



UNIVERSIDADE DE BRASÍLIA
FACULDADE DE ECONOMIA, ADMINISTRAÇÃO, CONTABILIDADE E
CIÊNCIA DA INFORMAÇÃO E DOCUMENTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA INFORMAÇÃO E DOCUMENTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA INFORMAÇÃO

Recuperação de informações sobre *log* de eventos apoiada em
ontologia

ANDRÉ LUIZ DIAS ESBÍZARO

Brasília
2006



UNIVERSIDADE DE BRASÍLIA
FACULDADE DE ECONOMIA, ADMINISTRAÇÃO, CONTABILIDADE E
CIÊNCIA DA INFORMAÇÃO E DOCUMENTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA INFORMAÇÃO E DOCUMENTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA INFORMAÇÃO

Recuperação de informações sobre *log* de eventos apoiada em ontologia

ANDRÉ LUIZ DIAS ESBÍZARO

Brasília
2006



UNIVERSIDADE DE BRASÍLIA
FACULDADE DE ECONOMIA, ADMINISTRAÇÃO, CONTABILIDADE E
CIÊNCIA DA INFORMAÇÃO E DOCUMENTAÇÃO
DEPARTAMENTO DE CIÊNCIA DA INFORMAÇÃO E DOCUMENTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA INFORMAÇÃO

Recuperação de informações sobre *log* de eventos apoiada em ontologia

ANDRÉ LUIZ DIAS ESBÍZARO

Dissertação apresentada ao Departamento de Ciência da Informação e Documentação da Universidade de Brasília, como parte dos requisitos para obtenção do título de Mestre em Ciência da Informação.

Orientadora:

Área de Concentração:

Linha de Pesquisa:

Prof^a Dr^a Marisa Bräscher Basílio Medeiros

Transferência da Informação

Arquitetura da Informação

Brasília
2006



UNIVERSIDADE DE BRASÍLIA
Faculdade de Economia, Administração, Contabilidade e Ciência da Informação e Documentação (FACE)
Departamento de Ciência da Informação e Documentação (CID)
Programa de Pós-Graduação em Ciência da Informação (PPGCIInf)

FOLHA DE APROVAÇÃO

Título: Recuperação de informações sobre *log* de eventos apoiada em ontologia

Autor: André Luiz Dias Esbízaro

Área de Concentração: Transferência da Informação

Linha de Pesquisa: Arquitetura da Informação

Dissertação submetida à Comissão Examinadora designada pelo Convênio Mestrado Interinstitucional (Minter) em Ciência da Informação do Programa de Pós-Graduação em Ciência da Informação (PPGCIInf) do Departamento de Ciência da Informação e Documentação (CID) da Universidade de Brasília (UnB) e a Universidade para o Desenvolvimento do Estado e da Região do Pantanal (UNIDERP) como requisito parcial para a obtenção do título de **Mestre em Ciência da Informação**.

Dissertação aprovada em: 22 de maio de 2006.

Aprovado por:

Prof.ª Dr.ª Marisa Bräscher Basílio Medeiros
Presidente – Orientador (UnB/PPGCIInf)

Prof. Dr. Mamede Lima Marques
Membro Interno - (UnB/PPGCINF)

Prof.ª Dr.ª Lígia Maria Arruda Café
Membro Externo – (UFSC)

Prof.ª Dr.ª Sofia Galvão Baptista
Suplente – (UnB/PPGCINF)

Aos meus pais, Hélio (*in memorian*) e Nilza (*in memorian*) por sempre acreditarem em
mim e terem sido meu exemplo de vida,

Aos meus irmãos, Hélio Jr e Helaine, por me incentivarem e auxiliarem em todos os
momentos da minha vida,

Aos meus tios, Carlos e Nair, e meus primos Fábio e Patrícia,
Por terem me ajudado tanto em Brasília,

Minha Gratidão.

Agradecimentos

A Deus pela dádiva da vida e do poder de escolha,

À minha orientadora, Prof^a Dr^a Marisa Bräscher Basílio Medeiros, que acreditou na minha capacidade para realizar meu mestrado. Obrigado, não somente pela ajuda nos momentos difíceis e pela oportunidade concedida, mas, principalmente, pela prova inequívoca de amizade, carinho, dedicação, seriedade, incentivo e de incansável disposição que me conduziu em todas as fases deste trabalho e pelos ensinamentos durante todo o curso,

A Professora Doutora Lígia Maria Arruda Café e ao Prof. Dr. Mamede Lima-Marques pela valiosa contribuição dada no exame de qualificação e pela disponibilidade em participar da banca examinadora,

Aos Professores do Programa de Pós-Graduação em Ciência da Informação da UnB pela competência, dedicação e amizade inquestionáveis demonstradas durante as aulas, nos seminários, nas palestras e nos encontros didáticos,

A todos os colegas que dividiram seu tempo comigo durante todo o mestrado. Em especial a vocês: Gilza, Inara, Jenner, Karita, Márcia, Paula (MS) e a Fernanda (DF) pelas discussões proporcionadas e experiências compartilhadas,

Aos Professores, Funcionários e Colegas da UNIDERP (Universidade para o Desenvolvimento do Estado e da Região do Pantanal), que também contribuíram de algum modo para a realização deste trabalho,

A todos aqueles que, embora não nominados aqui, direta ou indiretamente contribuíram para este trabalho,

Muito Obrigado.

RESUMO

A pesquisa propõe o desenvolvimento de uma ontologia de log de eventos do sistema operacional Microsoft Windows Server 2003, visando facilitar a recuperação de informações para apoiar o entendimento e a resolução de logs de eventos. Os sistemas de solução de eventos apresentam problemas relativos à falta de instrumentos para acesso ao conteúdo semântico dos registros e para armazenamento do conhecimento adquirido sobre log de eventos e à ausência de compreensão comum sobre logs de eventos e de relacionamentos entre os diferentes fatores que levam à ocorrência de erros nos sistemas operacionais. As etapas da metodologia abrangem: análise das características do site oficial *Events and Errors Message Center* e o programa aplicativo Event Viewer (Visualizar Eventos), levantamento de informações sobre log de eventos, modelagem da ontologia, simulação do uso da ontologia numa interface de busca sobre log de eventos e análise dos resultados obtidos na busca usando as ferramentas da Microsoft e na busca apoiada por ontologia. O resultado da pesquisa revela dificuldades no entendimento dos log de eventos, devido à dispersão das informações sobre logs nas ferramentas analisadas, à falta de relacionamento entre conceitos envolvidos na resolução dos logs, à ausência total de informações sobre determinados logs identificados. De maneira geral, conclui que a ontologia auxilia no entendimento dos log de eventos, mas que sua resolução depende de um nível mais profundo de descrição ontológica.

Palavras-Chave

Log de Eventos, Ontologia, Recuperação de Informação

ABSTRACT

This research proposes the development of an ontology of event logs in the Microsoft Windows Server 2003 operating system, aiming to facilitate the information retrieval to support the understanding and the resolution of event logs. The systems of solution of events present some problems relative to the lack of instruments to access the semantic content of the registers and storing the acquired knowledge on event logs and the absence of common understanding errors about event logs e the relationship among the different factors to the occurrence of operating system errors. The stages of the methodology enclose: analysis of the features of the official site *Events and Errors Message Center* and the application program Event Viewer (to visualize Events), survey of information on event logs, the modeling of the ontology, simulation of the use of the ontology in an interface of search on event logs and the analysis of the results gotten in the search using the Microsoft tools and in the search supported by the ontology. The result of the research discloses difficulties in the understanding of event logs, due to dispersion of the information on logs in the analyzed tools, to the lack of relationship between the involved concepts in the resolution of logs, and the total absence of information on determined identified logs. In general way, it concludes that the ontology assists in the understanding of event logs, but that its resolution depends on a deeper level of ontologic description.

Keywords

Event Log, Ontology, Information Retrieval

LISTA DE FIGURAS

Figura 1	Propriedades do evento selecionado aleatoriamente.	22
Figura 2	Resultado obtido no link das propriedades do evento.	23
Figura 3	Site Events and Errors Message Center.	23
Figura 4	Resultado obtido na busca pelo site (Language: English).	24
Figura 5	Resultado obtido na busca pelo site (Language: Portuguese - Portugal).	25
Figura 6	Tela inicial do programa aplicativo Visualizar Eventos.	30
Figura 7	Propriedades do evento selecionado aleatoriamente.	33
Figura 8	Tela inicial da “Busca Básica” (Basic Search).	38
Figura 9	Home Page do site <i>Events and Errors Message Center</i>	39
Figura 10	Resultado obtido na busca pelo site (Language: English).	41
Figura 11	Um sistema de recuperação de informações típico.	43
Figura 12	Componentes de um sistema de recuperação de informação.	44
Figura 13	A coleta de dados na <i>Web</i> : a superfície e a <i>Web</i> profunda.	48
Figura 14	Adaptação do triângulo do significado de Ogden e Richards.	52
Figura 15	Hipótese da representação do conhecimento.	57
Figura 16	Modelagem da memória humana.	62
Figura 17	Exemplo de rede semântica na representação do conhecimento.	63
Figura 18	Frames.	64
Figura 19	Exemplo da utilização de <i>frames</i> na representação do conhecimento.	65
Figura 20	Representação simplificada de um neurônio.	67
Figura 21	Representação de uma rede neural artificial.	68
Figura 22	Estrutura de um sistema especialista.	70
Figura 23	Exemplo de Rede Bayesiana.	74
Figura 24	<i>Framework</i> de Modelagem de Domínio.	74
Figura 25	Especificação explícita de uma conceitualização.	80
Figura 26	Especificação explícita de uma conceitualização na forma gráfica.	80
Figura 27	Categorização de conceitos.	81
Figura 28	Níveis de representação para o conhecimento.	82
Figura 29	Usos para ontologias.	94
Figura 30	Ontologia como Inter-Lingua.	96
Figura 31	Exemplo de ontologia como inter-lingua.	97
Figura 32	Diferentes níveis da taxonomia do vinho.	102
Figura 33	Tela inicial do Protege.	120
Figura 34	Definição da sintaxe e semântica dos dados.	122
Figura 35	Representação em grafo.	124
Figura 36	Elementos do RDF Schema.	126
Figura 37	Antigo Site Windows Server 2003 Events and Errors.	138
Figura 38	Microsoft Virtual PC 2004.	139
Figura 39	Microsoft Windows Server 2003 Enterprise Edition.	140
Figura 40	Seqüência utilizada nas instalações das máquinas virtuais.	141
Figura 41	<i>Registry</i> do Windows.	141
Figura 42	Mecanismo de busca oficial da Microsoft.	142
Figura 43	Resultado do mecanismo de busca oficial da Microsoft.	143
Figura 44	Resultado do <i>link</i> selecionado no mecanismo de busca oficial.	143
Figura 45	Modelagem inicial da ontologia de domínio.	147
Figura 46	Modelagem inicial da ontologia de domínio.	156

Figura 47	Exemplos da complexidade do domínio.....	161
Figura 48	interface de busca apoiada em uma ontologia (simulação)	162
Figura 49	Eventos Acpi (5 e 4)	165
Figura 50	Eventos atapi (26)	165
Figura 51	Ontologia parcial de Software	166
Figura 52	Eventos cdrom (7, 11 e 51).....	167
Figura 53	Ontologia parcial de Trilhas	167
Figura 54	Eventos disk (11 e 11)	168
Figura 55	Eventos dmboot (2 e 1).....	169
Figura 56	Eventos Save Dump (1001).....	170
Figura 57	Eventos ntfs (50) e w32time (14)	170
Figura 58	Eventos Rasman (20159).....	171
Figura 59	Ontologia parcial de Dispositivo de Rede.....	172
Figura 60	Eventos Server (8022, 8021, 8015, 7027, 3100 e 14503).....	172

LISTA DE TABELAS

Tabela 1	Solução hipotética de um problema (Visualizar Eventos).....	25
Tabela 2	Resumo das fontes pré-registradas encontradas	148
Tabela 3	Eventos dos Aplicativos que obtiveram resultados na busca.	150
Tabela 4	Eventos de Sistema que obtiveram resultados na busca.....	151
Tabela 5	Resultados na busca no site oficial da Microsoft.....	157

LISTA DE ABREVIATURAS E SIGLAS

AD	ACTIVE DIRECTORY
AI	ARTIFICIAL INTELLIGENCE
CASE	COMPUTER-AIDED <i>SOFTWARE</i> ENGINEERING
CIRL	COMPUTATIONAL INTELLIGENCE RESEARCH LABORATORY
CML	CONCEPTUAL MODELING LANGUAGE
DAML+OIL	DARPA AGENT MARKUP LANGUAGE
DNS	DOMAIN NAME SYSTEM
DTD	DOCUMENT TYPE DEFINITIONS
FAT	FILE ALLOCATION TABLE
FIPA	FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS
IP	INTERNET PROTOCOL
ISI	INFORMATION SCIENCES INSTITUTE
ISO	INTERNATIONAL STANDARDS ORGANIZATION
KIF	KNOWLEDGE INTERCHANGE FORMAT
KR	KNOWLEDGE REPRESENTATION
NT	NEW TECHNOLOGY
NTFS	NT FILE SYSTEM
OIL	ONTOLOGY INFERENCE LAYER
OWL	ONTOLOGY WEB LANGUAGE
PAL	PROTÉGÉ AXIOMATIC LANGUAGE
RDF	RESOURCE DESCRIPTION FRAMEWORK
SHOE	SIMPLE HTML ONTOLOGY EXTENSIONS
TOVE	TORONTO VIRTUAL ENTERPRISE
XML	EXTENSIBLE MARKUP LANGUAGE
XOL	ONTOLOGY EXCHANGE LANGUAGE
XP	EXPERIENCE
XTM	XML TOPIC MAPS
WWW	WORLD WIDE WEB
W3C	WORLD WIDE WEB CONSORTIUM

SUMÁRIO

INTRODUÇÃO	14
1.1 CONTEXTUALIZAÇÃO	16
1.2 DEFINIÇÃO DO PROBLEMA	20
1.3 QUESTÃO DE PESQUISA	26
1.4 OBJETIVOS	26
1.4.1 OBJETIVO GERAL.....	26
1.4.2 OBJETIVOS ESPECÍFICOS.....	27
1.5 JUSTIFICATIVA	27
REVISÃO DE LITERATURA	29
2.1 LOG DE EVENTOS	29
2.1.1 VISUALIZAR EVENTOS.....	30
2.1.2 SISTEMAS OPERACIONAIS	35
2.1.2.1 Microsoft Windows.....	35
2.1.3 SITE <i>EVENTS AND ERRORS MESSAGE CENTER</i>	37
2.2 RECUPERAÇÃO DE INFORMAÇÃO	42
2.2.1 MECANISMOS DE BUSCA	46
2.3 TERMINOLOGIA	49
2.3.1 LÍNGUA DE ESPECIALIDADE.....	50
2.3.2 UNIDADE TERMINOLÓGICA	51
2.3.2.1 Conceito	52
2.3.2.2 Termo.....	54
2.4 REPRESENTAÇÃO DO CONHECIMENTO	57
2.4.1 FORMAS DE REPRESENTAÇÃO DO CONHECIMENTO.....	62
2.4.1.1 Redes semânticas	62
2.4.1.2 <i>Frames</i>	64
2.4.1.3 Redes Neurais	67
2.4.1.4 <i>Scripts</i>	68
2.4.2 REPRESENTAÇÃO DO CONHECIMENTO NA INTELIGÊNCIA ARTIFICIAL.....	69
2.4.2.1 Sistemas especialistas	69
2.4.2.2 Sistema de resolução de problemas.....	71
2.4.2.3 Agentes Inteligentes	72
2.4.2.4 Redes Bayesianas.....	73
2.4.3 MODELAGEM DE DOMÍNIO	74
2.5 ONTOLOGIAS	75
2.5.1 ORIGEM.....	77
2.5.2 DEFINIÇÕES	78
2.5.3 CLASSIFICAÇÃO E FORMALISMO.....	89
2.5.4 USO DA ONTOLOGIA	91
2.5.4.1 Comunicação	94
2.5.4.2 Interoperabilidade	95
2.5.4.3 Engenharia de sistemas.....	98
2.5.5 METODOLOGIAS.....	98
2.5.5.1 Uschold e King	102
2.5.5.2 Grüninger e Fox.....	104
2.5.5.3 Amaya Berneras <i>et al.</i>	105
2.5.5.4 Fernández-López <i>et al.</i>	106

2.5.5.5 Swartout et al.	108
2.5.6 ROTEIRO PARA A CONSTRUÇÃO DA ONTOLOGIA.....	109
2.5.6.1 Determinação do domínio e escopo da ontologia	110
2.5.6.2 Pesquisa sobre ontologias e outras estruturas existentes no domínio.....	113
2.5.6.3 Enumerar os termos do domínio	113
2.5.6.4 Definição das classes, hierarquia e propriedades.....	114
2.5.6.5 Definição de restrições sobre as relações e determinação de instâncias.....	115
2.5.6.6 Implementação e instanciação da ontologia	116
2.5.7 FERRAMENTAS E LINGUAGENS	116
2.5.7.1 Protégé	119

PROCEDIMENTOS METODOLÓGICOS.....135

3.1 CLASSIFICAÇÃO DA PESQUISA.....	135
3.2 UNIVERSO DA PESQUISA.....	137
3.2.1 AMOSTRA	138
3.2.1.1 Instalação do Virtual PC 2004.....	139
3.2.1.2 Instalação do sistema operacional Windows Server 2003	140
3.2.1.3 Busca nos registros internos do Windows Server 2003	141
3.2.1.4 Busca no site oficial da Microsoft.....	142
3.2.2 ROTEIRO PARA A CONSTRUÇÃO DA ONTOLOGIA.....	144
3.2.3 TESTE DA ONTOLOGIA	152

ANÁLISE E RESULTADOS.....153

4.1 ANÁLISE DAS FERRAMENTAS DA MICROSOFT CORPORATION	153
4.2 MODELAGEM DA ONTOLOGIA	154
4.3 INTERFACE SIMULADA DE BUSCA	160
4.4 ANÁLISE DAS BUSCAS.....	164

CONSIDERAÇÕES FINAIS E SUGESTÕES DE PESQUISA.....174

4.1 SUGESTÃO DE TRABALHOS FUTUROS.....	176
---	------------

REFERÊNCIAS.....178

1

Introdução

Com a grande evolução das tecnologias, cada dia mais pessoas e sistemas computacionais estão conectados por meio de redes de computadores e ficam sujeitos a falhas que precisam ser corrigidas. Nesse contexto, falhas são inevitáveis, mas as suas consequências, principalmente nas empresas que estão cada vez mais dependentes da tecnologia, têm se tornado mais presentes nas organizações.

Atualmente, sistemas computacionais têm armazenado, processado e recuperado grandes quantidades de informação, tornando as tarefas de manipular e gerenciar essa informação bastante custosas. O colapso do sistema, a interrupção no fornecimento do serviço e a perda de dados devem ser evitados pelo uso adequado de técnicas viáveis e de fácil compreensão.

Tecnologias como sistemas operacionais, programas aplicativos, dispositivos de segurança e outros, têm auxiliado na identificação de falhas com o intuito de facilitar a geração de conhecimento que possa ser utilizado para que administradores de sistemas tomem decisões a respeito de problemas existentes.

As falhas registradas pelos diferentes tipos de tecnologias são denominadas *logs*. Estes gravam dados de qualquer ocorrência ou *evento* significativo no sistema ou em um aplicativo que exija que os usuários sejam notificados ou que uma entrada seja adicionada ao *log*.

As informações registradas em *log* de eventos não apresentam uma descrição semântica que possibilite correlacionar eventos ocorridos. Mesmo empresas como a Microsoft Corporation, que apresenta em seus sistemas operacionais aplicativos capazes de visualizar os eventos ocorridos, não realiza uma correlação entre as diferentes entradas e, ainda, o fato das descrições não seguirem um padrão de estrutura dificulta a

extração automática do valor semântico dos dados, ficando, assim, a cargo do administrador analisar essas descrições.

Com o intuito de facilitar a recuperação de informações para a resolução de *log* de eventos, propõe-se neste trabalho o desenvolvimento de uma ontologia que possa apoiar um mecanismo de busca em *log* de eventos dos sistemas operacionais da Microsoft Corporation.

O trabalho está estruturado da seguinte maneira: no primeiro capítulo, introdutório ao tema, a **Seção 1.1** contextualiza o desenvolvimento do estudo da representação do conhecimento, recuperação da informação e mecanismos de busca apoiados em ontologias, apresentando sistematicamente suas principais contribuições na busca de melhorias nos processos de recuperação da informação; as seções seguintes deste capítulo expressam a definição do problema (**Seção 1.2**), questão de pesquisa (**Seção 1.3**), objetivos: geral e específicos (**Seção 1.4** e **subseções 1.4.1** e **1.4.2**) e a justificativa que se encontra na **Seção 1.5**.

No **Capítulo 2** - Revisão de Literatura, a **Seção 2.1** e subseções aborda o *log* de eventos, *locus* da pesquisa, onde se apresenta sua definição, seus tipos, e sua importância no ambiente computacional contemporâneo, o programa aplicativo *Event Viewer* (Visualizar Eventos) da empresa que apresenta os eventos ocorridos, bem como os sistemas operacionais da Microsoft Corporation, e ainda, o site oficial *Events and Errors Message Center*¹ da empresa utilizado para se buscar informações e soluções sobre os *log* de eventos apresentados. A **Seção 2.2** e subseções trazem breves definições dos processos de recuperação de informação e mecanismos de busca apoiados em semântica, sem a pretensão de abordar aqui todas as iniciativas existentes e de ser exaustivo, apesar da preocupação em cobrir os itens mais representativos descritos recentemente na literatura. A **Seção 2.3** apresenta brevemente a Terminologia, o Conceito e suas características. A **Seção 2.4** e subseções abordam sucintamente a representação do conhecimento e suas formas, incluindo sua utilização na Inteligência Artificial. Já a **seção 2.5** e subseções abordam a ontologia de maneira comentada: origem e definições, tipos, metodologias, ferramentas e métodos de construção e

¹ http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

avaliação, suas principais características e finalidades, bem como, uma breve descrição da ferramenta Protégé, algumas características do XML, XML Schema, DTD, OWL, OWL Lite, OWL DL, OWL Full e RDF utilizados na modelagem e apresentação da ontologia.

O **Capítulo 3** - Procedimentos Metodológicos apresenta na **Seção 3.1** a classificação da pesquisa. Na **Seção 3.2** e subseções, o universo da pesquisa, a amostra e a descrição da metodologia que orientou a construção e a manipulação da ontologia, baseado na proposta de Noy e McGuinness (2001) e de Uschold *et al* (1997) que argumentam que não há uma maneira ou metodologia “correta” para desenvolver ontologias e, finalmente, como os testes foram efetuados e a proposta de análise dos registros.

O **Capítulo 4** - Análise e Resultados aborda os resultados alcançados segundo objetivos propostos neste trabalho, na **Seção 4.1** faz uma análise dos recursos apresentados nas duas ferramentas da Microsoft Corporation. Na **Seção 4.2** apresenta o processo de modelagem da ontologia em questão. A **Seção 4.3** demonstra o funcionamento da interface simulada de busca e, finalmente, a comparação entre os mecanismos de busca das **Seções 4.1** e **4.3**, são apresentados na **Seção 4.4** a Análise das buscas, apresentando os problemas e soluções encontrados.

O **Capítulo 5** - Considerações Finais e Sugestões de Pesquisa, apresentam as conclusões relevantes referentes ao trabalho realizado e as sugestões para trabalhos futuros.

1.1 Contextualização

A automação, originada nas profundezas históricas da humanidade, antecedeu a computação, onde o homem em sua busca insaciável e infinita por ferramentas que possam fazer suas tarefas, tenta automatizar as coisas rotineiras da vida.

Ao longo dos anos, o ser humano foi desenvolvendo e aperfeiçoando essas ferramentas que facilitavam estas operações, de tal maneira que conseguiu construir o computador.

A evolução social do homem confunde-se com as tecnologias desenvolvidas e empregadas em cada época. Diferentes épocas da história da humanidade são historicamente reconhecidas, pelo avanço tecnológico correspondente. As idades da pedra, do ferro e do ouro, por exemplo, correspondem ao momento histórico-social em que foram criadas “novas tecnologias” para o aproveitamento desses recursos da natureza de forma a garantir melhor qualidade de vida. O avanço científico da humanidade amplia o conhecimento sobre esses recursos e cria permanentemente “novas tecnologias”, cada vez mais sofisticadas. (KENSKI, 2003, p.20)

Não se pode discutir que o computador hoje em dia já faz parte de nossas vidas, assim como é o rádio, a televisão, o micro-ondas, o automóvel, enfim, tudo aquilo que o homem criou para substituí-lo nas suas tarefas e que, no passado, causaram grande impacto.

Hoje, os computadores cada vez mais avançados possibilitam cálculos em fração de segundos, e, desta feita, vemos que com o seu uso e acesso às informações de forma cada vez mais facilitada, a humanidade está ampliando suas possibilidades.

Com o advento da Internet, as buscas por informação se modificaram substancialmente, as tentativas de classificar os primeiros milhares de páginas *Web* que surgiam na tela do computador e de oferecer um serviço de busca em centenas de meios de comunicação, atualizado automaticamente a cada minuto, e com a capacidade de agrupá-las tematicamente ficaram cada vez mais difíceis, fazendo surgir novos estudos que mostram que a forma das pessoas se relacionarem, de buscar informação, de consumir conhecimento, de inventar, de se mover já não é a mesma.

Entre outras coisas, as pessoas estão mais seletivas, exigentes, mais conscientes do que a Internet tem a oferecer, e com isso têm surgido ferramentas mais potentes para o tratamento e a recuperação de informação que é objeto de estudo da Ciência da Informação e que teve um grande desenvolvimento na década de 60, bem como as formas de representação de conhecimento nelas utilizadas.

Este vasto campo de estudo se amplia a cada dia, especificamente com a *Web*, visto que o volume de informações públicas e gratuitas neste ambiente é um excelente material para o avanço das pesquisas em recuperação de informação. Computadores gigantescos estão disponíveis para rodar algoritmos complexos de recuperação e de ordenação em bases de dados, funcionando em tempo real tolerável. Iniciativas privadas e governamentais apóiam pesquisas acadêmicas em bibliotecas digitais, oferecendo um campo para explorar a descoberta e a recuperação em rede em ambiente controlado. Por outro lado, há coleções de bases de dados muito grandes, de alguma forma heterogênea, e de propriedade de empresas cujos interesses comerciais são a melhoria do projeto de interface e a efetividade da recuperação.

Assim, conforme conclui Schwartz (1998), raras foram as situações onde, no campo da Ciência da Informação, os interesses da pesquisa acadêmica e do desenvolvimento comercial de produtos estiveram tão próximos. Quando trazemos a questão da recuperação de informação para o ambiente da *Web* estamos nos referindo aos mecanismos de busca, já que são estes que tornam possível a recuperação de informação na Internet.

Logo após a chegada da *Web*, os mecanismos de busca tornam-se uma importante área de pesquisa, quando a tarefa de coleta e indexação começou a exceder a capacidade humana e os recursos técnicos disponíveis e, em 1996, as ferramentas de busca começaram a ser anunciadas em jornais, revistas e TV com a proliferação de diferentes produtos de busca - diretórios, motores de busca, meta ferramentas, serviços baseados em assunto, agentes pessoais e serviços de disseminação da informação (tecnologia *push*).

O crescimento dos estudos em mecanismos de busca pode ser avaliado pela extensa bibliografia existente. Estes documentos envolvem fontes de diferentes procedências, como revistas de informática e artigos de divulgação. Seus autores são da Ciência da Informação, da Ciência da Computação e de empresas de Marketing.

