas, restrições de propriedades e expressões enumeradas. As expressões booleanas são as expressões de conjunção, disjunção e negação entre as classes. As restrições de propriedades qualificam numericamente as restrições através das cardinalidades. As expressões enumeradas representam as classes que podem ter diversos valores.
- Classes para definir características matemáticas: São subclasses das classes *oil: TransitiveProperty*, *oil: FunctionalProperty* e *oil: SymmetricProperty*. Estas classes definem expressões matemáticas de transitividade, funcional e simetria.
- Classes para definir axiomas: São as subclasses da classe *oil: axiom*. Estas classes representam a taxonomia de definição de *disjoint* e *exhaustive* para as classes.
- Classes para definir tipos de dados: São subclasses de *oil:string* e *oil:integer*. São classes para definir os tipos de dados *string* e *integer*.
- Classes Pré-definidas: São subclasses de *oil:Top* e *oil:Bottom* . A classe *oil:Top* é a mais genérica das classes que são de um mesmo tipo. A *oil:Bottom* é uma classe vazia, e é do tipo de qualquer classe.

- **Propriedades ou Relacionamentos entre Classes:**

- *subClasseOf* : relaciona uma classe “filha” com a classe “pai”.
- *hasOperand*: relaciona uma expressão Booleana com os operandos (*And*, *Or* e *Not*).
- *individual*: representam instâncias das classes.

- *toClass*: utilizada para representar que o *range* de uma classe é uma outra.
- *toConcreteType*: utilizadas para representar classe concretas.
- *individualFiller*, *integerFiller* e *stringFiller*: utilizadas para representar valores de propriedades nas classes.
- *Number*: expressa cardinalidade.
- *inverseRelationOf*: expressa propriedades inversas entre classes.
- *hasObject*, *hasSubject* e *isCoveredBy*: representa as propriedades entre classe de *disjoint* e *exhaustive*.

- **OWL Ontology**

As informações que constituem o meta-modelo *OWL Ontology*, e proporcionam representação semântica no modelo ontológico, foram agrupadas em três conjuntos, formando três meta-modelos derivados que possuem níveis diferentes para representação semântica [Horridge 04]:

- O *OWL Lite*, que possui um conjunto menor de informações;
- o *OWL DL*, que possui um conjunto intermediário de informações, entre o *OWL Lite* e o *OWL Full*;
- e, finalmente, o *OWL Full*, que possui o conjunto completo de informações deste meta-modelo.

Esta divisão proporciona flexibilidade para selecionar o nível de semântica que melhor se adapte às necessidades do usuário no contexto do Udl que será modelado.

Abaixo apresentamos os principais componentes deste meta-modelo (os componentes apresentados estão presentes nos 3 meta-modelos do *OWL Ontology*) [Perez 04]:

- **Classes**, que são agrupadas em 7 categorias. Esta categorização é utilizada na definição de novas classes, assim como no meta-modelo *OIL Ontology*:

- Classes para definir Classes e Restrições: São subclasses de *owl:Class* e *owl:Restriction*. São utilizadas para definir classes e as restrições das propriedades das classes.
- Classes para definir propriedades: São subclasses de *owl:ObjectProperty*, *owl:DatatypeProperty*, *owl:TransitiveProperty*, *owl:SymmetricProperty*, *owl:FunctionalProperty*, *owl:InverseFunctionalProperty* e *owl:AnnotationProperty*. São utilizadas para definir propriedades que conectam classes a outras classes (*owl: ObjectProperty*), classes a tipos de dados (*owl:DatatypeProperty*). As classes *owl:TransitiveProperty*, *owl:SymmetricProperty* servem para definir características lógicas às propriedades. Já as *owl:FunctionalProperty*, *owl:InverseFunctionalProperty* são utilizadas definir restrições de cardinalidade.
- Classes para definir diferenças entre instâncias: São subclasses de *owl:AllDifferent*. São utilizadas para definir que duas instâncias são diferentes. Pois, em owl, duas instâncias com identificadores diferentes podem se referir a uma mesma instância.
- Classes para definir tipos de dados enumerados: São subclasses de *owl:dataRange*. São utilizadas para criar tipos de dados enumerados, que têm um conjunto de valores pré-definidos.
- Classes Pré-definidas: São subclasses de *owl:Thing* e *owl:Nothing*. Representam a mais e a menos genérica classe, respectivamente.
- Classes para descrever ontologias: São subclasses de *owl:Ontology*. É utilizada como classe raiz da OWL *ontology* contendo todas as suas definições.
- Classes para descrever o versionamento da ontologia: São subclasses de *owl:DeprecatedClass* e *owl:DeprecatedProperty*. São utilizadas com propósito de versionamento de ontologias. São classes ou propriedades que estão depreciadas em relação à versão corrente da ontologia.