A revista PC Week (2000) em uma retrospectiva da Internet apresentou que:

- A primeira geração da rede, a Internet, permitia apenas a troca de dados entre *máquinas* distintas;
- A segunda geração, a *World Wide Web* (teia de alcance mundial), ou simplesmente *Web*, provocou uma revolução por disponibilizar uma vasta gama de aplicativos e informações para as *pessoas*, tornando possível, ainda, o comércio eletrônico (*e-commerce*) entre clientes e empresas;

Porém, como as páginas só possuem informação léxica, mesmo os agentes e/ou robôs dotados de inferência encontram um ambiente hostil para a realização de suas tarefas, porque tanto o conteúdo das páginas como o relacionamento entre elas é difícil de ser compreendido por agentes de *software*, por encontrarem-se, em sua maioria, em linguagem natural.

As pessoas, em consequência, sofrem com essa falta de definições semânticas: por vezes os usuários possuem dados parciais sobre a informação que procuram e não podem utilizá-los para localizar esta informação.

A próxima geração da rede está sendo batizada de *Web Semântica*. Sua maior motivação é transformar os dados e aplicativos em elementos úteis, legíveis e compreensíveis para o *software*, ou, mais exatamente, para os *agentes inteligentes*, estudados na Inteligência Artificial, de forma a facilitar-lhes a comunicação dinâmica, a cooperação e o comércio eletrônico entre empresas.

As primeiras camadas da *Web Semântica* dispõem, desta maneira, de padrões de registro dos documentos e especificam seu preenchimento com termos inseridos numa estrutura significativa. A *Web Semântica* retoma procedimentos adotados pela Ciência da Informação desde a década de 70, qual seja: a definição de formatos de intercâmbio de registros bibliográficos e o desenvolvimento de esquemas de representação do conhecimento, entre outros.

As pesquisas atuais na *Web Semântica* têm como principal enfoque as ontologias. Essa tendência é evidenciada pelo desenvolvimento de uma variedade de

sistemas e arquiteturas visando prover a integração de ontologias, a criação de linguagens de representação do conhecimento, de bibliotecas e editores de ontologias.

As ontologias, ao ordenarem conceitos, incorporam à *Web* a preocupação com a organização da informação e, conseqüentemente, da atribuição de significados aos termos. A inclusão de conceitos numa estrutura, qualquer que seja esta estrutura, contextualiza a informação pela localização de um conceito em relação aos demais inseridos no sistema conceitual.

“Diante da necessidade de construir Arquiteturas de Informação, que ofereçam suporte para o entendimento da semântica dos dados, as pesquisas retornam aos princípios básicos usados como fundamentos na construção de arquiteturas, tais como: [...] as ontologias” (LIMA-MARQUES, 2004)

As ontologias apresentam vários benefícios, tais como facilitar a comunicação entre pessoas, facilitar o raciocínio e entendimento de determinado domínio, servir de referência para a obtenção de um consenso e eliminar contradições e inconsistências por meio de uma especificação não ambígua. Do ponto de vista metodológico, apesar de uma grande quantidade de ontologias já ter sido desenvolvida, por diferentes grupos, sob diferentes abordagens e usando diferentes métodos e técnicas, poucos trabalhos foram publicados sobre como proceder, mostrando as práticas, critérios de projeto, atividades, métodos e ferramentas utilizadas para a sua construção.

Desta forma, este trabalho busca a modelagem de uma ontologia que ajude no entendimento e na resolução de log de eventos registrando as etapas de construção da ontologia e as facilidades e dificuldades encontradas e demonstrando o potencial de sua utilização numa interface de busca.

1.2 Definição do problema

Em ambientes computacionais, o administrador tem em suas mãos uma grande quantidade de dados e informações que devem ser gerenciados e analisados a fim de tomar uma decisão caso o sistema apresente algum problema inesperado ou alguma atividade suspeita.

Esses dados podem ser de fontes² diretamente relacionadas a eventos³ ocorridos e/ou de fontes que podem caracterizar uma queda de desempenho nesse ambiente. Devido à grande quantidade de dados e informações disponíveis, torna-se muitas vezes imprescindível seu processamento em tempo hábil para uma tomada de decisão antes que um incidente alcance grandes proporções.

Nesse sentido, diversas empresas têm realizado esforços para catalogar e classificar dados e informações relacionadas ao ambiente computacional. A Microsoft Corporation, por exemplo, criou o programa aplicativo *Event Viewer* (Visualizar Eventos) que apresenta um padrão para nomenclatura de eventos ocorridos em seus produtos, com a finalidade de facilitar a identificação e resolução dos mesmos. Este aplicativo encontra-se abordado na **Seção 2.1.1**.

Outras empresas também registram a grande quantidade de informações relacionadas a violações de sistemas, mensagens de erros, alertas etc. Apesar dos avanços dessas empresas, tais esforços não agregam semântica às informações das ocorrências armazenadas e divulgadas. Entendendo-se por semântica a capacidade de processar o significado dos conceitos e não apenas a estrutura da informação. Sem o entendimento semântico, o usuário, profissional da área de informática ou agente de *software* não pode fazer relacionamentos, seja entre eventos de diversas fontes, seja entre eventos divulgados por uma mesma fonte.

Essa correlação é importante para estabelecer um vocabulário de conceitos e relacionamentos único e formal a respeito de *log* de eventos. Uma ontologia poderá contribuir no sentido de estabelecer esse vocabulário único.

² **Fonte** (*Source*) é o software que registrou o evento, que pode ser um nome de programa, como o "Word 6" ou um componente do sistema ou de um grande programa, como um nome de *driver*. (MICROSOFT, Visualizar Eventos, versão 5.1.2600.2180. Microsoft Corporation, Microsoft Windows)

³ **Evento** (*Event*) é qualquer ocorrência significativa no sistema ou em um aplicativo que exija que os usuários sejam notificados ou que uma entrada seja adicionada a um *log*. (MICROSOFT, Centro de Ajuda e Suporte, Microsoft Corporation, Microsoft Windows)

Neste trabalho propõe-se o desenvolvimento de uma ontologia capaz de representar e relacionar conceitos a respeito de *log* de eventos atualmente conhecidos nos sistemas operacionais da Microsoft Corporation.

A grande motivação para realizar este trabalho diz respeito à dificuldade de localizar informação para orientar os administradores e/ou usuários de sistemas na solução dos eventos apresentados pelo programa aplicativo *Event Viewer* (Visualizar Eventos) existente em alguns dos sistemas operacionais da Microsoft Corporation, abordado na **Seção 2.1.2**.

Na busca de informações, os usuários deparam-se com problemas decorrentes do modelo atual utilizado pela Microsoft, como por exemplo:

- Os mecanismos de buscas atuais da Microsoft Corporation não apresentam resoluções para todos os *log* de eventos apresentados, ou então, apresentam mais de uma resolução de forma ambígua, conforme exemplos apresentados nas figuras 1, 2 e 5;

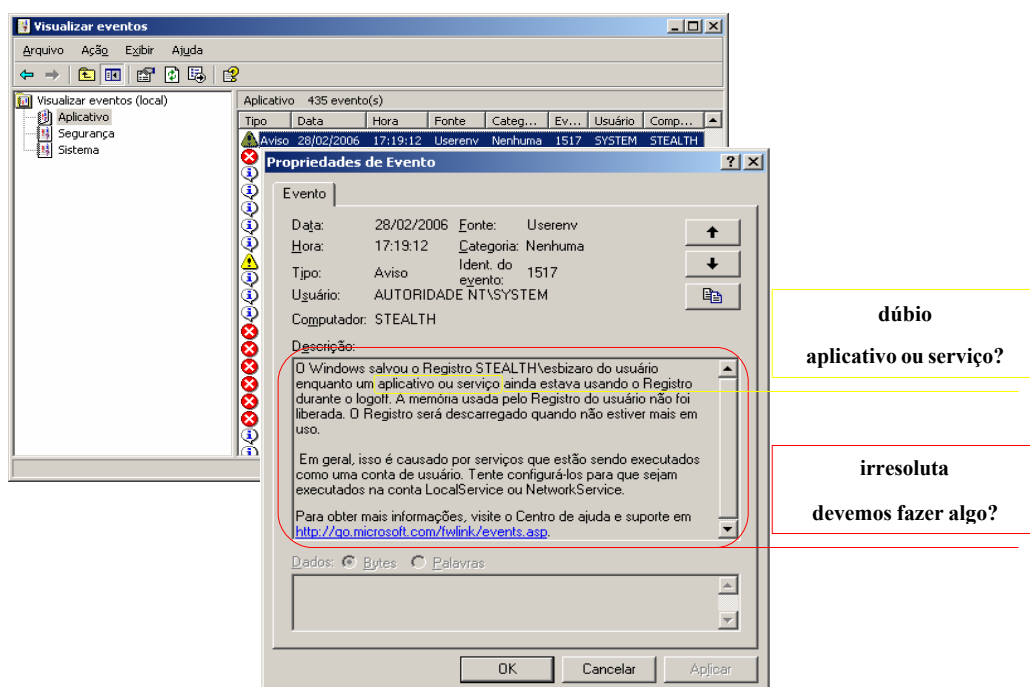


Figura 1 Propriedades do evento selecionado aleatoriamente.
Fonte: MICROSOFT, Visualizar Eventos, versão 5.1.2600.2180



Figura 2 Resultado obtido no link das propriedades do evento.
Fonte: MICROSOFT, Centro de Ajuda e Suporte

- O site oficial *Events and Errors Message Center* da empresa utilizado para se buscar informações e soluções sobre os *log* de eventos é em inglês, o que se constitui uma barreira a muitos usuários, figuras 3 e 4;
- A falta de compreensão comum sobre os *log* de eventos dificulta a resolução dos eventos ocorridos, o que muitas vezes só acaba sendo possível por um especialista da própria Microsoft, conforme exemplos das figuras 3, 4 e 5;

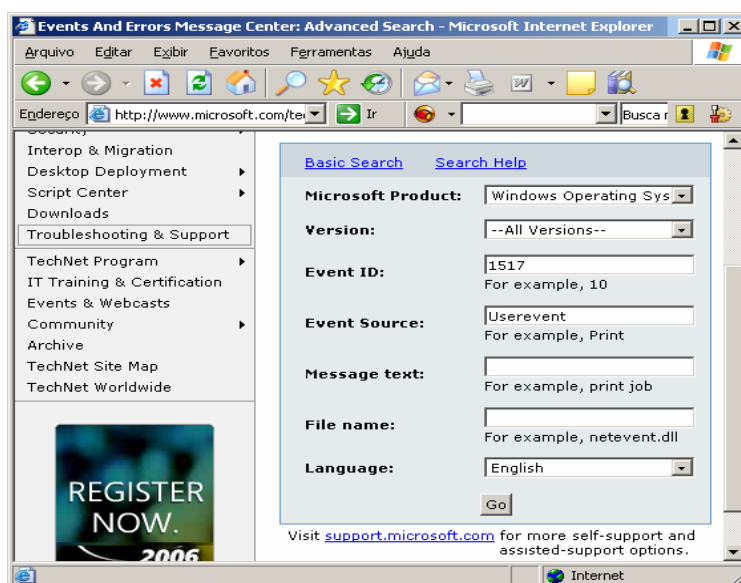


Figura 3 Site Events and Errors Message Center.

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

- Como o *log* de eventos não registra a resolução dos mesmos por parte dos administradores de sistemas, torna-se custosa a sua resolução em caso de recorrência da falha, pois o conhecimento adquirido não é armazenado e reaproveitado;
- Diferentes sistemas computacionais armazenam as informações de seus *log* de eventos de forma isolada, o que dificulta a resolução dos problemas. Por exemplo, o **Servidor A** pode apresentar um erro relacionado à replicação⁴ de dados com o **Servidor B**, que deve estar apresentando algum outro tipo de erro, o qual, se não tivermos acesso, não permitirá a resolução do problema;

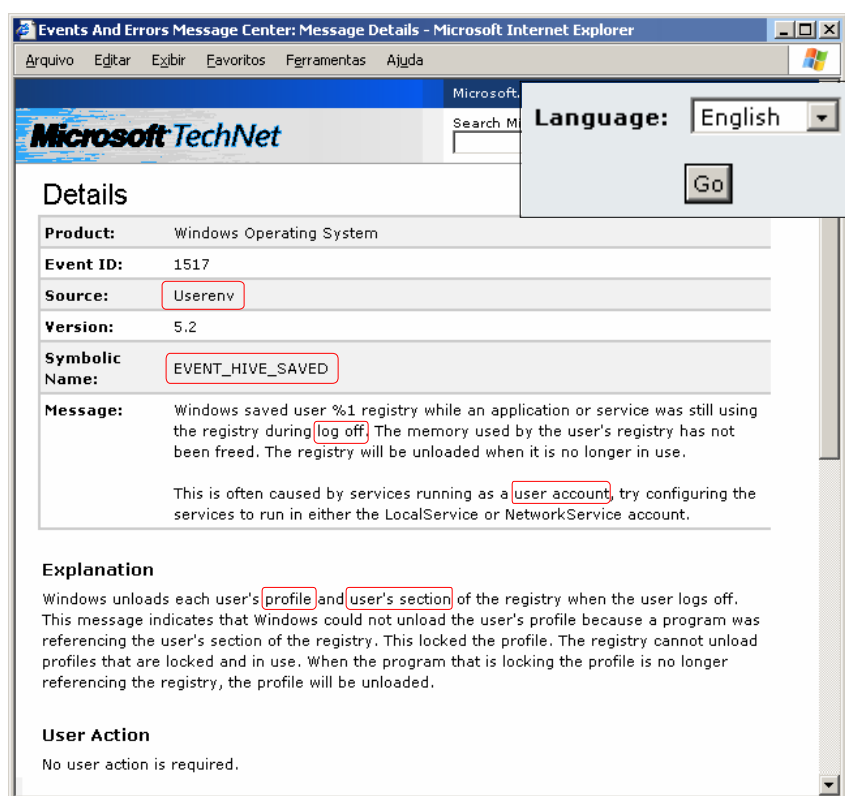


Figura 4 Resultado obtido na busca pelo site (Language: English).

Fonte: http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

⁴ **Replicação** é o processo de copiar dados de um armazenamento de dados ou sistema de arquivos para vários computadores a fim de sincronizar os dados. O *Active Directory* fornece replicação multimestre do diretório entre controladores de domínio dentro de determinado domínio. É possível gravar nas réplicas do diretório em cada controlador de domínio. Isso permite a aplicação de atualizações a todas as réplicas de determinado domínio. O serviço de replicação copia automaticamente as alterações de determinada réplica para todas as outras réplicas. (MICROSOFT, Microsoft Computer Dictionary, Fifth Edition eBook)

- A estrutura dos registros de *log* de eventos não permite a captura da semântica dos dados por um mecanismo de processamento automático;

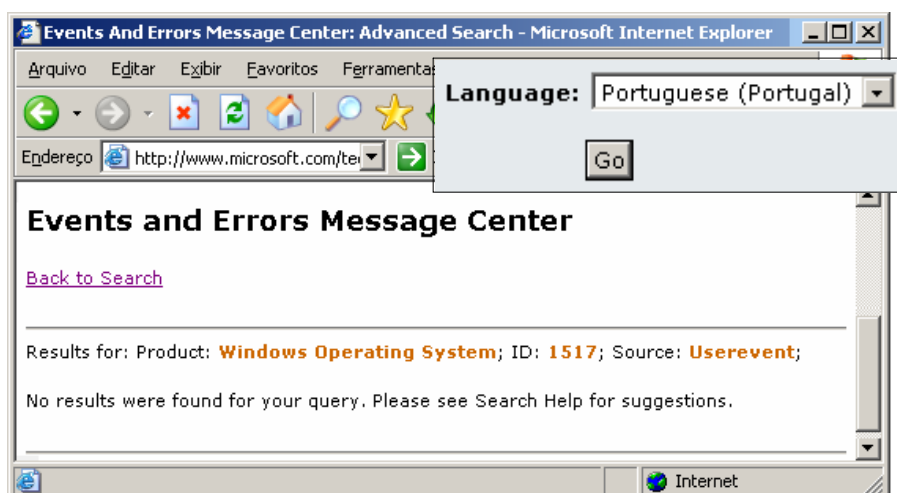


Figura 5 Resultado obtido na busca pelo site (Language: Portuguese - Portugal).
Fonte: http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Esses problemas são ainda mais críticos no caso do domínio em questão, já que novos eventos são descobertos todos os dias.

Por exemplo, imagine o programa aplicativo *Event Viewer* (Visualizar Eventos) apresentando o seguinte cenário:

Tabela 1 Solução hipotética de um problema (Visualizar Eventos).

	hipótese	solução
1	O <u>Servidor A</u> apresenta uma mensagem de erro no <i>log</i> do serviço de replicação de arquivos , que mostra que não foi possível o sincronismo dos dados com o <u>servidor B</u> .	O Administrador ou usuário, ao perceber o evento, buscaria efetuar manualmente o processo de sincronismo, que poderia ter sido ocasionado por um “gargalo” na rede, em caso de insucesso, buscaria informações sobre a conectividade da rede ou no próprio <u>Servidor B</u> .
2	O <u>Servidor A</u> apresenta também, uma mensagem de erro no <i>log</i> do sistema , que mostra que sua placa de rede não	O Administrador ou usuário, ao perceber o evento, desabilitaria e reabilitaria a placa de rede, trocaria o

	esta funcionando adequadamente.	<i>driver</i> da placa de rede, ou até mesmo, a própria placa de rede.
3	O <u>Servidor B</u> apresenta uma mensagem de erro no <i>log</i> do sistema , que mostra que o protocolo de rede IP não esta funcionando adequadamente.	O Administrador ou usuário, ao perceber o evento, verificaria se o protocolo de rede IP não foi desabilitado por algum motivo.
conclusão		<i>Falta relacionamento nas informações apresentadas.</i>

Fonte: autoria nossa

Assim, uma ontologia poderá ajudar as pessoas a raciocinar e entender sobre os *log* de eventos, atuando, dessa forma, como apoio para a obtenção de informações na resolução dos eventos ocorridos.

1.3 Questão de pesquisa

O uso de uma ontologia sobre *log* de eventos⁵ do sistema operacional Microsoft Windows Server 2003 facilitaria a recuperação de informações para apoiar o entendimento e a resolução dos mesmos?

1.4 Objetivos

1.4.1 Objetivo geral

Modelar uma ontologia de *log* de eventos do sistema operacional Microsoft Windows Server 2003 que visa facilitar a recuperação de informações para apoiar o seu entendimento e a sua resolução.

⁵ **Log de eventos** é o arquivo que contém informações e mensagens de erro para todas as atividades do computador. (MICROSOFT, Microsoft Computer Dictionary, Fifth Edition eBook)

1.4.2 Objetivos específicos

- analisar os recursos que as duas ferramentas da Microsoft Corporation, o site oficial *Events and Errors Message Center* e o programa aplicativo *Event Viewer* (Visualizar Eventos) oferecem aos usuários para busca de informações sobre *log* de eventos;
- modelar uma ontologia de *log* de eventos com a especificação de classes, propriedades e relacionamentos, baseada inicialmente nas informações contidas no programa aplicativo *Registry* (Editor do registro⁶) do sistema operacional Microsoft Windows Server 2003 Enterprise Edition, e instanciada pelos registros encontrados o site oficial *Events and Errors Message Center* e no site *Microsoft Windows Server 2003 TechCenter* da mesma empresa.
- simular o funcionamento de uma interface de busca apoiada em uma ontologia;
- analisar os resultados das buscas sobre *log* de eventos realizados com as duas ferramentas da Microsoft Corporation disponíveis e os obtidos na busca apoiada pela ontologia;

1.5 Justificativa

Atualmente, o registro das ocorrências diárias realiza-se por meio de *log* de eventos, que tornou possível a análise e transmissão dos mesmos de uma pessoa a outra. Tais eventos constituem uma fonte quase inesgotável, já que novos eventos surgem a cada dia.

⁶ **Registro** é um repositório de banco de dados para informações sobre a configuração de um computador. O Registro contém informações que o Windows consulta continuamente durante o seu funcionamento. O Registro é organizado hierarquicamente como uma árvore e é formado por chaves e suas subchaves, seções e entradas de valores. (MICROSOFT, Microsoft Computer Dictionary, Fifth Edition eBook)

A organização dessas informações dispersa e fragmentadas em um conjunto de conceitos, seus relacionamentos e regras poderá ajudar a analisar e esclarecer até mesmo características ausentes nas mensagens expressas nos eventos.

Busca-se, através da ontologia, obter um domínio ordenado e regular dos eventos mais importantes ocorridos. Por meio desta ontologia é possível que usuários, administradores e agentes de *software* infiram conhecimentos relativos a estes eventos, evitando que condições mais adversas venham a ocorrer.

A modelagem desses conhecimentos auxilia a análise de um problema, por meio dos seus relacionamentos. Por exemplo, se a ocorrência do evento A é pré-requisito para a ocorrência do evento B, então, ao encontrar rastros do evento B, uma ferramenta poderia já indicar que se procure também solucionar o evento A.

Um dos principais problemas decorrentes do gerenciamento de eventos é o controle das aplicações de correções (*patches*). Além de registrar quais correções foram aplicadas em seus *host's*⁷, o administrador deve levar em conta quais eventos são solucionados por tais correções.

Como a ontologia efetua uma estruturação formal das informações, a partir do momento que os eventos, suas características, correções e relacionamentos são colocados na ontologia, passa a ser possível cadastrar também parte do conhecimento que, atualmente, está presente apenas como conhecimento tácito nos especialistas, que, devido à experiência, geralmente sabem quais são os requisitos para a correção de um evento, quais outros são corrigidos em conjunto, etc. Isso pode ser catalogado, recuperado e utilizado para ser usado por ferramentas de correção.

Este trabalho não é conclusivo e tem como objetivo investigar e dar os primeiros passos para a utilização de ontologias para classificação de *log* de eventos em ambientes computacionais.

⁷ *Host* é o computador que executa um programa ou serviço de servidor usado pela rede ou pelos clientes remotos. Para o balanceamento de carga de rede, um cluster consiste em vários *host's* conectados através de uma rede local (LAN). (MICROSOFT, Microsoft Computer Dictionary, Fifth Edition eBook)

Revisão de Literatura

2.1 Log de eventos

Segundo (SANTANA, 2000), eventos são ações efetuadas pelos usuários, baseadas em diretivas de auditoria, ou ações efetuadas pelo próprio sistema operacional. Através dos eventos, ficamos sabendo sobre erros, tentativas de ruptura da segurança, entre outras informações ocorridas no sistema.

O serviço⁸ *log de eventos* é iniciado automaticamente quando você inicia o sistema operacional e todos os usuários podem exibir *log's* de aplicativo e de sistema, mas apenas os administradores podem obter acesso aos *log's* de segurança.

Por padrão, o *log* de segurança é desativado. Você pode utilizar a diretiva de grupo⁹ para ativar o *log* de segurança. O administrador também pode definir diretivas de auditoria no Registro, que fazem o sistema ser interrompido quando o *log* de segurança está cheio.

Devido à complexidade das aplicações atuais, vem tornando-se prática o registro em *log* de eventos, e como resultado, fica cada vez mais difícil a sua utilização pela grande quantidade de informações disponíveis, entretanto, eles apresentam como característica comum à exigência de se conhecer, pelo menos de uma forma geral, a estrutura interna dos documentos, o que pode trazer dificuldades e respostas imprecisas, principalmente aos usuários menos habilitados.

⁸ **Serviço** é o Programa, rotina ou processo que executa uma função específica do sistema para dar suporte a outros programas, especificamente em um nível baixo (próximo ao hardware). (MICROSOFT, Microsoft Computer Dictionary, Fifth Edition eBook)

⁹ **Diretiva de grupo** é mecanismo pelo qual as configurações da área de trabalho são efetuadas automaticamente, conforme definido pelo administrador. (MICROSOFT, Microsoft Computer Dictionary, Fifth Edition eBook)

A fim de minimizar esse problema, adotou-se, a utilização de ontologias que representam o domínio do problema, com o intuito de permitir a expressão da consulta seja feita mediante o uso dos conceitos da ontologia, e não necessariamente dos elementos que compõem a estrutura interna dos documentos.

Assim, utilizaremos na modelagem da ontologia, o programa aplicativo *Event Viewer* (Visualizar Eventos), componente do sistema operacional da Microsoft Corporation que mantém *log's* sobre eventos de programa, segurança e sistema, usado para visualizar e gerenciar *log* de eventos, recolher informações sobre problemas de *hardware* e de *software* e monitorar eventos de segurança.

2.1.1 Visualizar Eventos

O programa aplicativo *Event Viewer* (Visualizar Eventos) apresentado na figura 6, têm sido de difícil manipulação por parte de usuários com pouca experiência que são obrigados a utilizar linguagens textuais ou contratar um especialista para a resolução do problema ocorrido. Como a linguagem apresentada nos *log's* frequentemente não oferece compreensão intuitiva e auxílio visual, isso evita que usuários sem experiência consigam obter vantagem total dos mesmos.

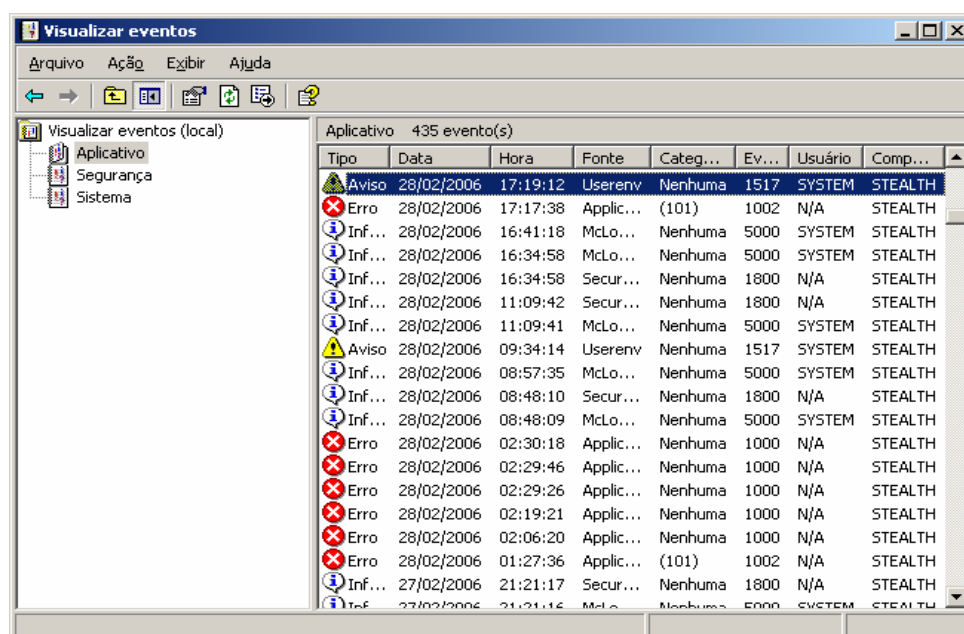


Figura 6 Tela inicial do programa aplicativo Visualizar Eventos
Fonte: MICROSOFT, Visualizar Eventos, versão 5.1.2600.2180

Os eventos são armazenados na forma de *log's*, e com eles, podemos monitorar informações sobre segurança e identificar problemas de *software*, *hardware* e sistema. Existem 3 tipos básicos de *log's*:

- **Log de sistema:** armazena os eventos registrados por componentes do Windows, por exemplo, a falha de um *driver* ou de outro componente do sistema ao ser carregado durante a inicialização é registrada no *log* do sistema. Os tipos de eventos registrados no *log* pelos componentes do sistema são determinados previamente pelo sistema operacional.
- **Log de aplicativo:** armazena os eventos registrados por aplicativos ou programas, por exemplo, um programa de banco de dados pode registrar um erro de arquivo no *log* do aplicativo;
- **Log de segurança:** registra os eventos de segurança, como tentativas de *logon* válidas e inválidas, entre outros, assim como, eventos relacionados ao uso de recursos, como criar, abrir ou excluir arquivos ou outros objetos. Um administrador pode especificar os eventos que serão registrados no *log* de segurança, por exemplo, se ativarmos a auditoria de *logon*, todas as tentativas de *logon* no sistema serão registradas no *log* de segurança.

Um computador com o Windows configurado como controlador de domínio registra eventos em dois *log's* adicionais:

- **Log do serviço de diretório:** armazena os eventos registrados pelo serviço de diretório do Windows, por exemplo, os problemas de conexão entre o servidor e o catálogo global são registrados no *log* do serviço de diretório.
- **Log do serviço de replicação de arquivos:** armazena os eventos registrados pelo serviço de replicação de arquivos do Windows, por exemplo, as falhas na replicação de arquivos e os eventos que ocorrerem enquanto os controladores de domínio estiverem sendo atualizados com informações

sobre alterações de *sysvol*¹⁰ serão registrados no *log* de replicação de arquivos.

Um computador com o Windows configurado como servidor de sistema de nomes de domínios (DNS¹¹) registra eventos em outro *log*:

- **Log do servidor DNS:** armazena os eventos registrados pelo serviço associado à resolução de nomes DNS para endereços de protocolo Internet (IP¹²) do Windows.

Com relação aos eventos de *log* de sistema, aplicativo, serviço de diretório, serviço de replicação de arquivos e servidor DNS, podemos ter 3 tipos de eventos:

- **Informação:** registro que exibe informações sobre operações bem sucedidas de um aplicativo;
- **Aviso:** registro que pode indicar um problema futuro;
- **Erro:** registro que indica problemas significativos nas operações do sistema.

Já os eventos de *log* de segurança podem ser:

- **Auditoria com êxito:** registra auditorias executadas com sucesso, como tentativa de *logon* efetuada com sucesso;
- **Auditoria sem êxito:** registra auditorias executadas sem sucesso, como tentativa de *logon* efetuada sem sucesso.

¹⁰ **sysvol** é Diretório compartilhado que armazena a cópia do servidor dos arquivos públicos de domínio, que são replicados entre todos os controladores do domínio no domínio. (MICROSOFT, Microsoft Computer Dictionary, Fifth Edition eBook)

¹¹ **DNS** ou sistema de nomes de domínios é um banco de dados distribuído de modo hierárquico que contém mapeamentos de nomes de domínios DNS para vários tipos de dados, como, por exemplo, endereços IP. (MICROSOFT, Microsoft Computer Dictionary, Fifth Edition eBook)

¹² **IP** é Protocolo que pode ser roteado no conjunto de protocolos TCP/IP responsável pelo endereçamento IP, pelo roteamento, pela fragmentação e pelo reagrupamento dos pacotes IP. (MICROSOFT, Microsoft Computer Dictionary, Fifth Edition eBook)

Os eventos do programa aplicativo *Event Viewer* (Visualizar Eventos), estejam eles, apresentando uma *informação*, um *aviso*, um *erro*, uma *auditoria com êxito* ou uma *auditoria sem êxito*, sempre conterão a mesma estrutura característica, seu *cabeçalho*, sua *descrição*, *dados opcionais* e um *link para a um site da Microsoft*, conforme apresentado na figura 7 abaixo:

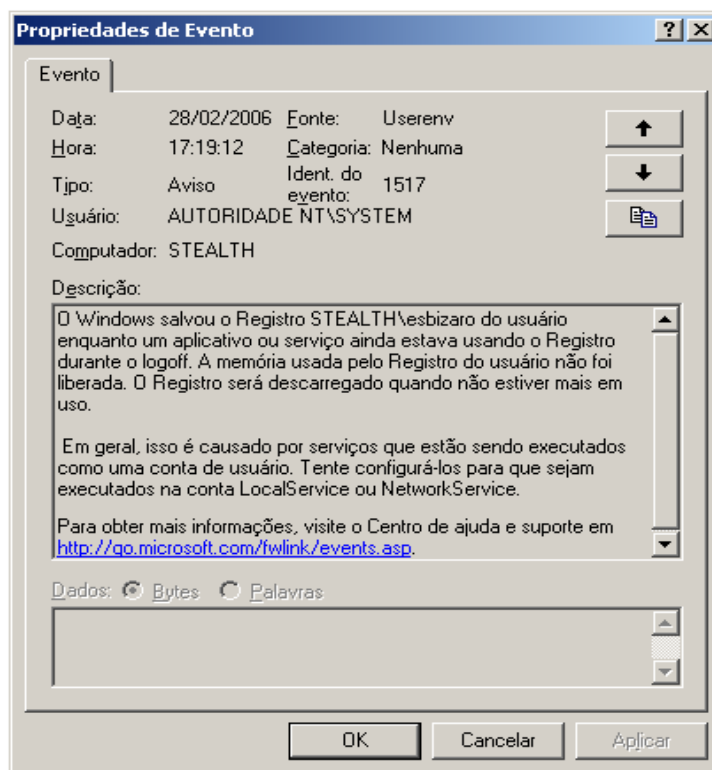


Figura 7 Propriedades do evento selecionado aleatoriamente
 Fonte: MICROSOFT, Visualizar Eventos, versão 5.1.2600.2180

O cabeçalho apresenta as seguintes informações:

- **Data:** a data na qual o evento ocorreu;
- **Hora:** a hora local na qual o evento ocorreu;
- **Usuário:** o nome de usuário em nome do qual o evento ocorreu;
- **Computador:** o nome do computador onde o evento ocorreu;
- **Identificação do Evento:** um número que identifica o tipo de evento específico;

- **Fonte:** o software que registrou o evento, que pode ser um nome de programa, como o "SQL Server," ou um componente do sistema ou de um grande programa, como um nome de *driver*;
- **Tipo:** conforme apresentados anteriormente, refere-se a uma classificação da gravidade do evento: *Erro*, *Informação*, *Aviso*, *Auditoria com êxito* ou *Auditoria sem êxito*, que são representados por um símbolo em sua exibição;
- **Categoria:** uma classificação do evento definida pela fonte do evento, usada principalmente no *log* de segurança.

O formato e o conteúdo da descrição do evento variam de acordo com o tipo de evento, mas ela é freqüentemente a parte mais útil das informações, indicando o que acontece ou o significado do evento. É nela também, que encontramos o link para um site da Microsoft, que com nossa aprovação, envia as informações sobre o evento pela Internet para a Microsoft sob a forma de uma consulta, que é usada para localizar informações mais detalhadas no site da mesma.

Segundo a empresa, dentre os dados coletados para a seqüência de consulta estão informações que descrevem o *tipo de mensagem* de evento, a *fonte* do Registro, *data e hora*, o *nome* e a *versão do produto Microsoft* e do *arquivo*, ainda segundo a mesma, os dados coletados limitam-se apenas aos itens necessários para a recuperação de mais informações sobre o evento nos artigos da base de conhecimento existente no site da Microsoft. Sendo assim, os *nomes dos usuários* e *endereços de email*, *nomes de arquivos* sem relação com o evento registrado, *endereços de máquina* e *outras formas de informações pessoalmente identificáveis* não são coletados para a seqüência de consulta.

Existe ainda um campo *dados* ou *dados opcionais*, que apresenta dados binários que podem ser exibidos em bytes ou em palavras, cujas informações geradas pelo programa que gerou o registro de evento pode ser interpretado por um técnico de suporte familiarizado com o programa de origem.

2.1.2 Sistemas Operacionais

Podemos dizer que um computador não possui nenhuma utilidade prática sem pelo menos um sistema operacional instalado. Resumidamente, ele é responsável pelo funcionamento do computador, controle dos periféricos, execução de aplicativos, gerência de memória, rede, etc.

Sistema operacional ou plataforma é um programa que gerencia o armazenamento e recuperação das informações no computador. Os programas como o Word, o Excel, o Internet Explorer, o Netscape, etc. só podem rodar, funcionar ou executar sobre um sistema operacional, ou seja, sobre uma plataforma.

2.1.2.1 Microsoft Windows

Microsoft Windows, Windows ou simplesmente Win é um sistema operacional muito popular atualmente existente, criado pela Microsoft Corporation.

O Windows é um produto comercial, com preços diferenciados para cada versão, embora haja uma enorme quantidade de cópias ilegais instaladas.

Apesar de o sistema ser conhecido pelas suas falhas críticas na segurança e como plataforma de vírus e spywares, o impacto deste sistema no mundo atual é simplesmente incalculável devido ao enorme número de cópias instaladas. Um conhecimento mínimo do sistema, do seu funcionamento, da sua história e do seu contexto é, para muitas pessoas, indispensável, mesmo leiga em informática.

A palavra *windows* em inglês significa janelas. Sua interface é baseada num padrão de janelas que exibem informações e recebem respostas dos usuários através de um teclado ou de cliques do mouse.

Este padrão de interface não foi, no entanto, criado pela Microsoft, e as versões do Windows estão preocupadas em atender essas necessidades no mercado empresarial e doméstico, e suas características e falhas devem ser vistas neste contexto.

O Windows só começa a ser tecnicamente considerado como um sistema operacional a partir da versão Windows 95 lançada em Agosto de 1995, já que antes não existiam sistemas gráficos sendo executados sobre alguma versão dos sistemas DOS, como MS-DOS, PC-DOS ou DR-DOS, que é um sistema operativo que não dispõe de interface gráfica, funciona através de comandos de texto introduzidos no teclado pelo usuário.

A partir do MS-DOS, diversas versões do Windows surgiram, Windows 1.0, 2.0, 3.x, 95, NT, 98, ME, 2000, XP, 2003 e atualmente Vista. Mas das versões existentes, somente são foco desde trabalho as versões 2000, XP e 2003, por apresentarem por padrão os registros de *log* de eventos em seus sistemas.

2.1.2.1.1 Microsoft Windows XP

O Windows XP é a versão do Windows que funciona em um sistema de arquivos formatado com FAT32 (File Allocation Table - tabela de alocação de arquivos) ou NTFS(New Tecnology File Sistem - nova tecnologia de sistema de arquivos), que garante maior segurança e integridade ao sistema operacional. A sigla XP deriva da palavra eXPeriência.

Versões: Professional, Home, Tablet PC, Embedded e Starter Edition

2.1.2.1.2 Microsoft Windows 2000

O lançamento desse Windows marcou o começo da era NT (*New Tecnology*) ainda com defeitos de segurança, como por exemplo, as senhas serem salvas em um arquivo próprio e visível o que facilitava a ação de Hackers e invasores, ainda não apresentava muita semelhança com o XP no visual que até então ainda era o mesmo do ME, mas, com uma pequeníssima diferença que dava um charme: o mouse já possuía sombra, porém foi um grande passo para a popularidade dos NT's.

Versões: Professional, Server, Advanced Server e Datacenter Server.

2.1.2.1.3 Microsoft Windows 2003

Lançado pela Microsoft em 24 de abril de 2003, o Windows Server 2003 é um sistema operacional de rede desenvolvido como sucessor do Windows 2000 Server. Em seu núcleo está uma versão do Windows XP com algumas funções desligadas para permitir um funcionamento mais estável do sistema.

Assim como o Windows 2000, este apresenta o Active Directory¹³ como principal ferramenta para a administração de uma rede. É um sistema utilizado extritamente em redes de computadores.

O Windows Server 2003 da Microsoft trouxe novas melhorias aos serviços de rede e ao Active Directory, que agora implementa mais funcionalidades em relação ao Windows 2000 Server.

Versões: Standard Edition, Enterprise Edition, Data Center Edition e *Web Edition* (32 e 64 bits).

2.1.3 Site *Events and Errors Message Center*

O objetivo deste texto é conduzir o usuário a uma análise dos recursos apresentados, sobre como realizar uma pesquisa gratuita na base de dados do site oficial *Events and Errors Message Center*, de abrangência internacional, mantida pela Microsoft Corporation.

O usuário poderá ter acesso ao site por meio do endereço http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx onde se observa que inúmeras opções são apresentadas ao usuário e, vemos a possibilidade de dois tipos de pesquisas: a Busca Básica (*Basic Search*) e a Busca Avançada (*Advanced Search*).

¹³ O serviço de diretório que armazena informações sobre objetos em uma rede e disponibiliza essas informações a usuários e administradores de rede. (MICROSOFT, Microsoft Computer Dictionary, Fifth Edition eBook)

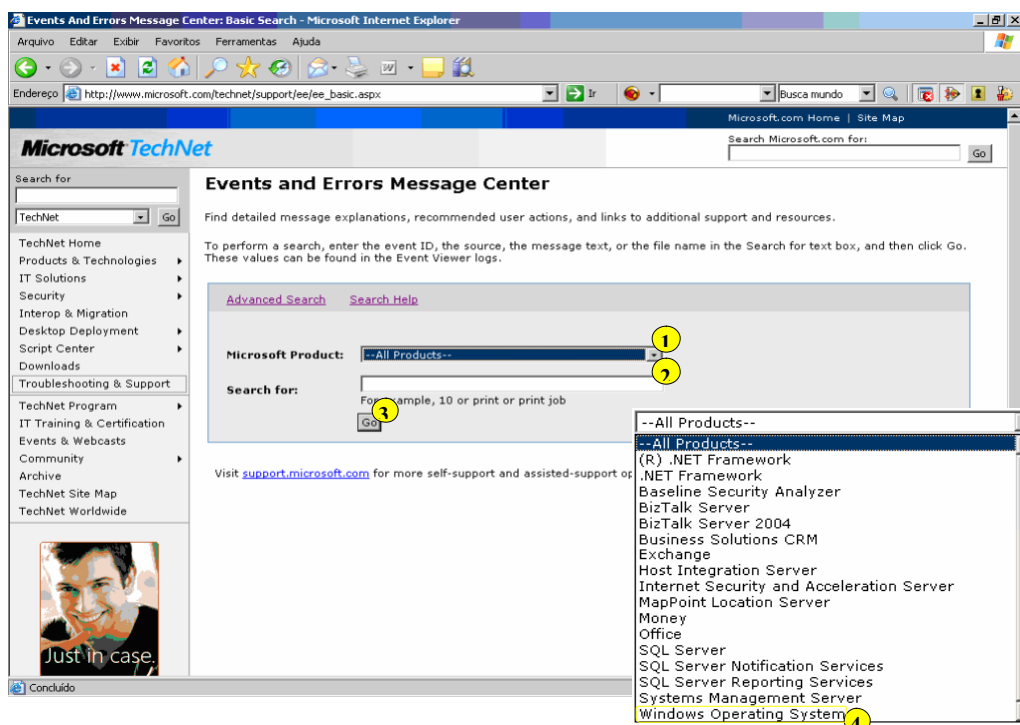


Figura 8 Tela inicial da “Busca Básica” (Basic Search)
Fonte: Microsoft Corporation, 2006.

Na busca básica apresentada na figura 8 acima, será procurada informações em todas as versões do produto selecionado, e temos as seguintes opções:

- 1) **Microsoft Product:** onde podemos selecionar o produto específico da Microsoft que desejamos procurar, ou utilizar a opção “todos os produtos” (*--All products--*) é útil quando não se sabe ao certo qual a aplicação que está causando o problema ou qual sistema operacional que você está usando. Se várias aplicações estão sendo executadas, o evento ou o erro pode não deixar claro qual produto esta associado a ele. Por exemplo, o usuário do Outlook recebe uma mensagem de erro, este erro que parece ser relacionado ao Outlook, pode na verdade estar sendo causado por um problema no sistema operacional ou pelo Exchange.
- 2) **Search for:** onde podemos fornecer um ID do evento (*Event ID*), uma fonte do evento (*Event Source*), um nome de arquivo (*File Name*), alguma mensagem de texto (*Message Text*), ou uma combinação destes. O número máximo de caracteres permitidos é de 225 caracteres.

- 3) **Botão Go:** após informar os dados acima, basta clicar no botão Go ou pressionar a tecla enter para enviar a consulta.

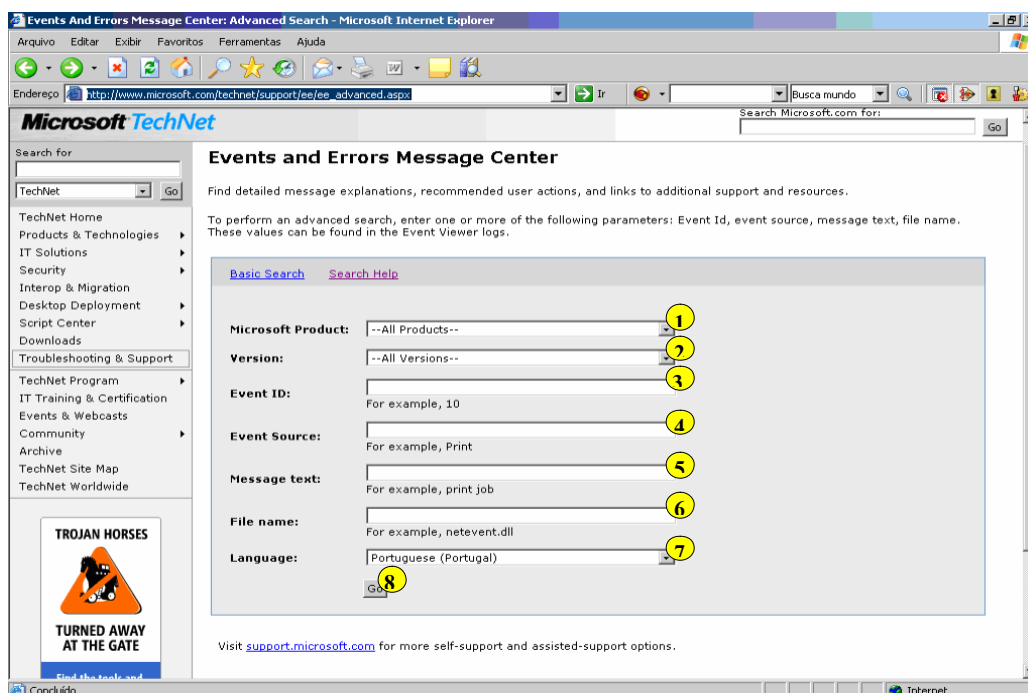


Figura 9 Home Page do site *Events and Errors Message Center*
Fonte: Microsoft Corporation, 2006

E ainda, sugere utilizarmos à busca avançada (*Advanced Search*), apresentada na figura 9, que apresenta-se como tela inicial e fornece melhores opções para um maior controle dos resultados obtidos, dependendo dos critérios de busca:

- 1) **Microsoft Product:** conforme citado na busca básica;
- 2) **Version:** Primeiramente, devemos selecionar um produto de Microsoft, ou aceitar a opção “todos os produtos” (*--Allproducts--*). A lista da versão é povoada somente depois que um produto de Microsoft é selecionado e também podemos utilizar a opção “todas as versões” (*--Allversions--*) quando não for capaz de encontrar alguma informação para uma versão específica de um produto da Microsoft;
- 3) **Evento Id:** O evento ID pode ser encontrado no registro do visor do evento, e é um número que identifica o tipo de evento específico;

- 4) **Event Source:** A fonte do evento pode ser encontrada no registro do visor do evento, e é o *software* ou um componente do sistema ou de um grande programa que registrou o evento;
- 5) **Message Text:** O texto da mensagem pode ser encontrado nos registros do visor do evento. Você pode inserir até 225 caracteres na caixa de texto da mensagem e os resultados da busca conterão todas as palavras incorporadas a esta caixa de texto, por exemplo, se você inserir o serviço que falhou na caixa de texto da mensagem, os resultados da busca serão as páginas que contêm o serviço e a falha;
- 6) **File Name:** O nome do arquivo pode ser encontrado no registro do visor do evento;
- 7) **Language:** Selecione a língua apropriada que você quer usar para filtrar seus resultados ou aceite a seleção padrão;
- 8) **Botão Go:** conforme citado na busca básica.

Da lista de produtos apresentada, neste trabalho sempre e somente utilizaremos apenas a opção *Windows Operation System*, relativa aos sistemas operacionais da Microsoft.

Já em relação à versão do produto apresentado, neste trabalho sempre e somente utilizaremos apenas a opção 5.2 (*Windows Server 2003*), relativa ao sistema operacional Microsoft Windows Server 2003 em todas as suas edições.

E, em relação à língua a ser pesquisada, neste trabalho sempre e somente utilizaremos apenas a opção *english*, pois em outras línguas não obtivemos resultados nas pesquisas efetuadas.

Para resultados melhores: na ajuda do site nos é informado que para termos melhores resultados em nossas buscas, devemos tentar incorporar mais palavras à

consulta, por exemplo, utilizando a busca por *serviço falhou*, ao invés de simplesmente *serviço*.

Para buscas em frases: Para efetuarmos buscas com frases inteiras, devemos colocar aspas ao redor das palavras que desejamos buscar.

Se nenhuma das caixas de texto contiver valores, a busca não retornará nenhum resultado.

Sejam pesquisas efetuadas pela busca básica (*Basic Search*) ou pela busca avançada (*Advanced Search*), as mesmas sempre retornaram seus dados seguindo a estrutura apresentada na figura 10 abaixo.

Os elementos *Product* ou *Microsoft Product*, *Event ID*, *Source* ou *Event Source*, *Version* já foram descritos anteriormente, mas com relação ao elemento *Symbolic Name*, não foram encontradas descrições para o mesmo, por isso, ele foi ignorado neste trabalho.

Details	
Product:	Windows Operating System
Event ID:	1517
Source:	Userenv
Version:	5.2
Symbolic Name:	EVENT_HIVE_SAVED
Message:	Windows saved user %1 registry while an application or service was still using the registry during log off. The memory used by the user's registry has not been freed. The registry will be unloaded when it is no longer in use. This is often caused by services running as a user account, try configuring the services to run in either the LocalService or NetworkService account.
Explanation	
Windows unloads each user's profile and user's section of the registry when the user logs off. This message indicates that Windows could not unload the user's profile because a program was referencing the user's section of the registry. This locked the profile. The registry cannot unload profiles that are locked and in use. When the program that is locking the profile is no longer referencing the registry, the profile will be unloaded.	
User Action	
No user action is required.	
<input type="checkbox"/> Related Knowledge Base articles	
You can find additional information on this topic in the following Microsoft Knowledge Base articles:	
<ul style="list-style-type: none"> • Warning error 1517 in Event Viewer after first restart 	
Explains the presence of an event log warning that appears when you first	

**Figura 10 Resultado obtido na busca pelo site (Language: English).
Fonte: Microsoft Corporation, 2006.**

O elemento *Message* retornado, refere-se a *Descrição* apresentada no programa aplicativo *Event Viewer* (Visualizar Eventos), mas com caracteres identificadores (%1) que são substituídos no momento de gravação no programa aplicativo.

Já os elementos *Explanation*, *User Action* e *Related Knowledge Base articles* apresentados, indicam uma possível explicação e ação a ser tomada por parte do usuário, onde, o último será também ignorado nessa pesquisa, por sair do lócus de trabalho da mesma.

As informações acima apresentadas nos mostram que a monitoria cuidadosa dos *log* de eventos pode ajudar a prever e identificar as fontes dos problemas, bem como, confirmar problemas com hardware e software.

Assim sendo, vemos que nosso trabalho, com um cuidadoso e metodológico estudo apoiado em uma ontologia possivelmente servirá de suporte para entender e resolver os numerosos eventos que podem ocorrer no sistema operacional em uso.

2.2 Recuperação de Informação

Segundo (Rijsbergen, 1979), desde os anos 40 o problema do armazenamento e recuperação de informações tem atraído grande atenção, declarando que: “nós temos grande quantidade de informação cujo acesso rápido e preciso torna-se cada vez mais difícil”.

Vemos que, com o aumento ininterrupto da quantidade de informação disponível e com a escassez de tempo para procurá-la, o processo de recuperação de informações que aprimora cada vez mais as suas técnicas, torna-se muito importante nas suas tentativas de aperfeiçoar e facilitar a busca aos usuários, objetivando cada vez mais, obter melhores resultados.

Lancaster (1993) *apud* Feitosa (2005) apresenta que para qualquer necessidade específica de informação, haverá sempre muito mais itens que não possuem relevância

ou pertinência ao assunto pesquisado pelo consulente (itens inúteis) do que itens que possuam relevância ou pertinência às necessidades desse usuário (itens úteis), apresentando ainda, que a principal função de um sistema de recuperação de informações é permitir que o usuário localize o maior número possível de itens relevantes.

Neste trabalho, *relevância, pertinência, precisão e revocação* não são estudados, pelo fato de buscarmos na ontologia um apoio para a recuperação de informações, que traria um entendimento subjetivo ao usuário, e que não possibilitaria a aplicação de fórmulas para a representação dos coeficientes acima citados.

Dentre os principais problemas na recuperação de informação esta o fato de se obter a representação adequada de cada documento, que pode ser, por exemplo, uma lista de palavras extraídas consideradas relevantes, e também a forma correta de consulta para utilização no ambiente computacional. (Rijsbergen, 1979) apresenta um sistema típico de recuperação de informação, conforme figura 11 abaixo:

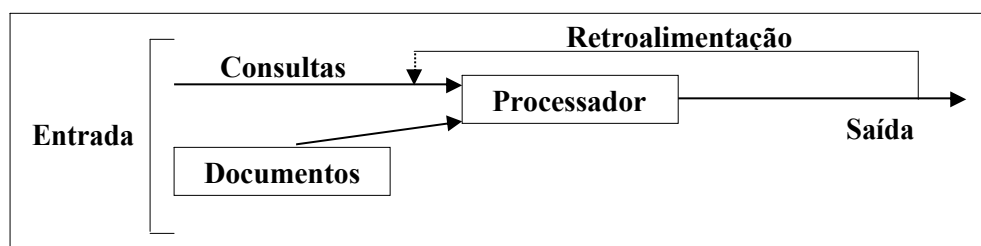


Figura 11 Um sistema de recuperação de informações típico
Fonte: Rijsbergen, (1979, p. 4)

Vemos na figura 11 que, na *entrada* temos, tanto os *documentos* que estarão representados e estruturados de alguma forma, como as *consultas* feitas pelos usuários do sistema em um determinado *processador*, que efetua a busca da informação solicitada e apresenta uma determinada *saída* como resultado do processo, que ainda pode ser utilizada como retro-alimentação para a realização de uma nova busca baseada nos resultados obtidos.

Como atualmente já não se pode falar em crescimento do volume de publicações, mas em uma verdadeira explosão. As bibliotecas digitais, que são as publicações armazenadas e manipuladas eletronicamente, aparecem como um

paradigma para esse sistema de representação de informações típico, na tentativa de melhorar a busca e a apresentação de informações desejadas.

Conforme Kim e Seo (2002), tradicionalmente os sistemas de recuperação de informação estão focados na busca e elaboração de um *ranking* de documentos em resposta a uma pergunta do usuário. Mas, o problema da recuperação de informação se acentua à medida que os usuários requerem um acesso mais rápido a informação relevante, mediante suas consultas que ainda estão sendo feitas por meio de palavras-chave, em bancos de dados contendo grandes quantidades de informação que estão sendo cada vez mais difíceis de serem acessadas.

A crescente complexidade dos objetos armazenados e o grande volume de dados exigem processos de recuperação cada vez mais sofisticados e, diante deste quadro, a recuperação de informação apresenta a cada dia novos desafios e se configura como uma área de significância cada vez maior.