- **Propriedades ou Relacionamentos entre Classes:**

Propriedades	Descrição	Meta modelo
owl:intersectionOf	Propriedade que define expressões de classe que representa conjunção.	OWL Lite
owl:unionOf	Propriedade que define expressões de classe que representa disjunção.	OWL DL
owl:complementOf	Propriedade que define expressões de classe que representa negação.	OWL DL
owl:oneOf	Propriedade que define expressões de classe que representa coleção de indivíduos.	OWL DL
owl:onProperty	Propriedade que define expressões de classe para definir restrições de propriedade.	OWL Lite
owl:allValuesFrom	Propriedade que define expressões de classe para definir valores de restrições.	OWL Lite
owl:hasValue	Propriedade que define expressões de classe para definir facetas.	OWL DL
owl:someValuesFrom	Propriedade que define expressões de classe para definir restrições existenciais.	OWL Lite
owl:mincardinality	Propriedade que define expressões de classe para definir restrições numéricas.	OWL Lite
owl:maxcardinality	Propriedade que define expressões de classe para definir restrições numéricas.	OWL Lite
owl:inverseOf	Define o inverso da propriedade.	OWL DL
owl:sameAs	Define equivalência entre recursos.	OWL DL
owl:equivalentClass	Define equivalência entre classes.	OWL DL
owl:equivalentProperty	Define equivalência entre propriedades.	OWL DL

Propriedades	Descrição	Meta modelo
owl:sameIndividualAs	Define equivalência entre instâncias	OWL DL
owl:differentFrom	Define duas instâncias como diferentes.	OWL DL
owl:disjointWith	Representa o <i>disjoint</i> entre classes.	OWL DL
owl:distinctMembers	Utilizado para definir a lista de instâncias que são diferentes de outras.	OWL DL
owl:versionInfo	Propriedade que fornece informações da versão corrente da ontologia.	OWL DL
owl:priorVersion	Propriedade que define qual versão corrente da ontologia teve uma versão anterior.	OWL DL
owl:incompatibleWith	Propriedade que define que a ontologia é incompatível com outra ontologia.	OWL DL
owl:backwardCompatibleWith	Propriedade que define que a ontologia é compatível com outra ontologia.	OWL DL
owl:imports	Propriedade que referencia outras ontologias Owl que foram importadas nesta ontologia.	OWL DL

Tabela 2 – Lista de propriedades da OWL *Ontology* [retirado de Perez 04].

- É importante salientar que as propriedades utilizadas pelo OWL *Lite*, também poderão ser utilizadas pelo OWL DL e OWL *Full*, e as propriedades utilizadas pelo OWL DL são utilizadas tanto pelo mesmo, quanto pelo OWL *Full* [Perez 04].

Analisando os meta-modelos, percebemos que as diferenças entre eles é na composição das informações que a ontologia deve possuir, ou seja, na quantidade de representações semânticas que disponibiliza para o Udl.

A seleção de um meta-modelo deve, segundo [MacGuinness 04], depender do critério de melhor adaptação as suas necessidades, ou seja, selecionar um meta-modelo mais completo, que possibilite uma gama de representações semânticas, não significa estar escolhendo o melhor meta-modelo.

Acreditamos que, para este trabalho, o “*frame ontology*” é o mais adequado por conseguir fornecer, com seu conjunto de informações, um bom conhecimento do Udl com maior facilidade para criação dos modelos. O “*frame ontology*” possui classes e propriedades mais simples que os outros meta-modelos. Isto facilita o desenvolvimento do modelo por torná-lo menos complexo, mesmo acarretando perda de representação semântica. Mesmo com isso, este meta-modelo nos fornece um bom conhecimento de Udl, ou melhor, o conhecimento considerado suficiente para esta primeira versão do método, onde se busca mostrar a capacidade do método em atingir seus objetivos de maneira mais simplificada. Contudo, não descartamos a possibilidade de evolução deste para contemplar meta-modelos que disponibilizem maior representação semântica do Udl.

3.4. Linguagens

A linguagem é utilizada para unificar as formas de representação da ontologia, de forma que o conhecimento da ontologia possa ser interpretado de maneira única [Perez 04]. Abaixo são apresentadas algumas linguagens de ontologias:

- **OIL**

A linguagem OIL (*ontology inference layer*) foi patrocinada através de um consórcio da Comunidade Européia, através do projeto *On-to-Knowledge*. Esta linguagem foi criada pela necessidade de uma linguagem expressiva que permitisse semântica necessária, e formalismo suficiente, de modo a permitir suporte a mecanismos de inferência. Os serviços de inferência oferecidos incluem detecção de inconsistências e a determinação de relacionamentos do tipo sub-classe de. [Breitman 04]