Para Gey (1992), um sistema de recuperação de informação pode ser estruturado conforme a Figura 12 abaixo:

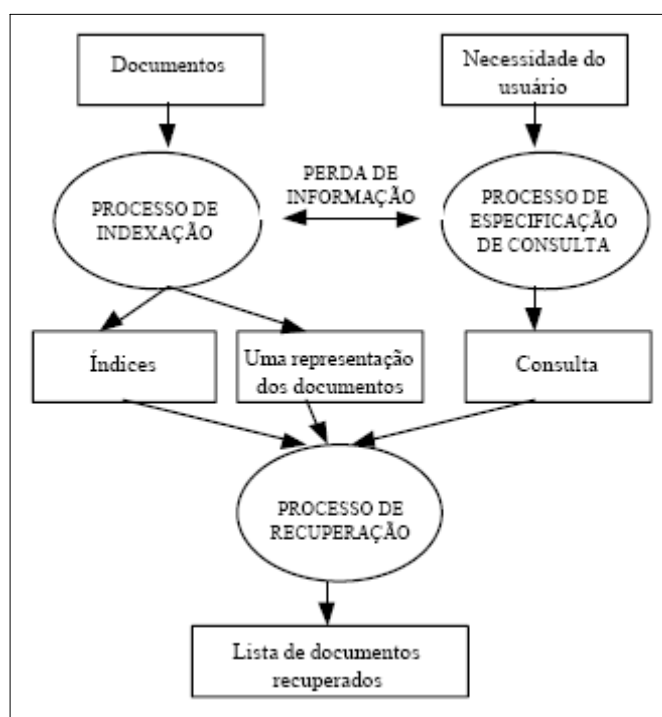


Figura 12 Componentes de um sistema de recuperação de informação.
Fonte: Gey, 1992

Os componentes do sistema incluem os documentos existentes, e seu processo de indexação, criando índices e outras formas de representação dos documentos, e as necessidades do usuário, formulando suas consultas especificam ao processo de recuperação de informações, que a partir das representações e estruturas de dados, recupera uma lista de documentos considerados relevantes.

Problemas, como a *sinonímia*¹⁴, apresentam-se quando tentamos fazer uma recuperação de informações apenas por palavras-chaves, por exemplo, no contexto deste trabalho, se alguém estiver procurando em uma consulta “Áudio do Windows Server 2003” em um sistema convencional de recuperação de informações, só os documentos em que a palavra “Áudio” estiver presente serão recuperados, mas naqueles documentos em que a informação esteja referenciada como “som”, “*sound*”, “volume”, “auto-falante”, “*loudspeaker*” não serão. Já em um sistema de recuperação de informações semânticas apoiado em ontologias, temos a possibilidade de se apresentar termos, tanto em outras línguas, quanto sinônimos, que resolveriam esse tipo de problema.

Mauldin (1991) explana que para se ultrapassar o limite de desempenho causado pelas recuperações de informação baseada em palavras-chave, é preciso fazer mais do que um relacionamento de palavras, um relacionamento conceitual ou de significado, e para esse relacionamento conceitual três aspectos devem ser levadas em consideração:

- Uma linguagem conceitual para representar significados;
- Um analisador sintático para converter sentenças em linguagem natural à linguagem conceitual, e;
- Um relacionador que determina se uma sentença na linguagem conceitual é a mesma ou uma instância de uma segunda sentença em linguagem conceitual.

Baeza-Yates e Ribeiro-Neto (1999) nos apresentam a recuperação de informação como a representação, armazenamento, organização e acesso a informação, onde, a

¹⁴ **Sinonímia** é a relação que se estabelece entre duas palavras ou mais que apresentam significados iguais ou semelhantes - SINÔNIMOS. Ex.: Cômico – engraçado, Débil - fraco, frágil, Distante - afastado, remoto. (<http://www.portugues.com.br/semantica/sinonimia.asp>)

organização dos elementos deve prover aos usuários um acesso fácil e rápido à informação que eles precisam.

Neste projeto a representação, armazenamento e organização da informação sobre *log* de eventos serão feitos em uma ontologia e, retomando a figura 11, teremos como entrada, tanto as consultas dos usuários como a base de conhecimento criada sobre *log* de eventos, que serão processados por um mecanismo de busca apoiado na ontologia, apresentando então os resultados obtidos, que ainda, pela mesma ontologia, serão esclarecidos na tentativa de facilitar o entendimento e a resolução dos mesmos por parte do usuário.

2.2.1 Mecanismos de busca

A Internet é um imenso repositório mundial de informações, que possui um potencial que permite a pessoas de diferentes culturas, com conhecimentos e interesses os mais variados, compartilhar informações tornando-as disponíveis publicamente e, ao mesmo tempo, procurar por conhecimentos e experiências já desenvolvidas e relatadas.

Para Alencar (2001), logo após a chegada da *Web* os estudos sobre mecanismos de busca tornaram-se cada vez mais extensos e amplos, mas todos eles mantendo em sua estrutura três subsistemas distintos, que se relacionam entre si:

- **Subsistema de aquisição/coleta de documentos:** está relacionado com a forma com a qual o mecanismo de busca obtém páginas para sua base de dados;
- **Subsistema indexador dos documentos:** diz respeito à forma como o mecanismo de busca armazena as informações que coleta, seja através da indexação em texto completo ou de representações das páginas (metadados);

- **Subsistema de ordenação dos resultados:** está relacionado com o algoritmo proprietário de cada mecanismo de busca e que, em geral, é o maior diferencial entre os serviços.

As informações disponíveis na Internet se apresentam em diversos formatos. Os recursos podem estar disponíveis em forma de textos, arquivos formatados, imagens, sons, vídeos. No entanto, a *World Wide Web*, mais comumente chamada *Web*, oferece uma interface que permite a disponibilização da maioria dos recursos disponíveis.

Independente do tamanho estimado, a Internet pode ser vista como uma grande massa de informações. Para localizar a informação desejada é necessário o desenvolvimento de ferramentas de busca cada vez mais poderosas que permitam uma melhor filtragem do material desejado.

Falar de recuperação de informação na Internet e WWW significa falar de mecanismos de busca, aqui entendidos como aqueles que estão diretamente relacionados à busca de informação no ambiente da *Web* e não na Internet como um todo.

Os mecanismos de busca e as meta ferramentas indexam a parte pública da *Web*. Uma das críticas a esses serviços é que as tecnologias que adotam não permitem o aprofundamento e a inclusão da parte “invisível” da *Web* (BERGMAN, 2001).

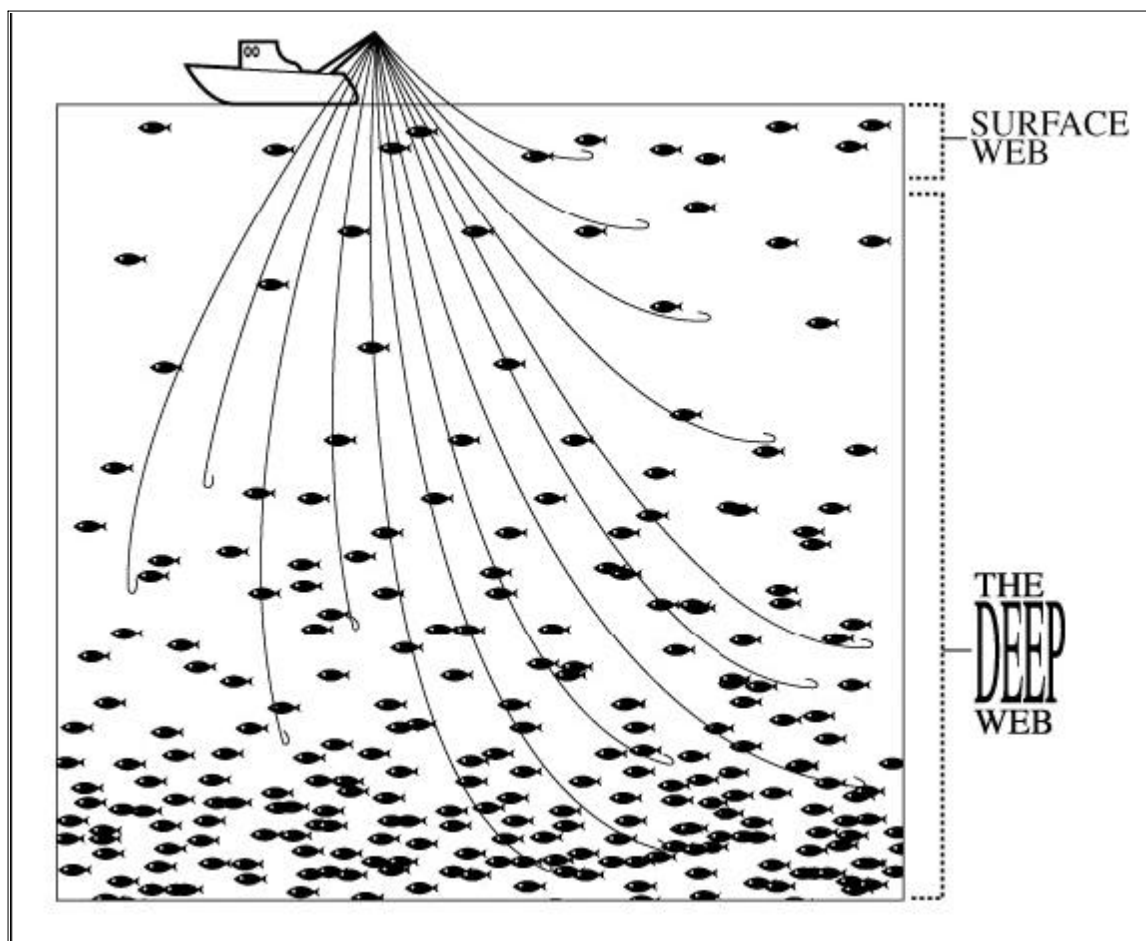


Figura 13 A coleta de dados na *Web*: a superfície e a *Web* profunda

Fonte: Bergman, M. K., 2001

Como as ferramentas disponíveis são incapazes de recuperar informações contidas num extenso e valioso conjunto de informações que não são acessíveis através dos mecanismos de busca, por exemplo, as informações que podem estar contidas em sites que necessitam de um processo de registro para entrar (*login* e senha), como jornais, periódicos, etc., ou ainda, em sites cujas páginas são geradas dinamicamente a partir de uma solicitação a um banco de dados, como é o caso do site oficial *Events and Errors Message Center* da empresa Microsoft Corporation, foco deste trabalho, que contém coleções de dados estatísticos que não são capturados pelos robôs que trabalham nos mecanismos de busca.

2.3 Terminologia

A Terminologia é considerada uma ciência ainda nova; mas, em contrapartida, o uso de termos técnico-científicos vem de tempos remotos:

A terminologia não é um fenômeno recente. Com efeito, tão longe quanto se remonte na história do homem, desde que se manifesta a linguagem, nos encontramos em presença de línguas de especialidade, é assim que se encontra a terminologia dos filósofos gregos, a língua de negócios dos comerciantes cretas, dos vocábulos especializados da arte militar, etc. (Rondeau, 1984)

Segundo Eugênio Wüster (1998, p.26), fundador da teoria terminológica, ela é um campo interdisciplinar, no qual a lingüística é um dos pólos de convergência, ao lado da Lógica, da Ontologia, da Ciência da Informação e das diversas “áreas do saber”.

Segundo KRIEGER (2000), para Maria Teresa Cabré, a teoria wüsteriana:

estabelece um objeto de análise e umas funções de trabalho muito restritivas, porquanto limita o objeto às unidades unívocas normalizadoras próprias dos âmbitos científicos-técnicos, reduz a atividade terminológica à recolha de conceitos e termos para a normalização (fixação de noções e denominações normatizadas) dos termos, circunscreve os âmbitos especializados à ciência e à técnica e limita seus objetivos para assegurar a univocidade da comunicação profissional, fundamentalmente no plano internacional. (1998, p.2)

As proposições inovadoras da terminologia se estabelecem à luz do princípio de que o conhecimento só pode ser produzido e apreendido mediante sua materialização nos diferentes sistemas semióticos, dentre os quais o verbal é predominante no universo das ciências e tecnologias. Com isso, os propósitos que competem a uma teoria da terminologia passam a ser assim compreendidos:

A terminologia trata de conceitos e, portanto, de estruturas de conhecimento apenas na medida em que estão representados no léxico da língua. Os conceitos são elementos da estrutura do conhecimento e, como tais, ocupam um lugar importante dentro da filosofia das ciências e das teorias cognitivas. (Sager, 1993, p.36)

A terminologia tendo um papel importante nas particularidades existentes nas línguas de especialidade, motivadas pela existência de uma comunidade lingüística, nos

apresenta que a aprendizagem de uma determinada língua é facultativa aos indivíduos, mas, a aceitação do indivíduo pelo grupo pode depender dessa aprendizagem.

São características das línguas de especialidade profissionais: *possuírem íntima relação e existência dependentes do grupo correspondente; obrigatoriedade para profissionais de uma área e para mediadores de textos profissionais, como tradutores, jornalistas especializados etc; função de comunicação; serem distintas da língua geral nos níveis morfológicos, sintático e lexicológico, bem como, em relação ao aspecto de ensino-aprendizagem; são línguas normalizadas*; o que deixa claro o importante papel da Terminologia nesse contexto.

Guardando estreita relação com a Ciência da Informação, a Terminologia também tem preocupações com a elaboração, organização, armazenamento, recuperação e representação do conhecimento, no sentido de otimizar essas operações.

Segundo Cabré (1993), a Terminologia traz sua colaboração à Informática a partir da elaboração dos conceitos que utilizam os elementos teóricos por ela proporcionados, criando com ela ferramentas que resultem em grande utilidade, sobretudo para a Terminologia Aplicada, trazendo assim, benefícios nas áreas de recuperação de informação, inteligência artificial e sistemas especialistas.

Para Felber (1984) Terminologias são importantes para a organização do conhecimento nas áreas de relacionamento dos conceitos, lembrando que, em todos os idiomas, incontáveis novos conceitos surgem diariamente em todos os campos da atividade humana e devem ser expressos por termos existentes ou por novos termos, formados pela combinação de elementos das palavras, como raízes e afixos.

2.3.1 Língua de especialidade

O conceito denominado por língua de especialidade tem sido definido como um conjunto limitado de características de uma determinada língua natural, com o objetivo

de garantir a comunicação utilizando-se de palavras ou expressões como meio de representação dos conceitos.

Segundo Wright; Budin (1997 *apud* Feitosa, 2005), os lingüistas e os terminólogos geralmente utilizam os termos *língua de especialidade*, *linguagem especial*, *linguagem especializada* ou *linguagem para propósitos especiais*, para designarem subsistemas léxicos da língua geral.

A Terminologia, preocupa-se com os termos, isto é, unidades lexicais atribuídas a conceitos utilizados em línguas de especialidade, que ocorrem em um dado campo ou domínio do conhecimento. As unidades terminológicas que são extraídas dessas línguas de especialidade geralmente representam um número limitado de partes do universo do discurso: nomes, verbos, adjetivos, advérbios, bem como expressões fraseológicas, que combinados com conectivos na língua geral produzem um discurso especializado coerente.

2.3.2 Unidade terminológica

Atualmente, a Terminologia fundamenta-se em um modelo tripartite, baseado em três pontos chaves: o objeto, o conceito e o termo (signo). A figura 14 abaixo apresenta uma adaptação do triângulo do significado de Ogden e Richards (1960) caracterizando a relação existente entre objeto, conceito e signo.

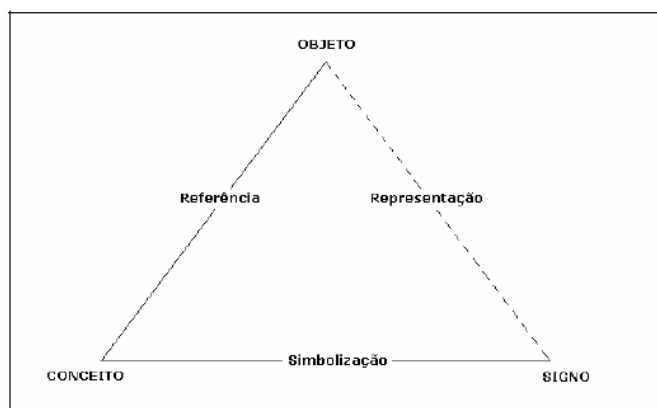


Figura 14 Adaptação do triângulo do significado de Ogden e Richards
Fonte: Brandão, 2005

A ligação tracejada entre objeto e signo evidencia o fato de que tal relação só pode ser construída a partir de uma ponte conceitual. Partindo da figura 14 é possível afirmar que os objetos (também conhecidos como referentes) são referenciados por conceitos (também conhecidos como referências ou significados) simbolicamente representados por signos. De maneira indireta, os signos representam, aleatoriamente e não exclusivamente, os objetos. Aleatoriamente pelo fato de não haver relação causal entre o objeto e o símbolo escolhido para sua representação. Não exclusivamente pelo fato do mesmo signo poder representar objetos diferentes. Nesse caso, a identificação do objeto de referência não pode prescindir de sua inserção no contexto.

Para Dahlberg (1978) os conceitos são constituídos de características que traduzem os atributos dos objetos designados. A decomposição do conceito para identificação de tais características só é possível através da coleta das proposições verdadeiras sobre o objeto designado e da extração - a partir das proposições - de atributos predicáveis do objeto. A autora afirma que os elementos do conceito são obtidos a partir do método analítico-sintético.

2.3.2.1 Conceito

De acordo com a lógica estóica (CHAUI, 1995), a possibilidade de existência do conhecimento está condicionada à existência de três elementos: o objeto individual, sua representação mental e sua representação simbólica, sendo através da representação

mental que se torna possível estabelecer a relação entre o objeto individual e sua representação simbólica.

Reconhecendo a importância destes três elementos, Dahlberg (1978) formulou a sua teoria do conceito, com que pretendia revelar os mecanismos de construção de um conceito, seus elementos constituintes, as possíveis relações existentes entre eles e suas categorias gerais.

Para Dahlberg (1978) os conceitos são representações mentais de um objeto individual (presente “aqui e agora”) ou geral (independente de tempo e espaço). Dessa maneira são conceitos individuais aqueles que são postos no lugar de objetos individuais e conceitos gerais aqueles que são postos no lugar de objetos gerais do pensamento. Os conceitos individuais são compreendidos pelos conceitos gerais podendo, dessa forma, serem reduzidos a eles.

O conceito é constituído de elementos que se articulam numa unidade estruturada: os enunciados ou proposições. A partir da reunião e compilação de proposições verdadeiras a respeito de determinado objeto formam-se os conceitos. O signo é o elemento de representação simbólica do objeto. Ele traduz e fixa essa compilação de proposições verdadeiras. Dessa maneira, Dahlberg (1978) define o conceito como “a compilação de enunciados verdadeiros sobre determinado objeto, fixada por um símbolo linguístico”.

Um conceito é uma unidade do pensamento, geralmente expressa por um termo, um símbolo literal ou qualquer outro símbolo. Quando retrata um único objeto é chamado conceito individual e geralmente é representado por um nome próprio ou por um símbolo, quando representa dois ou mais objetos, é chamado de conceito geral e a designação é representada por um termo.

Conceitos são representações mentais de objetos individuais ou, por abstração, de um conjunto de objetos individuais que possuem qualidades (ou características) em comum. Assim sendo, um conceito, que é acima de tudo um elemento do pensamento que existe apenas na mente humana necessita de um símbolo lingüístico (termo, letra, símbolo gráfico) para ser expresso ou comunicado.

Podem representar não apenas seres ou coisas, mas também, qualidades (expressas por nomes), ações (expressas por verbos ou nomes) e até localizações, situações ou relações (expressas por advérbios, preposições, conjunções ou nomes).

2.3.2.1.1 Características

Para Felber (1984 *apud* Feitosa, 2005), uma característica é um elemento de um conceito que serve para descrever ou identificar certa qualidade de um objeto individual, que é também por si só, um conceito. São utilizadas para comparar conceitos, classificar conceitos, para fórmular a definição de um conceito, bem como para formar os termos atribuídos aos conceitos.

As características podem ser intrínsecas, inerentes aos objetos, como *forma*, *tamanho*, *material*, *cor* etc, ou extrínsecas de propósito e de origem, a primeira que designam o propósito dos objetos, como *aplicação*, *funcionamento*, *performance*, *localização* etc e a segunda, que designam a origem do objeto, como *inventor*, *produtor*, *país de origem*, *fornecedor* etc.

Dahlberg (1978) analisa as características de um conceito, denominada análise conceitual, apresentando que os mesmos, na sua dimensão extensional, podem assumir o caráter de verdadeiras classes de objetos, desde que as características dos conceitos de tais objetos, por meio de um ou mais conceitos, estejam contidas nos conceitos de um sistema de conceitos.

2.3.2.2 Termo

Felber (1984) afirma que um termo é qualquer símbolo convencional utilizado para representar um conceito definido em um campo de assunto. Sager (1990) corrobora essa afirmação, argumentando que os termos são representações lingüísticas de conceitos e que, diferentemente da língua geral, em que a arbitrariedade do signo é

aceita, as línguas de especialidade tentam sistematizar princípios de designação e nomeação de conceitos, de acordo com regras pré-estabelecidas ou princípios gerais.

Cabré (1993) ressalta que, como qualquer outra unidade significativa do sistema lingüístico, os termos fazem parte de um sistema estruturado, no qual ocupam um determinado nível e se relacionam, por um lado, com as demais unidades do mesmo nível e, por outro, com as unidades dos demais níveis, participando conjuntamente na construção do discurso.

Para Rondeau (1984, p.19), o termo é essencialmente um signo lingüístico, isto é, uma unidade lingüística que compreende um significante e um significado. O autor adota a nomenclatura de *denominação* (termo) para o significante e de *noção* (conceito) para o significado. Uma característica do termo é o fato de que o seu significado é definido em relação ao conjunto de conceitos pertencentes ao mesmo domínio. Em suma, um termo não pode ser considerado isoladamente, pois apresenta-se como uma representação de um objeto em um conjunto semântico que pode ser uma disciplina, uma ciência, um domínio de atividades, uma técnica, entre outras áreas, todas pertinentes a um domínio especializado.

Os termos devem ser tão concisos quanto possível, sem, contudo, perderem a clareza do seu significado. Termos abreviados podem ser formados a partir de um termo simples, de um termo composto, ou de uma frase, repetindo sílabas ou letras isoladas do termo de origem, mas devem ser evitados se o termo completo for suficientemente conciso.

Termos que possuem mais de um significado, plurivalentes, podem ser homônimos ou polissêmicos em relação a outros termos, e esse fenômeno faz com que aumente a ambigüidade em relação aos seus significados. Quando o fenômeno da ambigüidade ocorre, é necessário que se utilizem termos diferentes para a explicação dos diferentes significados de um termo ambíguo.

A monossêmia é a relação entre termos e conceitos, na qual um termo representa apenas um conceito. A mononímia é a relação entre termos e conceitos em que um conceito possui apenas um termo que o designa em uma determinada língua. A

homonímia é caracterizada por envolver a relação entre termos e conceitos, na qual os termos em uma determinada língua apresentam formas idênticas, sejam elas fonéticas ou gráficas, mas que designam conceitos diferentes e não-relacionados. Termos que possuem fonia idêntica, mas com grafia distinta são ditos homófonos. Termos de grafia idêntica, mas com pronúncia distinta são ditos homógrafos. Homônimos perfeitos possuem idênticas grafias e pronúncia. A sinonímia é a relação entre diferentes termos que designam o mesmo conceito em uma determinada língua. Se termos são atribuídos a conceitos cujas intensões são quase idênticas, são chamados de quase-sinônimos (ISO WD 704.1, 1996 *apud* Feitosa, 2005).

Os aspectos relativos à Teoria do Conceito, estudados no âmbito da Terminologia, são relevantes para os estudos de Representação do Conhecimento e para a área de Organização e Recuperação de Informação.

Os conteúdos de livros e documentos digitais ou não, seus conceitos representando uma necessidade básica de comunicação, são compostos de objetos, pensamentos e idéias presentes no mundo cognoscível e imaginário da humanidade, que podem se referir as coisas concretas ou abstratas, registrando o conhecimento humano:

“Um conceito é uma rede de padrões de inferências, associações e relacionamentos que são predicados ou ditos de outra forma trazidos em cena através do ato da categorização"[...] a cristalização ou formalização do pensamento inferencial, nascida da percepção sensorial, condicionada pela operação do cérebro humano e delineada pela experiência humana. Ela repousa na fundamentação de todo pensamento mas ela é pragmática e instrumental. É permanente e efêmera. Permanente porque sem ela, a cognição é impossível; efêmera porque ela pode ser rejeitada quando sua utilidade é esgotada" (SHERA, 1957 *apud* ALVARENGA, 2001).

Atualmente, com a Internet, temos conteúdos cada vez mais elaborados, textos, sons, imagens, utilizando-se das linguagens disponíveis, que ultrapassaram as paredes de empresas, instituições, universidades e bibliotecas para um meio on-line, através dos serviços da grande rede de comunicação, que ao invés de serem acessados fisicamente, passam a ser processados eletronicamente, exigindo maior entendimento semântico obtido diretamente no ciberespaço.

Em nossos dias, constata-se uma grande evolução nas pesquisas relacionadas ao estudo do conceito, aqui abordado sucintamente, principalmente na Biblioteconomia e

Ciência da Informação, que desde seus primórdios contou com pesquisadores levantando questões específicas, que utilizavam métodos e princípios voltados aos processos de organização e recuperação de informações.

2.4 Representação do conhecimento

A representação do conhecimento foi considerada por muito tempo um dos elementos principais da Inteligência Artificial, e uma parte crítica para a solução de outros problemas. (NEWELL, 1982) As subáreas da representação do conhecimento abrangem desde os aspectos puramente filosóficos da epistemologia até os problemas mais práticos do tratamento de grandes quantidades de dados.

Esta diversidade é focada em seu problema central, codificar o conhecimento humano e todas as suas variáveis, de tal forma que ele possa ser utilizado por processos capazes de raciocinar inteligentemente sobre o mundo. Este objetivo é bem apresentado por Brian Smith na sua hipótese da representação do conhecimento:

“Qualquer processo inteligente realizado por uma máquina, será formado por ingredientes estruturais que: (a) serão naturalmente percebidos pelos observadores externos como uma descrição proposicional do conhecimento exibido pelo processo, e (b) independentemente de tal atribuição semântica externa, terão um papel formal, causal e essencial na geração do comportamento que manifesta tal conhecimento.” (SMITH, 1982)

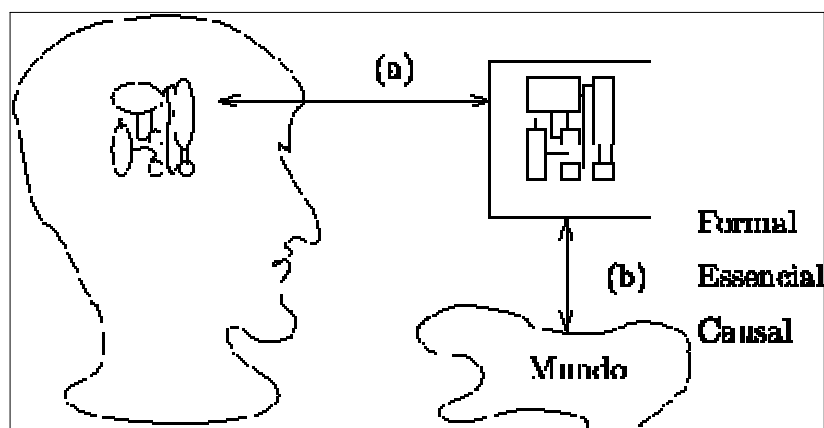


Figura 15 Hipótese da representação do conhecimento
Fonte: Smith, 1982

Uma representação bem sucedida de algum conhecimento deve, então, estar em uma forma que seja compreensível por seres humanos, e deve induzir o sistema a comportar-se como se conhecesse esse conhecimento. Os "ingredientes estruturais" que realizam estes objetivos são encontrados tipicamente nas linguagens para representação do conhecimento, tanto teoricamente quanto de forma implementada, e têm sido desenvolvidos durante esses anos.

Nas Pesquisas do *Computational Intelligence Research Laboratory (CIRL) at the University of Oregon* (1993), a representação do conhecimento é “o estudo de como o conhecimento sobre o mundo pode ser representado e quais os tipos de raciocínio podem ser feitos com esse conhecimento”.

Segundo Davis *et al* (1993), o conceito de representação de conhecimento pode ser mais bem entendido a partir das seguintes definições ligadas aos papéis que poderá desempenhar:

- Uma representação de conhecimento é um mecanismo usado para se raciocinar sobre o mundo, em vez de agir diretamente sobre ele. Neste sentido, ela é, fundamentalmente, um substituto para aquilo que representa. Este papel conduz, naturalmente, a dois importantes aspectos. O primeiro diz respeito à sua identidade projetada. Deve haver alguma forma de correspondência especificada entre o substituto e seu referente planejado no mundo. Esta correspondência é a semântica da representação. O segundo é a fidelidade, no entanto a fidelidade perfeita é, em geral, impossível, pois a única representação completamente precisa de um objeto é o objeto em si. Qualquer outra representação é imprecisa e, inevitavelmente, contém simplificações.

- Uma representação de conhecimento é uma resposta à pergunta "Em que termos devo pensar sobre o mundo", isto é, um conjunto de compromissos ontológicos. Uma vez que toda representação é uma aproximação imperfeita da realidade, ao selecionarmos uma representação, estamos tomando um conjunto de decisões sobre como e o que ver no mundo. Portanto, selecionar uma representação significa fazer um conjunto de compromissos

ontológicos. Esses compromissos determinam o que pode ser visto, enfocando alguma parte do mundo em detrimento de outras. Esta forma de ver o mundo não é apenas um efeito colateral da escolha de uma representação; ao contrário, o efeito focalizador é a parte essencial do que a representação oferece, já que a complexidade do mundo real é esmagadora. Assim, o comprometimento ontológico feito por uma representação pode ser uma de suas mais importantes contribuições.

- Uma representação de conhecimento é uma teoria fragmentada de raciocínio que especifica que inferências são válidas e quais são recomendadas. Uma representação é motivada por alguma percepção de como as pessoas argumentam ou por alguma crença sobre o que significa raciocinar de forma inteligente. A teoria de raciocínio inteligente, embutida em uma representação, é geralmente implícita, mas pode se tornar mais evidente pelo exame de três componentes: a concepção de inferência inteligente, o conjunto de inferências que a representação sanciona e o conjunto de inferências que ela recomenda. Enquanto as inferências sancionadas indicam o que pode ser inferido, as inferências recomendadas dizem respeito ao que deve ser inferido. Esta orientação é necessária, pois o conjunto de inferências sancionadas é tipicamente muito grande para ser usado indiscriminadamente. Estes componentes podem ser vistos, também, como as respostas da representação a três questões fundamentais: (i) O que significa raciocinar de forma inteligente? (ii) O que podemos inferir a partir do que conhecemos? (iii) O que devemos inferir a partir do que conhecemos?
- Uma representação de conhecimento é um meio de computação pragmaticamente eficiente. Na realidade, esta questão aborda a utilidade prática da representação. Se ela torna coisas possíveis, mas não facilmente computáveis, a representação pode, então, não ser de muita valia para o problema em mãos.
- Uma representação de conhecimento é um meio de expressão, isto é, uma linguagem na qual se pode dizer coisas sobre o mundo. Esta é, também, uma questão de utilidade prática da representação. Se ela permite expressar certas

situações, mas isto não é feito facilmente, muitas vezes não podemos, como usuários, saber se a representação não é capaz de expressar alguma coisa que gostaríamos de dizer ou se simplesmente não sabemos como usá-la. Uma representação é a linguagem na qual nos comunicamos e, assim, devemos ser capazes de falar sem esforço heróico.

Segundo Sowa (2000), a representação do conhecimento é um assunto multidisciplinar que aplica teorias e técnicas de três outros campos:

- **Lógica** que prove a estrutura formal e as regras de inferência;
- **Ontologia** que define os tipos de coisas que existem no domínio da aplicação;
- **Computação** que apóia as aplicações que distinguem a representação do conhecimento da filosofia pura.

Sem a lógica, uma representação de conhecimento é vaga, sem critérios para determinar se os comandos são redundantes ou contraditórios. Sem ontologia, os termos e símbolos são mal-definidos, passíveis de serem confundidos e de produzir confusão. E sem modelos computáveis, a lógica e a ontologia não podem ser implementadas em programas de computador. Portanto, a representação do conhecimento é a aplicação de lógica e ontologia à tarefa de construir modelos computacionais para algum domínio (SOWA, 2000, p. 12).

Segundo Campos (2004), os mecanismos de representação de conhecimento permitem, assim, que processos de formalização sobre os objetos e suas relações, em contextos predefinidos, possam ser facilmente representados. No âmbito da ciência da computação, eles servem para auxiliar a implementação de estruturas computáveis. No âmbito da ciência da informação, possibilitam a elaboração de linguagens documentárias verbais e notacionais, visando à recuperação de informação e à organização dos conteúdos informacionais de documentos. No âmbito da terminologia, esses mesmos mecanismos permitem a sistematização dos conceitos e, conseqüentemente, a elaboração de definições consistentes.

Um dos desenvolvimentos mais importantes na aplicação da representação do conhecimento nos últimos 20 anos foi proposta, estudada e desenvolvida em linguagens de representação de conhecimento baseadas em *frames*. Apesar das diferenças existentes entre essas linguagens, todas são baseadas na especificação de objetos (conceitos) e dos relacionamentos entre eles, cujas principais características são (Welty, 1996):

- **Orientação a objetos:** todas as informações sobre um conceito específico são armazenadas com esse conceito, por exemplo, nos sistemas baseados em regras, a informação sobre um conceito pode ser dispersa por toda base de regras.
- **Generalização/especialização:** longamente reconhecidas como aspectos chaves da cognição humana, as linguagens de representação do conhecimento fornecem uma maneira natural de agrupar conceitos em hierarquias nas quais os conceitos de alto nível representam os mais gerais e compartilham atributos dos conceitos abaixo.
- **Raciocínio:** a habilidade de expressar de uma maneira formal que a existência de alguma parte do conhecimento implica na existência de alguma outra parte anteriormente desconhecida é importante para a representação do conhecimento. Cada linguagem de representação do conhecimento apresenta uma diferente forma de raciocínio.
- **Classificação:** dado uma abstrata descrição de um conceito, a maioria das linguagens de representação do conhecimento fornece a habilidade de determinar se o conceito se encaixa na sua descrição, isto é na verdade, uma forma especialmente comum de raciocínio.

Orientação a objetos e generalização/especialização ajudam a fazer a representação do conhecimento mais compreensível a humanos, enquanto raciocínio e classificação ajudam um sistema a se comportar como se ele conhecesse o que é

representado. Sistemas baseados em *frames*, então, se ajustam aos objetivos da hipótese de representação do conhecimento.

De acordo com Davis *et al* (1993), a representação do conhecimento é definida pelas cinco funções que ela exerce: um substituto da realidade; um conjunto de compromissos ontológicos; a materialização da concepção de um raciocínio, com um conjunto de inferências que ela apóia e recomenda; um meio para computação eficiente e uma linguagem que permite aos humanos trocar conhecimento.

2.4.1 Formas de representação do conhecimento

2.4.1.1 Redes semânticas

Proposta por Quillian (1968), utiliza no seu formalismo os nós que representam os elementos do domínio e arcos que representam as relações entre estes elementos, explicando diversos resultados experimentais sobre o comportamento da memória humana. É uma notação gráfica composta por nodos interconectados que apresenta relações entre elementos em um domínio, que podem ser usados para representação de conhecimento, ou como ferramenta de suporte para sistemas automatizados de inferências sobre o conhecimento. (SOWA, 2002)

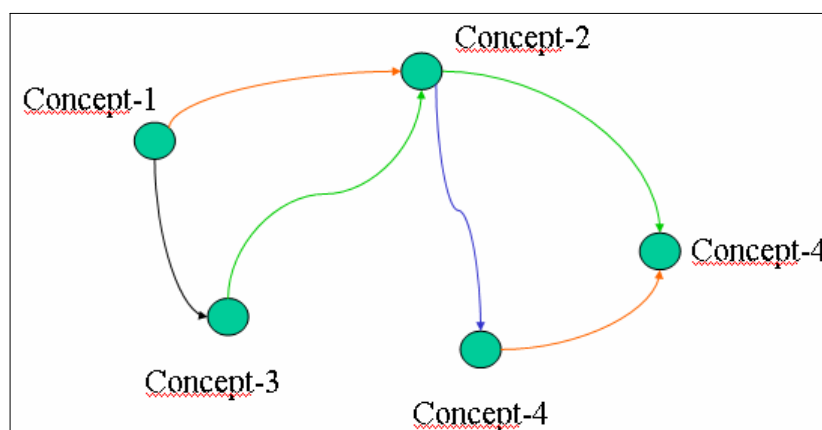


Figura 16 Modelagem da memória humana.
Fonte: Quillian, 1968

Considerada uma forma de programação orientada a objetos, têm algumas vantagens como a modularidade e a facilidade de visualização por parte das pessoas, através de simples diagramas de nós e *link's*.

As redes semânticas se concentram nas categorias de objetos e relacionamento entre eles. Os nodos representam substantivos, adjetivos, pronomes e nomes próprios. Dependendo do sistema, os nós podem ser utilizados para representar predicados, classes, palavras de uma linguagem, etc. Os arcos são reservados basicamente para representar verbos transitivos e preposições.

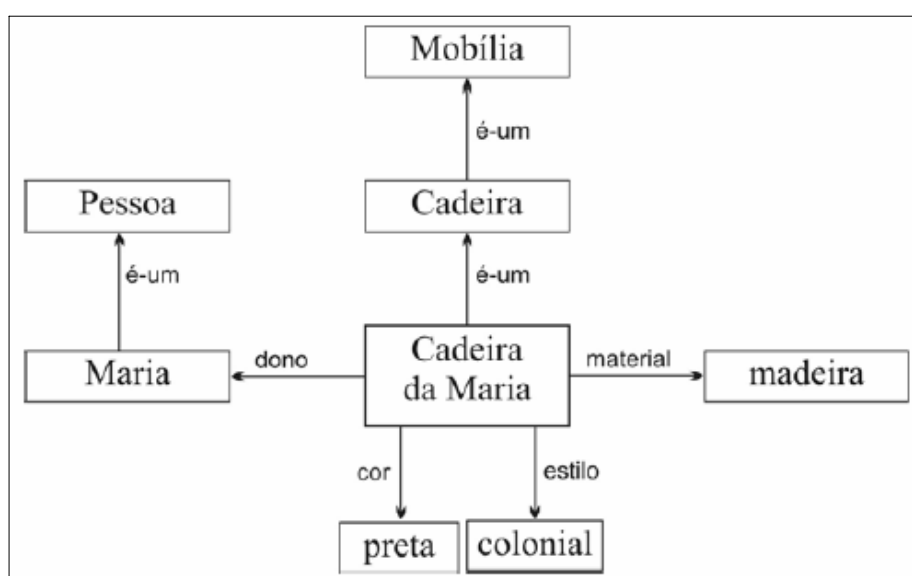


Figura 17 Exemplo de rede semântica na representação do conhecimento
Fonte: Ferneda, 2003.

A figura 17 mostra um exemplo simples da utilização de redes semânticas. Ela representa conceitos sobre “móbia”. As relações *é-um* são bastante comuns em sistemas de redes semânticas e determinam uma herança de propriedades. As demais relações (*dono*, *cor*, *estilo* e *material*) são específicas do domínio e representam propriedades dos conceitos (Rich, 1988, p.253).

Woods (1975), em seu artigo, analisa o significado dos arcos nas redes semânticas e chama a atenção para a necessidade de uma semântica formal que fundamente os sistemas baseados em redes semânticas. Ferneda (2003) nos informa que este artigo foi seguido de uma série de outros que descreviam a utilização das redes semânticas apenas como uma notação sintática alternativa para fórmulas lógicas ou, apresentavam as redes semânticas como um método independente da representação do

conhecimento, utilizando o formalismo lógico apenas como ferramenta para definição de uma semântica para os nós e os arcos.

Algumas dessas redes foram desenvolvidas explicitamente para a implementação de hipóteses sobre os mecanismos de cognição humanos, enquanto outros foram desenvolvidos primariamente para eficiência computacional (SOWA, 2002). Algumas vezes, razões computacionais podem levar às mesmas conclusões que as evidências psicológicas, provando que existe uma ligação conceitual entre elas.

2.4.1.2 Frames

Introduzido inicialmente em 1975 por Marvin Minsky, integram conhecimento declarativo sobre objetos e eventos e conhecimento procedimental sobre como recuperar informações ou calcular valores onde, um elemento de um domínio é representado por um conjunto de características.

“Um *frame* possui um nome que identifica o conceito por ele definido e um conjunto de atributos (*slots*). Um *frame* é uma estrutura de dados complexa, útil para modelar objetos do mundo real. Assemelham-se a registros de um banco de dados, porém mais poderosos e expressivos.” (MINSKY, 1975)

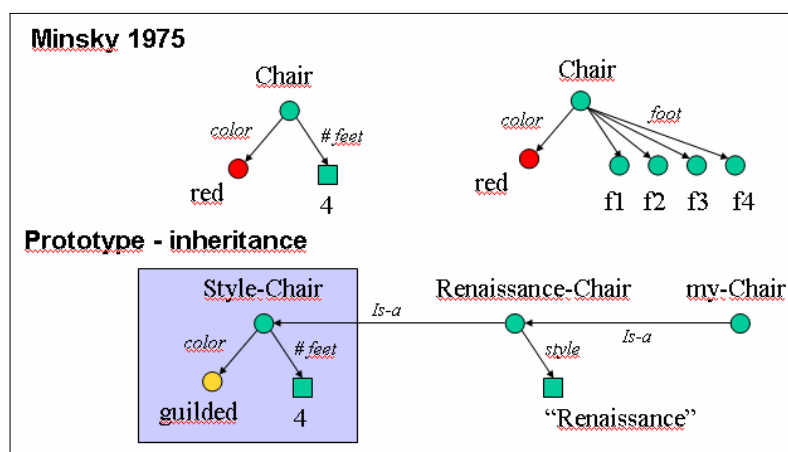


Figura 18 Frames

Fonte: Minsky, 1975

Os atributos apresentam propriedades, chamadas *facet*s, que dizem respeito ao tipo de valores e às restrições de número que podem ser associadas a cada atributo. Assim como nas redes semânticas, uma das características nos *frames* é a possibilidade de que sejam criados novos subtipos de objetos que herdem todas as propriedades da classe original. Essa herança é bastante usada tanto para a representação do conhecimento como para a utilização de mecanismos de inferência.

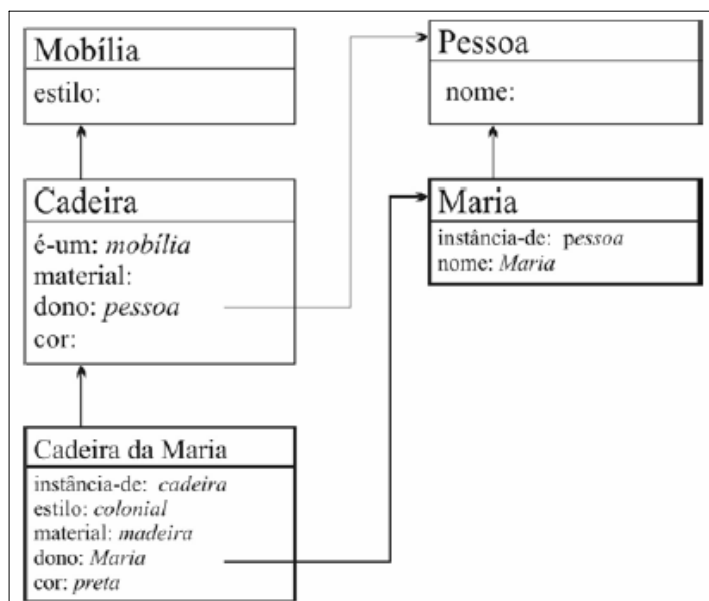


Figura 19 Exemplo da utilização de *frames* na representação do conhecimento
Fonte: Ferneda, 2003.

Basicamente um *frame* é uma coleção de atributos (“*slots*”), e valores a eles associados. Cada *frame* representa uma classe ou uma instância (elemento de uma classe). A criação de um sistema de *frames* é possível graças ao fato de que o valor de um atributo de um *frame* pode ser um outro *frame*. Um sistema de *frames* pode assim definir uma hierarquia de classes, como na figura 19. A relação *é-um* define uma relação transitiva de *subclasse*. A relação *instância-de* define a classe a qual um determinado elemento pertence. Os *frames* “*Mobília*” e “*Pessoa*” são exemplos de classes. O *frame* “*Cadeira*” é também uma classe, mas é ainda uma *subclasse* de “*Mobília*”, e herda desta a propriedade *estilo*. “*Maria*” é um elemento (ou instância) da classe “*Pessoa*”. O *frame* “*Cadeira de Maria*” é uma instância da classe “*Cadeira*”.

Conforme vemos em Freitas (2003), os *frames* são considerados precursores dos atuais objetos imperativos, guardando substancial semelhança a esses por possuírem:

- **Classes**, que funcionam como os conceitos das ontologias, normalmente organizados em hierarquias. As classes ligadas com classes mais altas são subclasses destas classes mais altas, que, por sua vez, são superclasses destas classes mais baixas;
- **Atributos**, que definem as características de membros de uma classe, preenchidos com valores do tipo especificado para o atributo. Este tipo pode ser um dos tipos básicos (inteiro, string, booleano, float, símbolo, etc) ou o tipo especial *instância* de classe, que é responsável por uma função muito importante em sistemas de *frames*: definir relações entre classes. Por exemplo, a classe Professor teria um atributo “Orientados”, que seria da classe Estudantes;
- **Instâncias** de classes, objetos criados e modelados de acordo com a definição das características da classe (seus atributos), forma de preenchê-los, etc;
- **Herança**, que são relações do tipo *é-um* entre as classes, bem semelhantes ao conceito de subclasse dos objetos. É a herança que define a hierarquia. Diferentemente da maioria dos sistemas orientados a objetos, pode existir herança múltipla.

Além da herança múltipla, *frames* têm uma outra característica própria. São dotados de estruturas pré-definidas, chamadas de *facet*s, para especificar restrições sobre os atributos. As *facet*s mais comuns, além do tipo, presentes em linguagens de *frames* são:

- *Valor default*, que atribui um valor para o atributo, caso nenhum tenha sido assinalado na criação de uma instância ou subclasse;
- *Valores permitidos (allowed-values)* para o atributo;
- *Domínio*, que determina conjuntos de valores esparsos, possíveis para um atributo (por exemplo, 1..100, ou seja, de 1 a 100);

- *Classes permitidas (allowed-classes)* para o atributo, se ele é do tipo instância de classe;
- *Cardinalidade*, determinando o número máximo e mínimo de elementos de um atributo, pois ele pode ser multivalorado;
- *Documentação*, para descrever o atributo.

Observe-se, ainda, que a faceta *valor default* faz com que instâncias sejam consideradas exceções à classe, o que, por sua vez, faz com que um sistema de *frames* se enquadre como uma *lógica não-monotônica*¹⁵, uma vez, que, se um dado é acrescentado à base de conhecimento, a conclusão do raciocínio torna-se diferente da normalmente esperada.

Outra característica é a existência de atributos inversos, que estabelecem uma correspondência biunívoca entre si. Por exemplo, o atributo “Orientados” da classe Professor é inverso do atributo “Orientador” da classe Aluno.

2.4.1.3 Redes Neurais

Sabe-se que o cérebro é composto de bilhões de *neurônios*, que são células formadas por três seções com funções específicas e complementares: *corpo*, *dendritos* e *axônio*, e que trocam informações na forma de impulsos nervosos provenientes de outras células. A conexão entre dois neurônios é chamada de sinapse, e através delas, os neurônios se unem e formam as redes neurais.

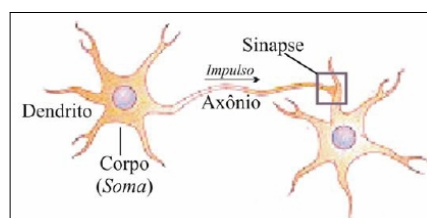


Figura 20 Representação simplificada de um neurônio
Fonte: Ferneda, 2003.

¹⁵ **não-monotônica**, o que significa que podemos tolerar ambigüidades e informações incompletas.

As primeiras informações mencionadas sobre a neuro-computação datam de 1943, em artigos de McCulloch e Pitts, em que sugeriam a construção de uma máquina baseada ou inspirada no cérebro humano. (Tatibana e Kaetsu¹⁶, 2002) As redes neurais artificiais consistem em um método de solucionar problemas de inteligência artificial, construindo um sistema que tenha circuitos que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas.

De uma forma simplificada, uma rede neural artificial pode ser vista como um grafo onde os nós são os neurônios e as ligações fazem as funções das sinapses, como na figura 21:

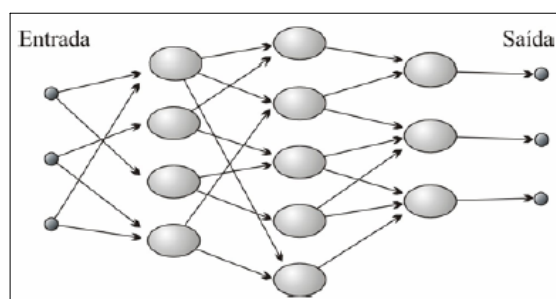


Figura 21 Representação de uma rede neural artificial
Fonte: Ferneda, 2003.

As redes neurais são particularmente úteis para lidar com dados ligados a valores reais, consistindo em uma simulação, através de “*software*” ou “*hardware*”, de alguns aspectos do sistema nervoso biológico, na tentativa de permitir a realização de tarefas cognitivas de forma mais eficaz do que as obtidas através do processamento serial, onde se deseja obter uma saída também dotada de valores reais.

2.4.1.4 *Scripts*

Script é um tipo de *frame* que descreve temporariamente o que aconteceu. Um exemplo comumente usado é a descrição de irmos a um restaurante, aguardando para sentar-se, receber o cardápio, fazer o pedido, etc. São arquivos contendo uma linguagem

¹⁶ <http://www.din.uem.br/ia/neurais/>

interpretada, precisam de *software's* interpretadores porque são blocos não compilados, que são interpretados no tempo de execução.

Fáceis de editar, por poderem ser editados por qualquer editor de texto, eles são vulneráveis. É possível fazer um *script* para que execute atividades maliciosas ao ser interpretado. Esses são chamados de *scripts* maliciosos. Esses *scripts* podem ser programados para editar códigos de outros *scripts*. Os *scripts* maliciosos executam atividades de vírus, por exemplo: criar e excluir arquivos, modificar chaves do registro do Windows e fazer download de arquivos e executá-los.

2.4.2 Representação do conhecimento na Inteligência Artificial

Segundo Medeiros (1999), os modelos de representação do conhecimento na área de Inteligência Artificial constituem-se de estruturas completas ou subestruturas de uma estrutura de dados maior, denominadas de bases de conhecimento, nas quais conhecimentos sobre determinados domínios são armazenados e passíveis de serem utilizadas por um programa especialista para desempenhar suas funções.

O termo Inteligência Artificial surgiu em 1956 nos Estados Unidos buscava estudar como poderiam ser criadas máquinas inteligentes e como elas afetariam os processos existentes, tentando entender o comportamento de entidades inteligentes e como essas entidades podem ser criadas e utilizadas para determinados fins, apresentando algumas ramificações, como as descritas abaixo:

2.4.2.1 Sistemas especialistas

Segundo Ferneda (2003), um sistema especialista é um sistema computacional que procura representar o conhecimento de um especialista humano em um domínio particular, de maneira a auxiliar na tomada de decisões e na resolução de problemas relacionados a esse domínio.

Sistemas Especialistas são sistemas que solucionam problemas que são resolvíveis apenas por pessoas especialistas (que acumularam conhecimento exigido) na resolução destes problemas. (FEIGENBAUM, 1981)

Para Fávero¹⁷ (2004), um Sistema Especialista é aquele capaz de apresentar uma decisão justificada, a partir de uma base de informações que é projetada e desenvolvida para atender a uma determinada aplicação, tal como um especialista e limitada à determinada área do conhecimento humano.

Assim como o especialista, apresenta suas decisões sobre um assunto a partir de fatos que encontra, das hipóteses que formula e através do conhecimento encontrado em sua memória, armazenado durante os anos de sua formação e no decorrer de sua vida profissional que, de acordo com sua experiência acumulada sobre o assunto, possibilita sua decisão.

Sempre baseado no conhecimento prévio acumulado, pode, durante seu processo de raciocínio, não chegar a uma decisão se os fatos que dispõe para aplicar seu conhecimento não forem suficientes, ou ainda, chegar a uma conclusão errada, que é justificada em função dos fatos que encontrou.

Um sistema especialista é composto de: uma *base de conhecimento* na qual está representado o conhecimento relevante sobre o problema, e um conjunto de métodos de manipulação desse conhecimento: o *motor de inferência* (figura 22).

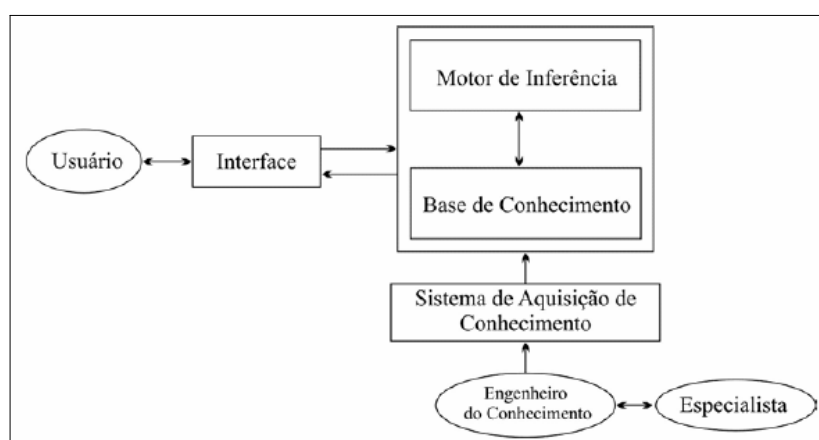


Figura 22 Estrutura de um sistema especialista
Fonte: Ferneda, 2003.

¹⁷ <http://www.din.uem.br/ia/especialistas/index.html>

Um Sistema Especialista deve, além de inferir conclusões, ter capacidade de aprender novos conhecimentos e, desse modo, melhorar o seu desempenho de raciocínio, e a qualidade de suas decisões.

Segundo a Wikipedia¹⁸, há geralmente 3 tipos de usuário envolvidos com sistemas especialistas. Primeiramente entre esses tipos há o usuário final, aquele indivíduo que utiliza o sistema para auxiliá-lo na resolução de problemas. Envolvidos com a construção e manutenção do sistema estão os outros dois grupos: especialista no domínio do problema, que constrói a base de conhecimento, e um engenheiro de conhecimento, que auxilia o especialista a representar o conhecimento e que determina a técnica de inferência necessária para se conseguir solucionar um problema.

2.4.2.2 Sistema de resolução de problemas

Já na abordagem de sistema especialista, todo o conhecimento relativo ao domínio do problema é codificado exclusivamente nas estruturas de dados. Nenhum conhecimento é armazenado nas instruções ou nos programas propriamente ditos. Vários benefícios surgem imediatamente dessa estratégia.