Atualmente, os editores para a linguagem OIL permitem tradução para outras linguagens com o RDF, por exemplo, mesmo que na tradução ocorra perda de expressividade, como é o caso do editor OILed. [Breitman 04]

- **OWL**

Esta linguagem foi criada pelo consórcio W3C, e foi projetada de modo a realizar uma especificação formal do Udl. [Breitman 04]

OWL possui três linguagens, assim como o meta-modelo OWL *ontology*:

- OWL *Lite*

OWL *Lite* suporta a criação de hierarquias de classificação dos termos, bem como algumas restrições mais simples, como cardinalidades. Não possuem axiomas nem estruturas de relacionamentos sofisticadas. [Breitman 04]

- OWL – DL

OWL-DL é mais expressivo que o OWL *Lite*. É baseado em *Description Logics* e, por isso, possibilita classificação automática de hierarquia e checagem de consistência na ontologia em conformidade com o respectivo meta-modelo. [Horridge 04]

- OWL *Full*

OWL *Full* é a mais expressiva das sub-linguagens de OWL. Deve ser utilizada em situações onde a expressividade semântica é mais importante do que garantir a completeza do meta-modelo, pois não é possível verificar consistência da ontologias representadas em OWL *Full*. [Horridge 04]

A maior parte das ferramentas para edição de ontologias já oferece suporte (ou pelo menos tradução) para a linguagem OWL. Este é o caso do OIEd e do Protege-2000. [Breitman 04]

Baseando-se no fato da maioria das ferramentas para edição de ontologias oferecerem tradução para diversas das linguagens de ontologias, não selecionaremos especificamente nenhuma linguagem para ser utilizada neste trabalho. O importante é a seleção do meta-modelo, pois com o modelo desenvolvido, a tradução para linguagens distintas torna-se trivial ao utilizar as ferramentas.

3.5. Ferramentas

Neste item mostraremos as ferramentas que apóiam a construção de ontologias, com algumas de suas características. Abaixo são exibidas algumas destas ferramentas:

- **Protege – 2000**

É uma ferramenta “*open source*”, “*stand-alone*”, de distribuição gratuita com arquitetura extensível, que permite a transformação da ontologia construída para outras linguagens como, XML, OWL, entre outras. O Protege permite a construção de ontologias de Udl, entradas de dados para criação de instâncias, bem como permite que outros aplicativos acessem sua base de dados. [Perez 04] [Breitman 04]

Atualmente, existem duas versões da ferramenta: *Frames* e OWL[Protege 00], que são versões para adaptar-se aos respectivos meta modelos.

- **Onto Edit**

É uma ferramenta com um ambiente extensível e flexível através da utilização de *plug-ins*, possibilitando transformar a ontologia desenvolvida para outras linguagens como RDF, XML, entre outras. Esta ferramenta está disponível em duas versões: uma *free* e uma profissional, com funcionalidades diferenciadas. [Perez 04] [Breitman 04]

- **OilEd**

Esta ferramenta foi desenvolvida em 2001 como um editor para OIL ontologias. Este editor não tem nenhuma metodologia de desenvolvimento associada e também não oferece suporte à integração, nem versionamento. Porém, é de fácil utilização pelo usuário, permitido a escrita e a checagem da ontologia (através do verificador FaCT), ou seja, checar se a ontologia desenvolvida está de acordo com o meta-modelo. Esta ferramenta é *free* e, para obtê-la, basta registrar-se no *site* e realizar o *download*. [Perez 04] [Breitman 04]

Para fins de utilização no nosso trabalho, selecionamos a ferramenta Protege em função de ser umas das ferramentas mais referenciadas na literatura, além de ter distribuição gratuita e possibilitar a tradução do modelo desenvolvido para algumas das principais linguagens de ontologias.

3.6. Conclusão

Neste capítulo apresentamos as principais definições e objetivos de ontologia, bem como os principais elementos que auxiliam para o desenvolvimento da mesma. São eles: os métodos, guiando a construção de ontologias de Udl, os meta-modelos, definindo as principais informações que a ontologia deve conter para modelar um Udl mais completo e as linguagens e ferramentas, auxiliando para melhor compreensão e facilitação de construção, respectivamente.

Podemos notar a importância da ontologia quando vemos a crescente utilização das mesmas como forma de estabelecer uma linguagem comum para compartilhar e reutilizar o conhecimento sobre o Udl do negócio, um dos objetivos da ontologia.[Cota 04]

Nos últimos dois capítulos, apresentamos as duas fontes de conhecimento (modelagem de processo de negócio e ontologia) que são utilizadas pelo método que será proposto no capítulo seguinte. Este método visa integrar o conhecimento obtido pelo modelo de processo do negócio e da ontologia do Udl.