Um exemplo pode ajudar a destacar a diferença entre a abordagem procedural tradicional e a abordagem de sistemas especialistas, como o exemplo do problema da declaração de imposto de renda.

Na abordagem tradicional, as estruturas de dados armazenam as informações do contribuinte e das alíquotas de imposto, e um programa representa o conhecimento necessário para a declaração de imposto de renda, contendo os comandos que relacionam as informações do contribuinte com as alíquotas e os cálculos necessários para realizar a declaração. É essa representação do conhecimento que é difícil para um especialista em impostos compreender ou modificar.

¹⁸ <http://en.wikipedia.org>

Na abordagem de sistema especialista, a informação sobre o contribuinte e as alíquotas, bem como os cálculos necessários, estão novamente armazenadas em estruturas de dados, mas o conhecimento que descreve o relacionamento entre elas também é armazenado nas estruturas de dados. O programa de um sistema especialista é independente do conhecimento do domínio do problema (no nosso exemplo, do conhecimento das alíquotas e rendas do contribuinte e do relacionamento entre elas, os cálculos) e serve para processar estruturas de dados independentes da natureza do problema que essas estruturas descrevem.

Por exemplo, há programas interativos específicos para capturar descrição de dados, programas para capturar e organizar as descrições, programas para processar as declarações que representam relacionamentos semânticos dentro do domínio do problema e um algoritmo para controlar a seqüência de processamento e o foco.

A arquitetura geral de um sistema especialista compreende dois componentes principais: um conjunto de declarações totalmente dependentes do domínio do problema e que é chamado de base de conhecimento ou base de regras, e um programa independente do domínio do problema (apesar de altamente dependente das estruturas de dados) chamado de motor de inferência.

2.4.2.3 Agentes Inteligentes

Segundo Lustosa (2004, p.4), na área de gerência de redes, detecção de intrusão, comércio eletrônico e educação a distância auxiliada por computador, a tecnologia de agentes inteligentes permite a flexibilidade da negociação entre o sistema e o usuário humano, além de estar sempre monitorando o ambiente, gerando informações que podem ser analisados posteriormente por outros sistemas inteligentes.

Um agente é um sistema de computador que está situado em um *ambiente* e que é capaz de executar *ações* autônomas de forma *flexível* neste ambiente, a fim de satisfazer seus objetivos de projeto. (WOOLDRIDGE, 1999 *apud* LUSTOSA, 2004). As três características suficientes para classificar um agente como inteligentes são:

- **Reativo:** agentes devem perceber seu ambiente e responder oportunamente às mudanças que ocorrem nele;
- **Pró-ativo:** agentes não devem simplesmente atuar em resposta ao ambiente, devem exibir um comportamento oportunista e direcionado ao seu objetivo e tomar a iniciativa quando apropriada;
- **Social:** agentes devem interagir, quando apropriado, com outros agentes artificiais ou humanos para completar suas próprias soluções de problemas ou ajudar outros com suas atividades.

Um agente é uma entidade à qual se pode associar uma identidade única e que é capaz de realizar cálculos formais. Um agente pode ser considerado como um meio que produz certo número de ações a partir dos conhecimentos e mecanismos internos que lhe são próprios. (GASSER, 2002 *apud* LUSTOSA, 2004)

Os agentes inteligentes, presentes na resolução dos problemas dos usuários comuns, são utilizados hoje em dia na Internet desde sites que comparam preços de produtos até mecanismos de busca inteligentes, otimizando os processos de recuperação da informação.

2.4.2.4 Redes Bayesianas

Modelo que trabalha com o conhecimento incerto e incompleto através da Teoria da Probabilidade Bayesiana publicada pelo matemático Thomas Bayes em 1763. O conhecimento incerto é aquele que apresenta deficiências. Em algum momento, os dados podem ser uma representação não exata, parcial ou aproximada da realidade. (BHATNAGAR; KANAL, 1986 *apud* LUSTOSA, 2004).

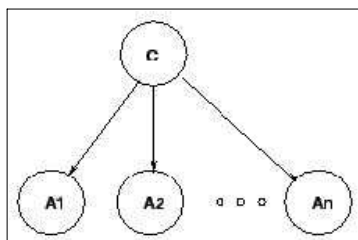


Figura 23 Exemplo de Rede Bayesiana
Fonte: Lustosa, 2004

A partir de novas informações com certo nível de incerteza, pode adaptar-se, o que é uma de suas principais características, alterando suas dependências e conceitos, permitindo que as probabilidades prévias confirmem ou criem novos conceitos.

Sua representação é feita através de um grafo direcionado acíclico no qual os nós representam variáveis de um domínio e os arcos representam a dependência condicional ou informativa entre as variáveis. Para representar a força da dependência, são utilizadas probabilidades, associadas a cada grupo de nós pais-filhos na rede (PEARL, 1988 *apud* LUSTOSA, 2004).

Sowa (2000) apresenta um estudo completo e detalhado sobre as diversas formas de representação do conhecimento, e é uma referência para quem deseja aprofundar o assunto.

2.4.3 Modelagem de domínio

Modelagem de domínio é o campo no qual uma aplicação de representação de conhecimento para um domínio específico é estudado e executado.

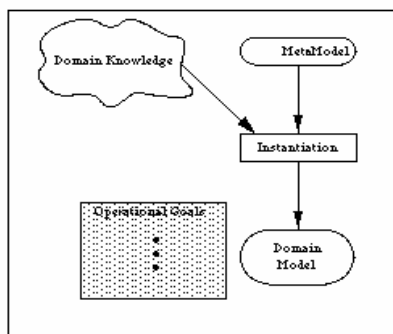


Figura 24 Framework de Modelagem de Domínio
Fonte: ISCOE, TAM, LIU, 1991

Na figura 24, *Domain Knowledge* (domínio do conhecimento) refere-se ao conhecimento possuído por um especialista do domínio, que não é bem definido e de certo modo de difícil acesso para outros, e que deve ser codificado de alguma forma.

MetaModel (meta-modelo) refere-se ao formalismo da Representação do Conhecimento, tipicamente uma linguagem de representação do conhecimento, que será utilizado no nível simbólico (NEWELL, 1982) para a representação deste conhecimento em computadores.

A *instantiation* (instanciação) refere-se ao processo de capturar o conhecimento do domínio e fisicamente representá-lo usando o meta-modelo, também conhecido como aquisição do conhecimento. (SCHOEN, 1991)

O *domain model* (modelo do domínio) refere-se à base de conhecimento que resulta da instanciação, e os *operational goals* (objetivos operacionais) tipicamente não são formalmente representados, mas refere-se a razão para a qual o modelo do domínio foi construído e será usado.

Na representação do conhecimento, uma das formas de modelagem de domínio atualmente utilizado é conhecida como ontologia.

2.5 Ontologias

Atualmente, a popularização de ferramentas para armazenamento e divulgação de informações em meios digitais cresce exponencialmente, dificultando o processo de classificação e recuperação das mesmas e, com a grande rede global de computadores, a Internet, um grandioso número de documentos de todos os tipos tem surgido, sua maioria escrita em linguagem natural, com suporte de diferentes mídias, dificultando cada vez mais a avaliação e estruturação adequada dessas informações.

Nesse sentido, e pelo fato do ser humano não conseguir processar uma grande fração desses documentos, é que surge a necessidade dos métodos automatizados de classificação e recuperação de informações que vêm sendo desenvolvidos. No entanto, a maioria deles baseia-se ainda em aspectos estatísticos e sintáticos, levando-se em consideração a frequência e a distribuição das palavras presentes nos documentos.

De modo geral, métodos semânticos automatizados ainda não são possíveis, ou estão em fase embrionária, máquinas não podem interpretar e entender o significado das informações em linguagem natural devido à complexidade de se obter a semântica de uma palavra em um determinado contexto.

Com isso, está surgindo uma nova semântica para o conteúdo da Internet, a *Web Semântica*, buscando expressar informações de forma precisa e interpretável por máquinas, onde agentes de *software* sejam capazes de processar um entendimento compartilhado sobre o significado dos termos que descrevem um dado.

Outro problema está relacionado à comunicação entre empresas e seus sistemas, diz respeito à recuperação e troca de informações ou conhecimento entre elas e a uma meta-linguagem que desse suporte e pudesse garantir a possibilidade dessas trocas sem inconsistências.

Em concordância com Uschold e Gruninger (1996, p.2-3), vemos que pessoas, organizações e sistemas¹⁹ precisam se comunicar entre si, mas suas necessidades, contextos e pontos de vista são muito variados em relação a determinado assunto. Utilizando-se de diferentes jargões, os seus conceitos, estruturas e métodos se diferem, e a consequência disso, é a falta de um entendimento compartilhado que os conduz a:

- Uma comunicação “pobre” entre essas pessoas e organizações;
- Dificuldades na identificação e definição de requerimentos para os sistemas, cujos métodos, paradigmas, linguagens e ferramentas, pelas suas diferenças, limitam-nos severamente em termos de:

¹⁹ Entende-se aqui por **sistema**, todos os *softwares* de computador que possam estar sendo utilizados. Por exemplo; sistemas operacionais, agentes de *software*, etc.

- ✓ Interoperabilidade e;
- ✓ Potencial de re-uso e compartilhamento.

Ainda Guarino *et al* (2000, p.1) explana que pesquisadores em sistemas de informação e engenharia de sistemas sabem perfeitamente que nenhum sistema continuará funcionando isolado por longo tempo, deixando claro que essas comunidades irão requerer algum tipo de integração e interoperabilidade, discutindo sobre os padrões de conteúdo na *Web* principalmente em termos de ontologias e padrões de *e-commerce*.

Nesse contexto, surge uma grande demanda para o desenvolvimento de sistemas que trabalhem diretamente com a informação ou conhecimento, surgindo novos instrumentos para a organização e estruturação desse conhecimento, sendo que, um deles é conhecido como “ontologia”.

2.5.1 Origem

O termo ontologia tem origem no grego “ontos” (ser) e “logos” (palavra), sendo usado em muitos discursos nas mais diversas áreas do conhecimento (ALMEIDA e BAX, 2003, p.2), com origem na palavra aristotélica “categoria”, que pode ser usada para classificar alguma coisa, serve de base para classificar qualquer entidade e introduz ainda o termo “*differentia*” para propriedades que distinguem diferentes espécies do mesmo gênero, sendo a conhecida técnica de herança o processo de mesclar *differentias* definindo categorias por gênero.

Entretanto, o termo que veio da Filosofia, apresenta em seu ponto de vista, com “O” maiúsculo, a representação do ramo da Filosofia que lida com a natureza e a organização da realidade (GAVA e MENEZES, 2003), com o objetivo de distinguir o estudo do ser como tal.

O dicionário Oxford de Filosofia define ontologia como “[...] o termo derivado da palavra grega que significa ‘ser’, mas usado desde o século XVII

para denominar o ramo da metafísica que diz respeito àquilo que existe” (BLACKBURN; MARCONDES, 1997 *apud* ALMEIDA; BAX, 2003).

A controvérsia está na definição do termo ontologia (com “o” minúsculo), cuja popularidade cresce a cada dia juntamente com termos relacionados como “relações ontológicas” e “compromissos ontológicos”, principalmente nas comunidades existentes na sociedade do conhecimento que nem sempre mantêm a mesma interpretação.

Na Ciência da Informação, o fato da representação do conhecimento também ser seu objeto de estudo, torna a ontologia um aspecto que deve ser explorado e investigado, mas, devido ao seu uso diversificado tende a ter um significado muito vago, existindo muitas definições para o termo.

2.5.2 Definições

Atualmente o termo ontologia tem sido utilizado na Ciência da Informação com um sentido diferente daquele tradicional adotado na Filosofia. São diversas as definições apresentadas na literatura e algumas delas são apresentadas em seguida com o objetivo de esclarecer o novo uso do termo.

Adotado pela comunidade de Inteligência Artificial (IA), o termo ontologia descreve uma explicação sistemática da existência, onde, o que “existe” é o que pode ser representado em um formalismo declarativo e com um texto legível por humanos, que Gruber (1993, p.1; 1996 *apud* CORAZZON, 2002, p.1) define como:

Uma ontologia é uma especificação explícita de uma conceitualização. [...] Em tal ontologia, definições associam nomes de entidades no universo do discurso (por ex. classes, relações, funções, etc) com textos que descrevem o que os nomes significam e os axiomas formais que restringem a interpretação e o uso desses termos [...].

O termo conceitualização corresponde a uma coleção de objetos, conceitos e outras entidades que se assume existirem em um domínio e os relacionamentos entre eles (GENESERETH e NILSON, 1987) e é uma visão abstrata e simplificada do mundo que se deseja representar. Como a ontologia herda ou incorpora uma espécie de visão do

mundo a respeito de determinado domínio, ela então concebe um conjunto de conceitos (entidades, atributos, processos etc), suas definições e suas inter-relações (USCHOLD e GRUNINGER, 1996, p.5), que pode ser implícita, estando os conceitos subentendidos por uma pessoa ou incorporados a uma parte do sistema. Por exemplo, em um componente de *software*, ou explícita, com a representação total ou parcial de uma conceitualização.

A definição mais popular dentro da comunidade de representação de conhecimento é a proposta por Grubber (1993) que diz que uma ontologia é "especificação formal e explícita de uma conceitualização compartilhada", onde formal significa que a ontologia pode ser expressa em uma linguagem formal, explícita significa que é um objeto de nível simbólico e, compartilhada indica que o conhecimento é aceito dentro da comunidade.

A ontologia deve fazer uma especificação formal de uma área de conhecimento (BENJAMINS e GÓMEZ-PÉREZ, 1999). Cinco componentes foram definidos para esta formalização (GRUBER, 1993):

- **Conceitos:** podem representar qualquer coisa em um domínio, como uma tarefa, uma função, uma estratégia, etc;
- **Relações:** representam um tipo de interação entre os conceitos no domínio, sendo a cardinalidade sempre n:n;
- **Funções:** são um caso especial de relações, sendo a cardinalidade n:1;
- **Axiomas:** são as sentenças que são sempre verdadeiras, independente da situação (são constantes);
- **Instâncias:** são utilizadas para representar os elementos do domínio.

Um exemplo (MOREIRA *et al*, 2004) pode ser útil para ilustrar as noções de conceitualização e especificação. Suponha a visão do mundo formada pelo domínio acadêmico. Uma conceitualização desta visão poderia conter os conceitos de "aluno",

"professor", "disciplina", "nota", "curso", "ministra", etc. Uma especificação explícita desta conceitualização em lógica de primeira ordem poderia ser algo como:

<i>Sentença em Lógica</i>	<i>Linguagem natural</i>
$\forall x \text{ disciplina}(x) \Rightarrow \exists y(\text{professor}(y) \wedge \text{ministra}(x, y))$	Para toda disciplina existe um professor que a ministra.
$\forall x \text{ disciplina}(x) \Rightarrow \exists y(\text{curso}(y) \wedge \text{parte-de}(x, y))$	Toda disciplina é parte de um curso.
$\forall x \text{ aluno}(x) \Rightarrow \exists y(\text{curso}(y) \wedge \text{curso}(x, y))$	Todo aluno cursa um curso.
$\forall x \text{ aluno}(x) \Rightarrow \text{Corpo-acadêmico}(x)$	Todo aluno é um elemento do corpo acadêmico.
$\forall x \text{ professor}(x) \Rightarrow \text{Corpo-acadêmico}(x)$	Todo professor é um elemento do corpo acadêmico.

Figura 25 Especificação explícita de uma conceitualização.
Fonte: Moreira et al, 2004

As descrições de um estado particular do domínio não são incluídas, tais como "João cursa Medicina" ou "Ana é professora". Uma apresentação menos formal da conceitualização acima poderia ser feita usando uma notação gráfica, onde os nós são os objetos e os arcos são relações:

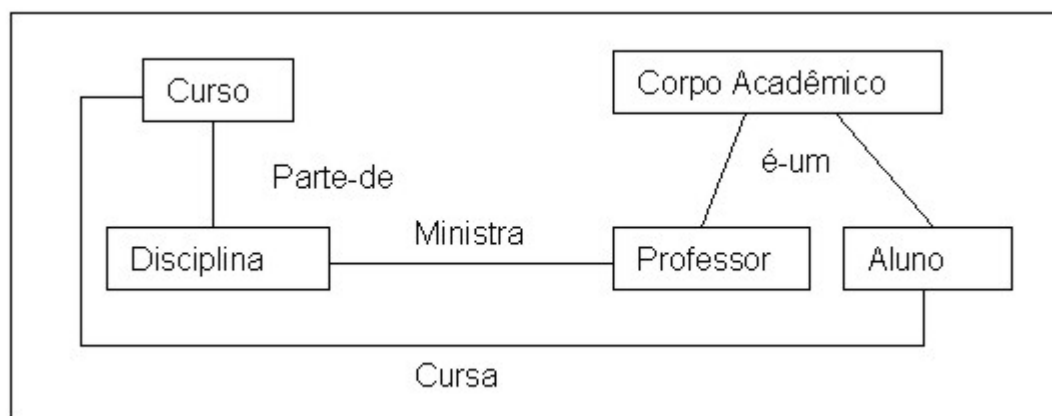


Figura 26 Especificação explícita de uma conceitualização na forma gráfica.
Fonte: Moreira et al, 2004

Apesar das representações acima capturarem uma porção de uma conceitualização, será que é este o conhecimento registrado por uma ontologia? Tradicionalmente, as representações acima seriam chamadas de bases de conhecimento e não de ontologia. Seria ontologia um novo nome para algo que já existia? Para (GUARINO, 1995b) o conhecimento acima não é o conhecimento que deve ser

registrado em uma ontologia. Segundo ele esse conhecimento pertence aos níveis denominados, *lógico e epistemológico* e não ao *nível ontológico*.

No nível lógico o conhecimento é codificado por meio de primitivas básicas como predicados e funções. Já na opinião de (GUARINO, 1995b), a distinção entre a visão epistemológica e a visão ontológica na representação do conhecimento está no fato de que a primeira se preocupa com a *forma do conhecimento* e a última se preocupa com a *forma do ser*. Então, a epistemologia fornece uma estrutura para o conhecimento e a ontologia fornece uma descrição das formas de seus elementos. Assim sendo, os conceitos apresentados nos outros níveis são vistos como objetos e sua natureza, como tal são observados.

O registro desta observação é o que uma ontologia formal deve conter. Por exemplo, usando-se as categorias elaboradas por (DAHLBERG, 1978) pode-se registrar o seguinte conhecimento de natureza ontológica sobre os conceitos capturados do domínio acadêmico:

Todo Professor e aluno é um objeto material .
Curso e disciplina são objetos imateriais .
Cursar é um processo .

Figura 27 Categorização de conceitos.
Fonte: Moreira et al, 2004

Pode-se observar que o conhecimento ontológico procura descrever a natureza dos objetos de forma independente de um problema específico. Uma vez que o propósito da modelagem é construir uma visão do mundo para comunicação de conhecimento, e não para a resolução de um problema específico, a necessidade do estabelecimento de compromissos ontológicos é mais importante do que a visão epistemológica. Portanto, a visão ontológica deve prevalecer sobre a visão epistemológica quando se trata de sistemas de recuperação de informação e compartilhamento de conhecimento.

Para situar o papel da ontologia na representação do conhecimento, pode-se basear nos níveis de representação mostrados na figura 28 a seguir, que é uma

adaptação da apresentada em (GUARINO, 1995b). Estes níveis foram propostos, originalmente, por (BRACHMAN, 1979 *apud* GUARINO, 1994). No entanto, a proposta de Brachman não incluía o nível ontológico.

<i>Nível</i>	<i>Primitivas</i>	<i>Exemplo</i>
Lógico	Predicados, funções	$\forall x \text{ aluno}(x) \Rightarrow \text{Corpo-acadêmico}(x)$ $\exists x \text{ aluno}(x) \wedge \text{Inteligente}(x)$
Epistemológico	Relações de estruturação	aluno é uma subclasse do corpo acadêmico. Existem alunos que são inteligentes.
Ontológico	relações ontológicas	Todo aluno é um objeto material . Inteligente é uma qualidade .
Conceitual	Relações Conceituais	
Lingüístico	Termos Lingüísticos	Substantivos - aluno, professor Adjetivo - inteligente. Verbo - cursar

Figura 28 Níveis de representação para o conhecimento.
Fonte: Moreira et al, 2004

Pode-se notar que o conhecimento representado pelas sentenças, em lógica de primeira ordem, pode ser expresso em linguagem natural como *todo aluno pertence ao corpo acadêmico* e *existem alunos que são inteligentes*. A tradução de sentenças de primeira ordem para a linguagem natural é facilitada pelo uso de palavras da linguagem natural como predicados (aluno, corpo acadêmico e inteligente).

Este é um artifício muito usado pelos projetistas de bases de conhecimento, que esperam, desta forma, vincular a teoria formal com o significado pretendido. No entanto, como salienta Guarino (1995b) este objetivo não pode ser alcançado desta forma, pois as palavras em linguagem natural não possuem uma semântica precisa devido à ambigüidade e à imprecisão de seus elementos. Por exemplo, pode-se utilizar a palavra livro como um símbolo na linguagem de representação do conhecimento com o intuito de associar o símbolo ao conceito de livro.

No entanto, a palavra livro é usada tendo mais de um sentido. O sentido associado à palavra na sentença *Pedro emprestou o livro* é diferente do sentido associado à palavra na sentença *Eu li o livro que Ana escreveu*. No primeiro caso, a palavra denota o conceito de livro como um objeto concreto, ou seja, no sentido de exemplar, identificado no tempo e no espaço. Já o segundo caso, a palavra denota o conceito de livro como um objeto abstrato, ou seja, no sentido de um produto intelectual, identificado por um autor e título.

No *nível epistemológico*, é capturada a estrutura do conhecimento por meio de primitivas de estruturação. Assim, no exemplo da figura 28, *aluno* é declarado como subclasse do *corpo acadêmico* e *inteligente* como uma qualidade. Esta visão, provavelmente coincide com a interpretação intuitiva da maioria das pessoas, que vêem *inteligentes* como um atributo de um objeto. No entanto, nada impede que outra pessoa tenha uma outra visão e interprete *inteligente* como um objeto e *aluno* como qualidade.

No entanto, é preciso fixar a interpretação, para que seja possível produzir bases de conhecimentos que possam ser reutilizáveis e compartilháveis. Segundo Guarino (1995b) a fixação da interpretação depende do *nível ontológico*, onde são estabelecidos os compromissos ontológicos. Assim interpretações espúrias, como *aluno* ser uma qualidade de *inteligente*, podem ser bloqueadas. No exemplo da figura 28, *aluno* é definido como pertencendo à categoria *Objeto material*, enquanto que *inteligente* é definido como pertencendo à categoria *qualidade*. Estas categorias fazem parte das categorias propostas por Dahlberg (1978). Os compromissos ontológicos podem ser registrados em qualquer linguagem suficientemente poderosa que permita a reificação²⁰, como a Lógica de Primeira Ordem.

No *nível conceitual* as primitivas possuem uma interpretação cognitiva bem definida baseada em conceitos elementares independentes de linguagem, como ações ou papéis temáticos. No *nível lingüístico* as primitivas são associadas diretamente à categorias lingüísticas, como nomes e verbos.

²⁰ **Reificação** vem do latim *res* ou coisa. Uma linguagem que permite reificação é uma linguagem que permite transformar predicados em termos. Por exemplo, podemos expressar em lógica de primeira ordem que *Pedro é homem* por meio da fórmula *homem(Pedro)* ou através da fórmula reificada *propriedade(Pedro, Homem)*. O predicado Homem na primeira fórmula tornou-se um termo na segunda. (Moreira *et al*, 2004)

Outro pesquisador que faz a distinção entre o aspecto ontológico e o epistemológico é (POLI, 2001). Para ele, uma ontologia "é a teoria de objetos. E é assim para todo tipo de objeto, concreto e abstrato, existente e não-existente, real e ideal, independente e dependente. Qualquer que sejam os objetos que se esteja lidando, a ontologia é a teoria deles".

Partindo desta visão, Poli (2001) argumenta que existe um equívoco em grande parte das definições sobre ontologia, confundido-a com a *epistemologia* que é a teoria dos diferentes tipos de conhecimento e das formas que são usados. Segundo ele, a diferença entre esses dois enfoques se torna evidente quando são listados os conceitos da ontologia e os da epistemologia. Os conceitos ontológicos são: objetos, processo, particular, individual, todo, parte, evento, propriedade, qualidade, estado, etc. Já os conceitos epistemológicos são: crenças, conhecimento, conhecimento incerto, conhecimento errado, etc.

Este conjunto de objetos e suas relações são expressos em um vocabulário representacional onde o programa baseado em conhecimento representa o conhecimento. A definição proposta por Gruber (1993) é discutida em (GUARINO E GIARETTA 1995 *apud* CORAZZON, 2002, p.1):

[...] um ponto inicial nesse esforço de tornar claro o termo, será uma análise da interpretação adotada por Gruber. O principal problema com tal interpretação é que ela é baseada na noção conceitualização, a qual não corresponde a nossa intuição. [...] Uma conceitualização é um grupo de relações extensionais descrevendo um 'estado das coisas' particular, enquanto a noção que temos em mente é uma relação intensional, nomeando algo como uma rede conceitual a qual se superpõe a vários possíveis 'estados das coisas'.

Uma definição intensional consiste de uma lista de características do conceito onde o vocabulário formado por predicados lógicos forma a rede conceitual que confere o caráter intencional às ontologias, por exemplo, lâmpada incandescente é a lâmpada elétrica que emite luz a partir do aquecimento de um filamento pela corrente elétrica. A lâmpada incandescente é definida como o auxílio do gênero mais próximo (lâmpada elétrica) e de suas características.

Uma definição extensional é uma enumeração de aspectos de todas as espécies que são do mesmo nível de abstração. Por exemplo, os planetas do sistema solar são Mercúrio, Vênus, Terra, Marte, Júpiter, Saturno, Urano, Netuno e Plutão.

A ontologia define as regras que regulam a combinação entre os termos e as relações. As relações entre os termos são criadas por especialistas e os usuários fórmulam consultas usando os conceitos especificados. Uma ontologia define assim uma “linguagem” (conjunto de termos) que será utilizada para fórmular as consultas.

Segundo Russel e Norvig (1995, p. 222) a palavra ontologia significa uma teoria particular, expressa por meio de um vocabulário, da natureza do ser e do existir, ou seja, um vocabulário ou uma lista informal dos conceitos em um domínio. Já os axiomas são considerados apenas na formalização.

Para Uschold e Gruninger (1996, p. 5), ‘ontologia’ é o termo usado para se referir ao entendimento compartilhado de algum domínio de interesse que pode ser usado como um *framework* unificado para resolver os problemas de comunicação, interoperabilidade e engenharia de sistemas.

BORST (1997, p.12) apresenta uma definição simples e completa, similar a de Gruber (1993), que abrange as principais características da ontologia: “uma ontologia é uma especificação formal e explícita de uma conceituação compartilhada”. Nessa definição, formal significa legível para computadores, especificação explícita diz respeito a conceitos, propriedades, relações, funções, restrições, axiomas, explicitamente definidos, compartilhado quer dizer conhecimento consensual e, conceitualização diz respeito a um modelo abstrato de algum fenômeno do mundo real.

É nesse contexto que identificamos a necessidade de se construir uma ontologia, formalizando conceitos e relações pertinentes a esse domínio. (GUARINO, 1998), ao rever a definição de conceitualização fazendo uso do aspecto intensional anteriormente citado, para obter uma interpretação mais satisfatória:

[...] ontologia se refere a um artefato constituído por um vocabulário usado para descrever uma certa realidade, mais um conjunto de fatos explícitos e aceitos que dizem respeito ao **sentido pretendido** para as palavras do

vocabulário. Este conjunto de fatos tem a forma da teoria da lógica de primeira ordem, onde as palavras do vocabulário aparecem como predicados unários ou binários. (*grifo nosso*)

Para (SOWA, 1999), ontologia é um “catálogo de tipos de coisas”, que se supõem existir um domínio, na perspectiva de uma pessoa que usa uma determinada linguagem.

Chandrasekaran e Josephson (1999) definem ontologias como teorias de conteúdo sobre tipos de objetos, propriedades dos objetos e relações entre objetos que são possíveis em um domínio de conhecimento específico. Segundo eles, as ontologias fornecem termos potenciais para descrever nosso conhecimento sobre o domínio.

Uma ontologia define um vocabulário comum para pesquisadores que precisam compartilhar informações dentro de um domínio (NOY e MCGUINNESS, 2001, p.1) e isso inclui também, máquinas que interpretam as definições de conceitos básicos dentro do domínio e as relações entre eles. Já na (p.2) apresentam uma importante diferença que existe entre uma ontologia e a programação orientada a objetos no que se refere ao desenho de suas classes e relações, onde eles apresentam que os programadores tomam suas decisões baseados em propriedades operacionais da classe, enquanto os projetistas de ontologias tomam suas decisões baseados em propriedades estruturais da classe.

Trata-se de “uma teoria que diz respeito a tipos de entidades e, especificamente, a tipos de entidades abstratas que são aceitas em um sistema que utiliza uma linguagem” (MERRIAM-WEBSTER; GOVE, 2002 *apud* CORAZZON, 2002, p. 1).

Segundo Almeida e Bax (2003, p.7) uma ontologia é criada por especialistas e define as regras que regulam a combinação entre termos e relações em um domínio de conhecimento e os usuários fórmulam consultas usando conceitos definidos pela ontologia buscando melhorias no processo de recuperação da informação.

Para RATHUNDE (2001 *apud* Feitosa, 2005) uma ontologia é uma especificação de uma conceituação, ou seja, uma descrição de conceitos e relações que existem em um domínio de interesse, basicamente objetivando restringir o número de possibilidades de interpretação do conceito dentro de um dado contexto, a partir de um

formalismo composto por conceitos e relações, suas definições, propriedades e restrições, descritas na forma de axiomas, sendo útil para apoiar a especificação e implementação de qualquer sistema de computação complexo através do seu mecanismo de representação do conhecimento que visa raciocinar sobre o domínio ao invés de agir diretamente sobre ele e pode ser desenvolvida para vários fins, mas, de uma forma geral, os seguintes propósitos são atingidos:

- Ajudar as pessoas a compreender melhor certa área de conhecimento, uma vez que no desenvolvimento de uma ontologia, as pessoas envolvidas no processo se vêem diante do desafio de explicar seu entendimento sobre o domínio em questão, o que as faz refletir e melhorar sua compreensão sobre esse domínio;
- Ajudar as pessoas a atingir um consenso no seu entendimento sobre uma área de conhecimento, pois diferentes especialistas têm entendimento diferenciado sobre os conceitos envolvidos que levam a problemas na comunicação e que, ao se construir uma ontologia, essas diferenças são explicitadas buscando-se um consenso sobre seu significado e sua importância;
- Ajudar outras pessoas a compreender certa área de conhecimento. Uma vez que existe uma ontologia sobre uma determinada área, uma pessoa que deseja aprender mais sobre essa área não precisa se reportar sempre a um especialista, podendo estudar a ontologia e aprender sobre o domínio em questão, absorvendo um conhecimento geral e de consenso (DUARTE, 2000).

De fato, todos esses propósitos estão sendo considerados para a construção de uma ontologia, cujo objetivo maior está relacionado ao entendimento deste complexo domínio.

Das várias definições apresentadas, utilizaremos neste trabalho a seguinte definição²¹: uma ontologia é uma descrição formal e explícita dos conceitos do domínio do discurso, contendo: as *classes*, que descrevem conceitos do domínio e que podem ter *subclasses* que representam conceitos mais específicos chamados subclasses em comparação a *superclasse*, e que se organizam em uma taxonomia hierárquica; os *slot's*, algumas vezes chamados de funções ou propriedades de cada conceito, descrevem várias características e atributos do conceito, as propriedades das classes e instâncias e os valores permitidos para eles; e as *facet's*, algumas vezes chamados de restrições nos *slot's*.

É importante notar que, apesar das diferentes noções sobre ontologia, (CHANDRASEKARAN e JOSEPHSON, 1999, p.22) apresenta um consenso sobre as mesmas nos seguintes aspectos:

- Existem objetos no mundo;
- Objetos têm propriedades ou atributos que podem ter valores;
- Podem existir várias relações entre os objetos;
- Propriedades e relações podem mudar com o tempo;
- Existem eventos que ocorrem em diferentes instantes de tempo;
- Existem processos nos quais os objetos participam e que ocorrem no tempo;
- O mundo e seus objetos podem ter diferentes estados;
- Eventos podem causar outros eventos ou estados como efeitos;
- Objetos podem ter partes.

²¹ Serão utilizadas como padrão letras minúsculas na definição de nomes de classes e *slot's*.

Nós então, definiremos instâncias individuais das classes da ontologia e preencheremos essas instâncias com informações específicas de valores ou restrições de cada *slot*, criando assim, uma base de conhecimento com certo grau de formalidade e classificada em determinado tipo.

2.5.3 Classificação e formalismo

Uma ontologia pode ser classificada em diferentes tipos (Falbo, 1998; Guarino, 1998; Studer *et al*, 1998), e algumas propostas definem tipos de ontologias relacionando-as à sua função, ao grau de formalismo, à sua aplicação, à sua estrutura e ao conteúdo da contextualização. Não temos aqui a pretensão de explorar todas elas, já que são bastante discutidas na literatura existente, cabendo apenas citar que diferentes autores definem tipos distintos, que analisados mais de perto, mostram-se similares.

Na visão de Uschold e Gruninger (1996), as ontologias podem ser classificadas em três categorias, quanto ao tipo de conhecimento que representam:

- **Ontologia de domínio**²²: conceituações de domínios particulares;
- **Ontologia de tarefas**: conceituações sobre a resolução de problemas independentemente do domínio onde ocorram;
- **Ontologia de representação**: conceituações que fundamentam os formalismos de representação do conhecimento.

Em Guarino (1995, 1998), é proposta outra abordagem, classificando as ontologias em níveis:

- **Ontologia genérica**: conceitos muito genéricos, independentes de um problema ou domínio particular;

²² **Domínio** é considerado um problema ou uma área de tarefa na qual muitas aplicações semelhantes serão desenvolvidas. (OLIVEIRA, 1999 *apud* RIOS, 2005)

- **Ontologia de domínio:** descrevem o vocabulário relativo a um domínio específico, através da especialização de conceitos presentes na ontologia de alto nível;
- **Ontologia de tarefa:** descreve o vocabulário pertinente a uma tarefa genérica ou específica através da especialização de conceitos presentes na ontologia de alto nível;
- **Ontologia de aplicação:** descreve conceitos dependentes do domínio e da tarefa particulares.

No âmbito do nosso projeto, e sob o ponto de vista de Uschold e Gruninger (1996) e Guarino (1995, 1998), nossa ontologia será classificada como de tarefa, unificando as conceituações sobre a resolução de problemas e descrevendo um vocabulário pertinente a uma tarefa genérica ou específica através da especialização de conceitos presentes na ontologia.

Para Uschold e Gruninger (1996, p. 6) o grau de formalidade com que o vocabulário é criado, com seus vários significados específicos apresenta-se em quatro formas diferentes:

- **Altamente informal:** quando expresso em linguagem natural de forma livre;
- **Semi-informal:** quando expresso em linguagem natural, mas de forma restrita e estruturada, reduzindo assim ambigüidades. Por exemplo, os conceitos descritos em '*Enterprise Ontology*';
- **Semi-formal:** quando expresso em uma formalidade artificial definida por uma linguagem. Por exemplo, a versão em Ontolingua de '*Enterprise Ontology*';

- **Rigorosamente formal:** quando os termos são meticulosamente definidos com uma semântica formal, teoremas e propriedades evidenciados por sua profundidade e completeza. Por exemplo, o projeto TOVE²³.

Aqui, cabe ressaltar que nossa ontologia será semi-informal, conforme o projeto ‘*Enterprise Ontology*’, mas sem o processo de formalização posterior realizado no mesmo, apresentando em linguagem natural um texto estruturado e restrito onde se buscará eliminar as ambigüidades encontradas.

2.5.4 Uso da ontologia

Segundo (SMITH, 1982 *apud* FALBO, 1998) uma ontologia é desenvolvida para satisfazer um dos seguintes propósitos:

- Permitir que múltiplos agentes compartilhem seu conhecimento;
- ajudar as pessoas a compreenderem melhor uma certa área de conhecimento;
- Ajudar pessoas a atingir um consenso no seu entendimento sobre uma área de conhecimento.

Noy e McGuinness (2001, p.1) nos pergunta: “Porque alguém desejaria desenvolver uma ontologia?”, e nos apresenta algumas razões como resposta:

- **Para o compartilhamento de um entendimento comum da estrutura da informação entre pessoas e agentes de *software*:** citando (MUSEN, 1992; GRUBER, 1993), que apresenta esta razão como um dos mais comuns objetivos no desenvolvimento de ontologias. Por exemplo, suponha diferentes sites da *Web* contendo informações médicas ou provendo um *e-commerce* de serviços médicos. Se estes sites compartilharem e publicarem uma mesma base ontológica dos termos que utilizam, agentes de *software*

²³ *TO*ronto *Virtual Enterprise*.

poderão extrair e agregar informações desses diferentes sites, que poderão responder a diversas questões de usuários ou adicionar dados em outras aplicações.

- **Para possibilitar a reutilização do domínio de conhecimento:** é a recente força que conduz por traz das pesquisas ontológicas. Por exemplo, modelos em variados domínios diferentes precisam representar a noção de tempo. Esta representação inclui noções de intervalo de tempo, passagem do tempo, medições de tempo, etc. se um grupo de pesquisadores desenvolve tal ontologia em detalhes, outros podem simplesmente reutilizá-la para seus domínios. Ainda, se nós precisamos construir uma grande ontologia, nós podemos integrar várias ontologias que descrevem partes do grande domínio e também podemos reutilizar ontologias gerais, e estendê-las para descrever nosso domínio de interesse.
- **Para tornar explícito a compreensão do domínio:** facilita a implementação de possíveis mudanças quando nosso conhecimento sobre o domínio mudar também. Por exemplo, os códigos de difícil compreensão das linguagens de programação não somente são difíceis de entender, como também, difíceis de alterar, principalmente para um programador com pouca experiência. Além disso, explícitas especificações do domínio de conhecimento são úteis para novos usuários que devem estudar o significado dos termos do domínio.
- **Para separar o domínio de conhecimento do domínio operacional:** o que é outro comum uso das ontologias. Por exemplo, nós podemos descrever a tarefa de configurar um produto desde seus componentes de acordo com especificações requeridas e implementar um programa que faça esta configuração independente dos produtos e componentes por si mesmos, conforme cita (McGuinness e Wright, 1998). Nós podemos desenvolver uma ontologia dos componentes do computador e suas características e aplicar um algoritmo para criar computadores feitos sob medida. Nós podemos ainda, usar o mesmo algoritmo para configurar elevadores se mantivermos uma ontologia de componentes de elevadores para isto.

- **Para analisar um domínio de conhecimento:** o que é possível, uma vez que a especificação dos termos esteja disponível. Uma análise formal dos termos é extremamente valiosa quando tentamos reutilizar ou estender ontologias já existentes (conforme cita McGuinness *et al*, 2000). **Freqüentemente a ontologia do domínio não é o objetivo por si mesma**²⁴. Desenvolver uma ontologia é semelhante a definir um conjunto de dados e suas estruturas para outros programas utilizarem. Métodos para resolução de problemas, aplicações independentes de domínio e agentes de *software* usam ontologias e bases de conhecimento construídas de dados ontológicos. Por exemplo, uma ontologia sobre vinhos e comidas e uma apropriada combinação entre elas, pode ser utilizada como base para algumas aplicações em uma ferramenta de gerenciamento de restaurantes: uma aplicação pode criar sugestões para o cardápio do dia ou responder questões de garçons e clientes. Outras aplicações podem analisar um inventario da adega e sugerir quais as categorias de vinho deve-se expandir e quais vinhos devem-se adquirir para o próximo cardápio ou livro de receitas.

A rica literatura existente parece concordar que um dos maiores propósitos da ontologia é seu reuso, seja como um vocabulário, uma especificação ou uma teoria lógica, sua utilização compartilhando entendimento entre pequenos grupos ou grandes comunidades, estrutura seus significados em bases de conhecimento, carregando suas sentenças lógicas e integrando modelos de diferentes domínios em coerentes *framework's*.

Nesse contexto, outra importante motivação para o uso de ontologias é o seu uso em processos da re-engenharia (onde se precisa de um modelo integrado da empresa, seus processos, sua organização, objetivos, e seus clientes) e em arquiteturas de multi-agentes distribuídos (onde diferentes agentes se comunicam e solucionam problemas).

²⁴ Grifo nosso.

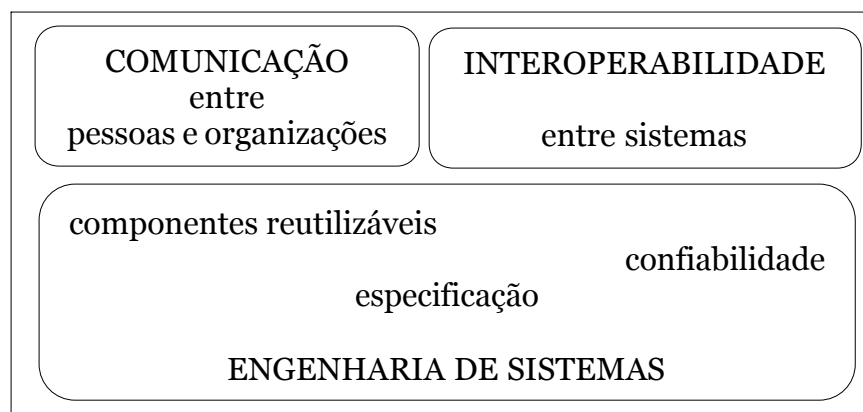


Figura 29 Usos para ontologias.

Fonte: Uschold e Gruninger (1996, p.7).

Uschold e Gruninger (1996, p. 7-14) subdividem o uso da ontologia em três categorias, conforme figura 29 acima.

2.5.4.1 Comunicação

Como a ontologia reduz a confusão conceitual e terminológica permitindo com isso um entendimento compartilhado e uma comunicação entre pessoas com diferentes necessidades, contextos e pontos de vista, alguns aspectos são considerados:

- **Modelos normativos:** com ela, podemos construir modelos que criem semântica para os sistemas, que sejam extensíveis e que possam ser refinados, permitindo assim, transformações semânticas entre diferentes contextos, compartilhando entendimento entre os mais variados sistemas integrados.
- **Redes de relacionamentos:** com ela, podemos registrar o que foi explorado e navegado na rede, identificando conexões lógicas entre os elementos dos modelos de sistemas, que é particularmente importante em aplicações que requerem o uso de múltiplas ontologias de diferentes domínios.
- **Consistência e não ambigüidade:** com ela, podemos prover definições não ambíguas para os termos usados pelos sistemas, mantendo a consistência

entre eles e as ontologias. Como os usuários de ontologias são diferentes das ferramentas de suporte, podemos criar ambientes, identificando sinônimos e/ou traduções de termos que possam representar diferentes significados para os diferentes tipos de pessoas.

- **Integrando diferentes perspectivas dos usuários:** com ela, podemos prover a integração de diferentes perspectivas, identificando as diferenças entre elas, através da compreensão do domínio de conhecimento compartilhado, que se tornou vital em um sistema onde se utilizam agentes para se comunicar. Por exemplo, pessoas em diferentes posições em uma organização terão diferentes perspectivas sobre o que faz a organização, que objetivos alcançar, e como estes objetivos devem ser alcançados. Uma ontologia estabelece então, os fundamentos para o desenvolvimento de padrões dentro de uma organização, que possa ser compartilhado e que todos os participantes usem essa terminologia padronizada em função dos objetivos e das relações da sua organização.

2.5.4.2 Interoperabilidade

A principal preocupação no uso de ontologias em domínios, como a modelagem de organizações e arquiteturas multi-agentes, é a criação de um ambiente integrador para diferentes ferramentas de *software*, que usam essas ontologias para lidar com diferentes usuários e com problemas de interoperabilidade na troca de dados.

- **Ontologias como Inter-Língua:** com ela, podemos prover a qualquer ambiente de tecnologia de informação para a re-engenharia de processos de negócios ou para sistemas multi-agentes uma descrição integrada do seu modelo organizacional para assim poder especificar suas atividades, recursos, objetivos, produtos, e serviços. Esta descrição serve como um repositório comum e acessível para várias ferramentas de *software* do ambiente, que também pode servir para integrar repositórios de dados existentes, seja pela padronização dos termos usados nestes repositórios, ou

pelo uso de tradutores que permitam a interoperabilidade entre eles e sua comunicação com os usuários.

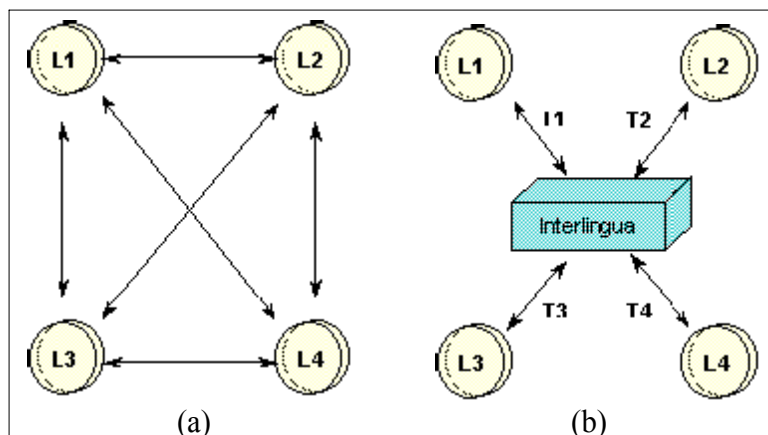


Figura 30 Ontologia como Inter-Lingua
 Fonte: Uschold e Gruninger, 1996.

Para facilitar essa interoperabilidade, ontologias devem ser usadas para suportar a tradução entre diferentes linguagens e representações. Uma possibilidade seria projetar um único tradutor para cada par de línguas que se quer fazer comunicar; porém, isto exigiria $O(n^2)$ traduções para n linguagens diferentes, como é mostrado na Figura 30(a). Usando uma ontologia como interlíngua para suportar a tradução, pode-se reduzir o número de traduções a $O(n)$ para n linguagens diferentes, isto iria requer somente traduções para uma ontologia nativa, como mostrada na Figura 30(b).

- **Dimensão da Interoperabilidade:**

Deve-se considerar a natureza das relações entre os usuários que compartilham as ferramentas e os dados, pois ela é vital para que as ontologias e ferramentas que são usadas dentro da mesma organização sejam compartilháveis e re-utilizáveis:

- ✓ **Inter-operabilidade interna:** Esta característica exige que todos os subsistemas interoperem e, desta forma, é necessário um controle direto da unidade organizacional;
- ✓ **Inter-operabilidade externa:** Esta característica é necessária quando uma unidade organizacional precisa comunicar com outros sistemas, procedentes do ambiente externo;

- ✓ **Ontologias integradas entre os domínios:** Outra diferença para interoperabilidade surge do assunto da integração de ontologias a partir de diferentes domínios para suportar tarefas. Por exemplo, uma ontologia para suportar sistemas *workflow*, necessitariam de ontologias para os processos, recursos, produtos, serviços e a organização. Então, as diversas ferramentas workflow iriam usar este conjunto de ontologias integradas;
- ✓ **Ontologias que integram as ferramentas de *software*:** De outro lado, pode-se necessitar diferentes ontologias no mesmo domínio devido à natureza do sistema. Por exemplo, diferentes ferramentas podem querer usar diferentes ontologias relacionadas a processos; para alcançar a interoperabilidade, precisa-se ter uma ontologia comum que ambas ferramentas possam usar. Este é o desafio mais dificultoso a enfrentar no uso de ontologias, visto que usualmente não é possível impor requisitos de integração sobre as ferramentas.

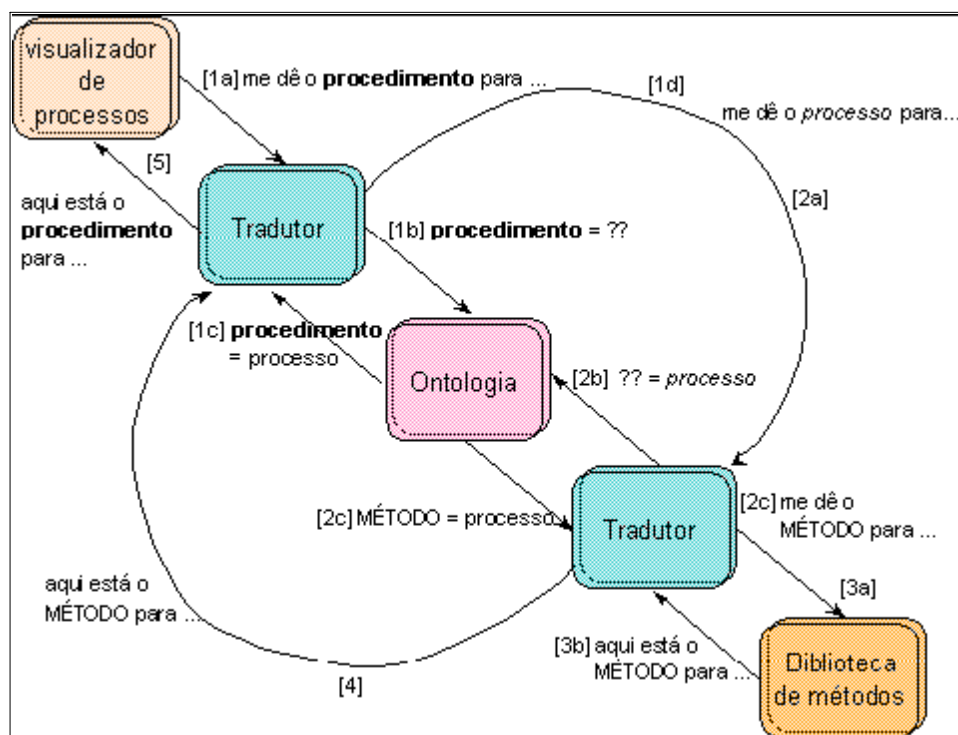


Figura 31 Exemplo de ontologia como inter-língua.
Fonte: Uschold e Gruninger, 1996.

A Figura 31 apresenta o uso de ontologias como uma interlíngua para integrar diferentes ferramentas de *software*. O termo PROCEDIMENTO é usado em uma ferramenta que é traduzida em outro termo, MÉTODO é usado por outras ferramentas por meio da ontologia, os termos são considerados os mesmos e dentro do conceito processo.

2.5.4.3 Engenharia de sistemas

As ontologias podem ser aplicadas para auxiliar a especificação e o desenvolvimento de sistemas de *software*:

- **Especificação:** o entendimento compartilhado dos problemas e tarefas pode ajudar no processo de identificação de requisitos e definição de uma especificação para um sistema de tecnologia de informação. Isto é especialmente vantajoso quando os requisitos envolvem diferentes grupos usando diferentes terminologias no mesmo domínio, ou múltiplos domínios.
- **Confiabilidade:** Uma representação formal também torna possível uma automatização na verificação e consistência do *software* com respeito à especificação das declarações.
- **Reutilização:** A compreensão compartilhada é a base para a especificação formal de entidades importantes, atributos, processos e suas inter-relações no domínio de interesse. Esta representação formal pode ser a alavanca para reutilização de componentes compartilhados em um sistema de *software*.

2.5.5 Metodologias

Fernández-López (1999, p. 4-1) explana que cada grupo de pesquisadores da engenharia de ontologias, que é uma disciplina relativamente nova e imatura, emprega sua própria metodologia, e que, é importante conhecer essas metodologias e quais os

problemas que ainda precisam ser resolvidos antes delas serem consideradas maduras e poderem ser aplicadas com o sucesso esperado.

Atualmente, os autores concordam com a imaturidade das metodologias desenvolvidas, o que é constatado pelo fato de ainda não serem amplamente aceitas, fazendo com que novas metodologias sejam ainda criadas pelos grupos de pesquisadores.

Existem alguns grupos de pesquisa que estão propondo uma série de passos e metodologias para o desenvolvimento de ontologias. Porém, esta tendência está ainda dando seus primeiros passos, no sentido de que o que acontece na maioria dos casos é a adoção de uma metodologia própria desenvolvida por cada grupo de pesquisa da área de ontologia. Noy e McGuinness (2001) argumentam que não há uma maneira ou metodologia “correta” para desenvolver ontologias, e apresentam três papéis fundamentais no projeto das mesmas:

- Não existe uma maneira correta de modelar um domínio, sempre existem alternativas viáveis. A melhor solução quase sempre depende da aplicação;
- O desenvolvimento de uma ontologia é necessariamente um processo iterativo;
- Conceitos em uma ontologia deveriam ser próximos dos objetos (físicos ou lógicos) e relacionamentos do domínio de interesse. Os conceitos, então, deveriam ser provavelmente substantivos (objetos) ou verbos (relacionamentos) em sentenças que descrevem o domínio.

Gruber (1993, p.2-3) nos apresenta um conjunto de critérios utilizados na modelagem de ontologias como uma proposta preliminar com o propósito de compartilhar conhecimento e interoperabilidade entre programas baseados em uma conceitualização compartilhada:

- **Clareza:** definições devem ser objetivas. Sempre que for possível, uma definição deve ser declarada através de axiomas lógicos, sendo preferível

que seja completa, com condições necessárias e suficientes, melhor que uma definição parcial, definida apenas por condições necessárias ou suficientes. Todas as definições devem ser documentadas com linguagem natural;

- **Coerência:** caso uma sentença inferida a partir de axiomas, contradiga uma definição ou exemplo dado informalmente, então a ontologia é incoerente;
- **Extensível:** uma ontologia deve permitir que novos termos possam ser definidos para usos especiais baseados no vocabulário existente, de maneira que não seja requerida a revisão das definições previamente existentes;
- **Mínimo compromisso com implementação:** a conceituação deve ser especificada no nível do conhecimento, isto é, sem depender de uma codificação particular no nível simbólico ou de codificação;
- **Mínimo compromisso com ontologia:** uma ontologia deve requerer o compromisso ontológico mínimo suficiente para dar suporte às atividades de compartilhamento de conhecimento desejadas.

Fernández-López (1999, p.4-3) apresenta alguns critérios que utilizou para analisar algumas metodologias, os quais embasam a seleção da mesma nesse projeto, conforme segue:

- **Herança da engenharia do conhecimento:** Considerando a influência dela na metodologia em questão;
- **Detalhamento da metodologia:** Considerando as atividades e técnicas precisamente especificadas na proposta da metodologia.
- **Recomendações para o formalismo do conhecimento:** Considerando o formalismo proposto para representar o conhecimento (lógica, *frames*, etc).
- **Estratégia para a construção da ontologia:** discutindo quais das seguintes estratégias são usadas no desenvolvimento da ontologia:

- ✓ **Dependente da aplicação**, onde a ontologia é construída baseada na aplicação de base de conhecimento, pelo significado do processo de abstração;
 - ✓ **Semi-dependente da aplicação**, onde possíveis cenários da ontologia são usados para identificar o estágio de especificação;
 - ✓ **Independente da aplicação**, onde o processo é totalmente independente do uso que a ontologia usará, seja base de conhecimentos, agentes, etc.
- **Estratégia de identificação de conceitos**: São consideradas as estratégias:
- ✓ **bottom-up**: segundo Uschold e Gruninger (1996), o processo de desenvolvimento inicia-se na definição mais específica das classes, deixando a hierarquia com o subsequente agrupamento dessas classes dentro dos conceitos mais gerais. Por exemplo, iniciamos criando as classes, e em seguida, criamos as superclasses comuns às classes já criadas.
 - ✓ **top-down**: segundo Uschold e Gruninger (1996), o processo de desenvolvimento inicia-se na definição mais geral dos conceitos do domínio e subsequentemente especializações desses conceitos. Por exemplo, iniciamos criando as classes, e em seguida, especializamos e criamos as subclasses dessas classes já criadas, podendo ainda, mais adiante, categorizá-las.
 - ✓ **middle-out** ou **combination**: os conceitos chave são estabelecidos buscando-se inicialmente os conceitos mais importantes, preferencialmente os mais gerais ou mais específicos. Os outros conceitos são obtidos por generalização ou especialização.

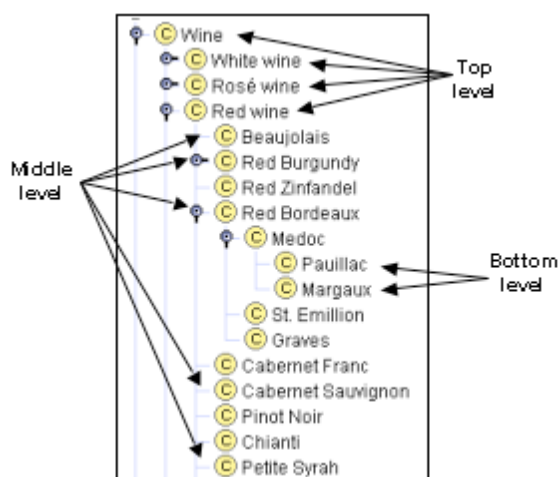


Figura 32 Diferentes níveis da taxonomia do vinho
 Fonte: Noy e McGuinness (2001)

Para Noy e McGuinness (2001, p.7), nenhum desses três métodos é melhor que o outro, sua utilização depende fortemente da visão pessoal do domínio. Se o desenvolvedor tem uma sistemática visão *top-down* do domínio, então pode ser mais fácil utilizar a estratégia *top-down*, ou seja, se tendemos a pensar nos objetos do domínio pela distinção mais geral primeiro, a estratégia *top-down* pode ser mais apropriada, mas se temos exemplos específicos, a estratégia *bottom-up* também pode ser mais apropriada.

2.5.5.1 Uschold e King

Esta metodologia, criada em 1995 é baseada na experiência do desenvolvimento da “Enterprise Ontology”, desenvolvida para o processo de modelagem empresarial, e esta dividida nas seguintes fases:

- **Identificação do propósito:** onde é importante deixar claro porque a ontologia esta sendo construída e qual a sua intenção de uso;
- **Construção da ontologia:** que está dividida em três partes:
 - ✓ **Captura da ontologia,** utilizando a estratégia *middle-out*:

- Identificação dos conceitos chave e relacionamentos do domínio de interesse, isto é, seu escopo;
 - Produção de um preciso texto não ambíguo para cada conceito e relacionamento;
 - Identificação dos termos que se referem a cada conceito e relacionamento.
- ✓ **Codificação:** envolve a explícita representação do conhecimento adquirido na fase de captura na forma de uma linguagem formal;
- ✓ **Integração a ontologias existentes:** integração a ontologias já existentes.
- **Avaliação:** onde os autores adotam a definição de Gómez-Pérez *et al* (1995): “para fazer um julgamento técnico das ontologias, o ambiente de desenvolvimento e a documentação a respeito da estrutura de referência [...] que pode ser, especificação de requerimentos, questões de competência e/ou o mundo real”
 - **Documentação:** recomenda as linhas gerais para o estabelecimento da documentação, de acordo com o tipo e propósito da ontologia.

O mais importante projeto desenvolvido usando esta metodologia é a *The Enterprise Ontology*²⁵, a qual é um conjunto de termos e definições relevantes para negócios empresariais. Foi desenvolvida sob um projeto empresarial por uma aplicação em Inteligência Artificial pela Universidade de *Edinburgh* com os parceiros: IBM, *Lloyd's Register*, *Logica UK Limited* e Unilever.

²⁵ <http://www.aiai.ed.ac.uk/project/enterprise/enterprise/ontology.html>

2.5.5.2 Grüninger e Fox

Esta metodologia, criada em 1995 é baseada na experiência do desenvolvimento do “*TOVE project ontology*”, desenvolvida para modelagem dos processos e atividades de negócios, que envolve, essencialmente, a construção de um modelo lógico de conhecimento que é especificado pela ontologia. Primeiramente, é feita uma descrição informal das especificações que dará origem a construção de um modelo formal do conhecimento, e esta dividida nas seguintes fases:

- **Captura de cenários:** é motivada por cenários específicos para aplicações, que buscam a resolução de problemas históricos, onde outras ontologias não se adequaram, mas que apresenta um conjunto intuitivo de possíveis soluções;
- **Fórmulação de questões de competência informal:** baseadas nos cenários obtidos, essas questões expressam requerimentos usando axiomas e definições, capazes de representar através da terminologia, respostas gerais e operações de composição ou decomposição. As questões de competência são usadas para avaliar o comprometimento ontológico que se encontra nos requerimentos.
- **Especificação da terminologia da ontologia a partir de uma linguagem formal:** que está dividida em duas partes:
 - ✓ **Obtendo a terminologia informal:** uma vez que as questões de competência informal estão disponíveis, um conjunto de termos usados pode ser extraído dessas questões, que servirão de base para a especificação da terminologia em linguagem formal;
 - ✓ **Especificação da terminologia formal:** uma vez que as questões de competência informal tenham sido colocadas para propor uma nova

ontologia ou a extensão de uma já existente, a terminologia é então especificada utilizando-se um formalismo, como, por exemplo, o KIF²⁶.

- **Fórmulação de questões de competência formal utilizando a terminologia da ontologia:** uma vez que as questões de competência tenham sido colocadas informalmente e a terminologia da ontologia tenha sido definida, as questões de competência são definidas formalmente;
- **Especificação de axiomas e definições para os termos da ontologia com a linguagem formal:** os axiomas da ontologia especificam as definições dos termos e restringem suas interpretações, são definidos em sentenças de lógica de primeira ordem para os objetos da ontologia e devem prover a definição da semântica ou significado desses termos. Se os axiomas propostos não forem suficientes para representar as questões de competência formal e caracterizar as soluções das questões, objetos ou outros axiomas devem ser adicionados até que sejam suficientes. O desenvolvimento desses axiomas é um processo iterativo.
- **Estabelecer condições para caracterizar a completeza da ontologia:** definir condições para que as questões de competência sejam completas.

Esta metodologia foi utilizada para construir *The TOVE project*²⁷, desenvolvido no Laboratório de integração empresarial da Universidade de Toronto, constitui-se de um modelo integrado e formalizado de senso comum empresarial usando a lógica de primeira ordem, ela inclui: *Enterprise Design Ontology*, *Project Ontology*, *Scheduling Ontology* e *Service Ontology*.

2.5.5.3 Amaya Berneras *et al.*

Esta metodologia, criada em 1996 é baseada na experiência do desenvolvimento do “*KACTUS project*”, que procura investigar a possibilidade de reutilização do

²⁶ Knowledge Interchange Format.

²⁷ <http://www.eil.utoronto.ca/enterprise-modelling/tove/index.html>

conhecimento em sistemas técnicos complexos e as funções da ontologia neles. O desenvolvimento da ontologia esta condicionado ao desenvolvimento da aplicação, onde, toda vez que a aplicação é construída, a ontologia que representa o conhecimento requerido para a aplicação é construído, e esta dividida nas seguintes fases:

- **Especificação da aplicação:** fornece o contexto da aplicação e a visão dos componentes que a aplicação adota como modelo;
- **Esboço preliminar baseado em categorias ontológicas relevantes de alto-nível:** em uma fase anterior, é desenvolvida uma lista de termos e tarefas utilizada como entrada para alcançar várias visões do modelo global de acordo com determinadas categorias ontológicas de alto-nível. Este processo envolve a busca de ontologias desenvolvidas para outras aplicações, as quais são refinadas e estendidas para serem usadas em uma nova aplicação.
- **Refinamento e estruturação da ontologia:** para se chegar a um esboço definitivo, devemos tentar assegurar que os modelos sejam o mais coerente possível, procurando chegar a máxima uniformidade dentro de cada módulo.

Esta metodologia foi utilizada para construir três ontologias, como resultado do desenvolvimento do mesmo número de aplicações. A primeira para diagnosticar falha na rede elétrica, a segunda para agendar a continuação de serviços depois da falha na rede elétrica e a terceira para controlar a rede elétrica baseado nas duas aplicações anteriores.

2.5.5.4 Fernández-López *et al.*

Esta metodologia, criada em 1996 foi desenvolvida pelo Laboratório de Inteligência Artificial da Universidade Politécnica de Madri, baseada na experiência do desenvolvimento do “*METHONTOLOGY*”, cujo *framework* habilita a construção de ontologias no nível de conhecimento, e permite a identificação do processo de

desenvolvimento, o ciclo de vida envolvendo protótipos e técnicas particulares para se executar cada atividade, e esta dividida nas seguintes fases:

- **Atividades de gerenciamento do projeto:** inclui planejamento, controle e garantia de qualidade. No *planejamento*, identificam-se quais tarefas serão executadas, como elas serão organizadas, por quanto tempo e quais os recursos são necessários para sua finalização, e esta atividade é essencial para ontologias que precisam usar outras já construídas, ou que requerem níveis de abstração ou generalização. No *Controle*, garantimos que as tarefas planejadas sejam finalizadas da maneira que elas foram programadas a ser. E, finalmente, na *garantia de qualidade*, asseguramos que a qualidade de cada produto desenvolvido (*ontologia, software e documentação*) é satisfatória.

- **Atividades orientadas ao desenvolvimento:** inclui especificação, conceitualização, formalização e implementação. A *especificação* expressa porque a ontologia esta sendo construída e quais as suas intenções de uso e quem serão seus usuários finais. A *conceitualização* estrutura o domínio de conhecimento como um modelo signficante no nível de conhecimento. A *formalização* transforma o modelo conceitual em um modelo formal ou semi-computável. A *implementação* constrói modelos computáveis em linguagem computacional. Finalmente, a *manutenção* atualiza e corrige a ontologia. FERNÁNDEZ-LÓPEZ (1999), apresenta detalhes de como executar as atividades de desenvolvimento, com exceção a formalização e manutenção.

- **Atividades de suporte:** incluem uma série de atividades, executadas ao mesmo tempo com as atividades orientadas ao desenvolvimento, sem os quais a ontologia não pode ser construída, incluindo aquisição do conhecimento, avaliação, integração, documentação e gerenciamento de sua configuração. Na *aquisição do conhecimento*, adquire-se conhecimento sobre dado domínio. A *avaliação* cria um julgamento técnico das ontologias, seu ambiente de *software* associado e a documentação com respeito à estrutura de referencia durante cada fase e entre as fases do ciclo de vida. A

integração de ontologias é requerida quando construímos uma nova ontologia reutilizando outras ontologias que já estejam disponíveis. Na *documentação*, apresenta-se clara e exaustivamente os detalhes de cada uma e de todas as fases e produtos gerados. O *gerenciamento de sua configuração* registra todas as versões da documentação, *software* e códigos da ontologia para controlar suas alterações.

Esta metodologia foi utilizada para construir *(Onto)²Agent*²⁸ que usa a *reference-ontology* como fonte de conhecimento e recuperação de descrições de ontologias que satisfaçam dado conjunto de restrições, *Chemical OntoAgent* para o domínio de elementos químicos e estruturas cristalinas, *Ontogeneration* gera textos em espanhol em resposta as perguntas dos alunos no domínio da química, *Environment pollutants ontologies* representando métodos de detecção de diferentes componentes da poluição sob vários meios: água, ar, terra, etc, *The reference-ontology* contém conhecimento sobre a comunidade científica no campo de aquisição de conhecimento, particularmente: cientistas, tópicos de pesquisas, projetos, universidades, etc. Foi proposta para construção de ontologias pela *Foundation for Intelligent Physical Agents*²⁹ (FIPA) que promove interoperabilidade entre aplicações baseadas em agentes.

2.5.5.5 Swartout et al.

Esta metodologia, criada em 1997 foi desenvolvida pelo ISI (*Information Sciences Institute*) para uso no processamento de linguagem natural, com mais de 50.000 conceitos organizados em uma hierarquia de acordo com o nível de abstração, e está dividida nas seguintes fases:

- Uma série de termos é tomada como origem;
- Esses termos são inseridos na base do SENSUS;

²⁸ <http://delicias.dia.fi.upm.es/OntoAgent>

²⁹ <http://www.fipa.org>

- Termos que podem ser relevantes dentro do domínio e ainda não inseridos são adicionados;
- Finalmente, um passo feito manualmente e que requer algum entendimento do domínio é a verificação da relevância dos nós por todo o caminho da árvore e da sub-árvore.

Uma metodologia para o planejamento de uma campanha militar aérea foi construída utilizando SENSUS, contendo uma visão geral dos elementos básicos que caracterizam os planos de uma campanha aérea: cenários, participantes, comandantes, etc. Esta ontologia inclui armas, sistemas em geral, combustíveis, etc.

2.5.6 Roteiro para a construção da ontologia

Existem na literatura propostas de metodologias para a construção de ontologias com abordagens e características diversas, as quais muitas vezes foram desenvolvidas com diferentes objetivos e para aplicações distintas. Para selecionar a metodologia mais adequada, seria necessário avaliar ontologias resultantes da aplicação de cada metodologia, uma tarefa que vai além dos objetivos desta pesquisa. Outra opção seria pesquisar sobre a maturidade das principais metodologias. Nesse sentido, optou-se por adotar um roteiro baseado nos estudos de (NOY; MCGUINNES, 2001) e (USCHOULD; KING, 2001), cujas etapas são apresentadas a seguir:

A metodologia de Noy e McGuinness (2001), criada em 1995, é baseada na experiência do desenvolvimento do “*TOVE project ontology*” para modelagem dos processos e atividades de negócios, que envolve, essencialmente, a construção de um modelo lógico de conhecimento que é especificado pela ontologia, onde, inicialmente foi feita uma descrição informal das especificações que deram origem a construção de um modelo formal do conhecimento, conforme foi utilizado no nosso projeto.

Seguindo os passos de Noy e McGuinness (2001, p.4) iniciamos enfatizando algumas regras fundamentais no projeto da ontologia, que serão referidos nesse trabalho

diversas vezes, apesar de parecerem dogmáticas, ajudarão à tomada de decisões no projeto em questão:

- Não existe uma forma correta de se modelar um domínio, mas sempre existirão alternativas viáveis. A melhor solução quase sempre depende da aplicação que se tem em mente e a extensão que se deseja alcançar;
- O desenvolvimento de uma ontologia necessariamente é um processo iterativo;
- Conceitos em uma ontologia podem se referir a objetos físicos ou lógicos, e relacionamentos em seu domínio de interesse. Eles provavelmente serão substantivos para os objetos e verbos para as sentenças que descrevem o domínio.

A metodologia de Uschold e King (2001), criada em 1995, é baseada na experiência do desenvolvimento da “*The Enterprise Ontology*”, com um conjunto de termos e definições relevantes para negócios empresariais. Foi desenvolvida pela Universidade de *Edinburgh* com os parceiros: IBM, *Lloyd’s Register*, *Logica UK Limited* e Unilever.

As etapas de elaboração de ontologias descritas a seguir tomam por base estes dois trabalhos mencionados, sintetizando os diferentes aspectos a serem observados no desenvolvimento de ontologias.

2.5.6.1 Determinação do domínio e escopo da ontologia

Um dos objetivos de uma ontologia é obter uma conceitualização consensual de um domínio do conhecimento, de forma a incrementar a recuperação da informação. Nesse sentido, deve-se definir o que se pretende com a ontologia a ser construída, ou seja, que tipo de respostas a estrutura deve ser capaz de fornecer ao usuário.

Segundo Noy e McGuinness (2001), inicia-se o desenvolvimento da ontologia pela definição do domínio e seu escopo. Para isso, respondem-se algumas questões básicas, como:

- O que o domínio da ontologia irá cobrir?
- Para qual finalidade nós usaremos a ontologia?
- Para quais tipos de questões de informação a ontologia irá prover respostas?
- Quem irá usar ou manter a ontologia?

As respostas dessas questões não mudarão durante o processo de desenvolvimento da ontologia, como o autor citou, mas elas limitarão sim o escopo do modelo.

Se a ontologia a ser desenhada for utilizada para ajudar o processamento em linguagem natural, pode ser importante a inclusão de sinônimos e outros tipos de palavras para os conceitos da ontologia.

Se as pessoas irão manter ou utilizar a ontologia para descrever o domínio em línguas diferentes, precisamos prover um mapeamento entre essas línguas.

Para Uschold e King (2001), a etapa inicial é a identificação do propósito, onde é importante deixar claro porque a ontologia está sendo construída e qual a sua intenção de uso.

Vários autores têm apresentado estudos para categorizar as ontologias de acordo com seus objetivos. Uma ontologia bibliográfica, por exemplo, tem com objetivo relacionar os tipos de dados que podem representar partes de uma referência bibliográfica. Tal ontologia é usada para organizar os conceitos e deve informar sobre as relações existentes entre referências bibliográficas, por exemplo: que trabalhos de um autor estão nas fontes consultadas? Qual a relação de um trabalho com outros artigos presentes em outras bases? Quantos artigos autores publicaram nos últimos cinco anos?

Quantas são as citações referentes ao autor? Quantas citações tiveram um artigo nos últimos X anos? Em que local em que se produzem mais publicações? (ALMEIDA, 2002).

A ontologia criada neste trabalho foi parcialmente classificada como uma ontologia de domínio, enquanto definia conceitos mais abrangentes dentro do seu escopo e de tarefa, pois seu objetivo está relacionado ao fornecimento de um vocabulário de conceitos desse domínio, sobre seus relacionamentos, sobre as atividades e sobre princípios que os governam (JASPER; USCHOLD, 1999).

Lembrando Uschold (1996), que as ontologias de tarefa representam as conceituações sobre a resolução de problemas independentemente do domínio onde ocorram.

Já Guarino (1994, 1997), propõe a ontologia de tarefa como à descrição de um vocabulário pertinente a uma tarefa genérica ou específica através da especialização de conceitos presentes na ontologia de alto nível.

Além da classificação, nossa ontologia, seguindo Uschold e Gruninger (1996, p. 6) apresenta o grau de formalidade semi-formal, onde o vocabulário criado foi expresso em uma formalidade artificial definida por uma linguagem, como por exemplo, a versão em Ontolingua de ‘*Enterprise Ontology*’ e no nosso caso, o *OWL* no *Protégé*.

Ainda segundo Noy e McGuinness (2001), uma das formas de se determinar o escopo da ontologia é esboçar uma lista de questões, chamadas questões de competência (Gruninger e Fox, 1995), que a ontologia utilizando uma base de conhecimento será capaz de responder. Essas questões servirão como teste depois: A ontologia contém informações suficientes para responder estes tipos de questões? As respostas requerem um particular nível de detalhamento ou representação de uma área em particular? Essas questões de competência são adequadamente esboçadas e não precisam ser exaustivas.

2.5.6.2 Pesquisa sobre ontologias e outras estruturas existentes no domínio

Noy e McGuinness (2001) nos informam que é importante considerarmos e checar se existe ontologia que possa ser utilizada, refinada ou estendida para o nosso domínio ou tarefa em particular.

Reutilizar ontologias pode ser requerido se nosso sistema precisar interagir com outras aplicações que já estejam comprometidas com uma ontologia particular ou com um vocabulário controlado.

Muitas ontologias já estão disponíveis em meios eletrônicos e podem ser importadas no ambiente de desenvolvimento ontológico que estivermos utilizando. O formalismo no qual a ontologia está representada frequentemente não importa, desde que muitos sistemas de representação de conhecimento possam importar e exportar ontologias. Mesmo que um sistema de representação de conhecimento não trabalhe diretamente com um formalismo em particular, traduzir a ontologia de um formalismo para outro não é tarefa difícil.

Existem inúmeras bibliotecas de ontologias disponíveis, como *Ontolingua ontology library*³⁰, *DAML Ontology Library*³¹, bem como as ontologias comerciais, por exemplo, *UNSPSC*³², *RosettaNet*³³, *DMOZ*³⁴, entre outras.

2.5.6.3 Enumerar os termos do domínio

Nesta fase é elaborada uma lista abrangente de termos relacionados ao assunto, sem preocupação quanto a ambigüidade entre conceitos, as relações entre eles ou qualquer propriedade que eles possam ter, ou ainda, se os conceitos são classes ou *slots* (NOY e MCGUINNESS, 2001).

³⁰ <http://www.ksl.stanford.edu/software/ontolingua/>

³¹ <http://www.daml.org/ontologies/>

³² <http://www.unspsc.org>

³³ <http://www.rosettanel.org>

³⁴ <http://www.dmoz.org>

É extremamente útil escrevermos uma lista dos termos que consideramos mais importantes e que gostaríamos de relatar sobre, ou explicar ao usuário. Sobre quais termos que gostaríamos de falar? Quais propriedades esses termos têm? O que nós gostaríamos de dizer sobre esses termos? (NOY e MCGUINNESS, 2001)

Busca-se nesse momento obter uma “grande lista” composta de possíveis termos da nova ontologia, obtidos da forma acima citada.

2.5.6.4 Definição das classes, hierarquia e propriedades.

Nesta fase, os termos mais frequentes resultantes da etapa anterior serão colocados em hierarquias, subordinando dos mais genéricos os termos mais específicos. A lista confeccionada procurará identificar aqueles termos que descrevem objetos e têm existência independente uns dos outros. Estes termos são candidatos a se tornar classes da ontologia, as quais são as principais referências na hierarquia.

Para isso, pode ser utilizada uma combinação das estratégias *top-down* e *bottom-up*, de acordo com o apresentado em (NOY e MCGUINNESS, 2001). Isto é, os principais conceitos são definidos primeiro e em seguida, refinados e/ou generalizados, para a definição das demais classes.

Em seguida, organizam-se as classes em uma taxonomia fazendo a seguinte pergunta: sendo instância de uma classe, a entidade será necessariamente uma instância para outras classes? Segundo (NOY; GUINNESS, 2001, p. 6), “se a classe A é uma superclasse da classe B, então cada instância de B é também uma instância de A”.

Termos da lista também podem ser descartados, quando necessário. Nem sempre apenas as classes possibilitam respostas às perguntas que a ontologia deve responder, conforme definido em seu escopo. Termos da lista que não são utilizados na definição das classes, podem ser propriedades dessas classes e chamados de propriedades, relações ou *slots*.

Cada classe possui algumas variáveis que armazenam valores descritivos de suas características. Essas variáveis são conhecidas como atributos. Por exemplo, a classe **Vulnerabilidade** tem o atributo **Descrição**. Alguns atributos têm como possíveis valores outras classes. Por exemplo, a classe **Produto** possui o atributo **possuiFornecedor**, que tem como possíveis valores as instâncias da classe **Fabricante**. Esses atributos que unem duas classes são conhecidos como relacionamentos. A partir deste ponto a ontologia já possui seu modelo conceitual e têm-se os conceitos de classes, relacionamentos, atributos e instâncias definidas.

A partir da última versão mais refinada, novas classes ainda podem ser obtidas pelo agrupamento semântico dos termos existentes. Todas as subclasses de uma classe herdam as relações daquela classe. O processo de definir novas classes e obter relações é interativo e pode se repetir muitas vezes, até mesmo após a obtenção de um modelo aceitável da ontologia. Após montar uma lista preliminar e trabalhar sobre ela, o próximo passo terá como objetivo definir propriedades, possíveis restrições sobre elas e determinar instâncias.

2.5.6.5 Definição de restrições sobre as relações e determinação de instâncias

As relações têm características chamadas restrições ou facetas, que descrevem características limitantes para as classes. Algumas restrições possíveis são: cardinalidade da relação, o tipo de valor da relação (*slot-value type*), a faixa de valores possíveis para relação (*range*) e as relações definidas para uma classe (*domain*). A cardinalidade de uma relação define quantos valores ela pode ter, por exemplo, um aluno pode retirar cinco livros na biblioteca. O tipo de valor descreve que tipos de valores podem ser utilizados na relação, sendo os mais comuns os *strings*, números, booleanos, etc. Além disso, o tipo de valor pode definir relacionamento entre indivíduos e, nesse caso é conhecido como *tipo-instância*. As classes permitidas para relações do tipo-instância determinam a faixa de valores. As classes às quais uma relação está ligada, ou seja, as classes que são descritas para uma relação determinam o domínio.

Após a definição das restrições, a última atividade relaciona-se à criação de instâncias individuais na hierarquia. Para definir uma instância é preciso escolher uma classe, criar uma instância individual daquela classe e preencher os valores das relações e possíveis restrições.

2.5.6.6 Implementação e instanciação da ontologia

Executadas as etapas previstas, a ontologia deve então ser implementada em uma ferramenta para construção de ontologias, a partir do modelo conceitual. A escolha da linguagem deve levar em conta as recomendações do consórcio W3C sobre linguagem para representação de ontologias e conteúdos da *Web semântica*.

Utilizando a ferramenta de construção de ontologias, as entradas da base de conhecimento são cadastradas, classificando as respectivas classes, capturando o conhecimento do domínio e fisicamente representando-o usando o meta-modelo. (SCHOEN, 1991)

Na seção a seguir são apresentadas algumas ferramentas e linguagens para construção de ontologias.

2.5.7 Ferramentas e linguagens

Além de metodologias, existem ferramentas utilizadas para a construção de uma ontologia. Por se tratar de uma tarefa dispendiosa, qualquer apoio na construção de ontologias pode representar ganhos significativos. (ALMEIDA e BAX, 2003) Alguns autores apresentam estudos comparativos das ferramentas disponíveis, por exemplo, o Portal OntoWeb³⁵ possui uma visão geral de várias ferramentas e uma comparação entre elas segundo os critérios citados, como Ontolingua Server³⁶, Ontosaurus³⁷, WebOnto³⁸,

³⁵ <http://babage.dia.fi.upm.es/ontoweb/wp1/onRoadMap/>

³⁶ <http://www.ksl-svc.stanford.edu:5915>

³⁷ <http://www.isi.edu/isd/ontosaurus.html>

³⁸ <http://webonto.open.ac.uk>

Protégé³⁹, OntoEdit⁴⁰, OilEd⁴¹, apresentando algumas características comuns entre elas, como uma clara e consistente interface, entre outras.

Segundo (Duineveld *et al*, 1999 *apud* Freitas, 2003), alguns dos fatores que tornam as ferramentas populares são: facilidades de uso, entendimento intuitivo da interface, visibilidade gradativa, interfaces gráficas, conexão a repositórios, portabilidade, interoperabilidade, organização dos arquivos gerados, documentação de alterações, suporte a trabalho cooperativo, extensibilidade e ferramentas de apoio que facilitem o desenvolvimento.

Vários exemplos de ferramentas para a construção de ontologias são apresentados na literatura. Neste projeto, utilizaremos a ferramenta *Protégé 2000*, que é um ambiente interativo para projeto de ontologias, de código aberto, que oferece uma interface gráfica para edição de ontologias e uma arquitetura para a criação de ferramentas baseadas em conhecimento, modulada e ainda permite a inserção de novos recursos (NOY; FERGERSON; MUSEN, 2000).

Para Freitas (2003), o ambiente Protégé que vinha sendo desenvolvido pelo Departamento de Informática Médica da Universidade de Stanford face às necessidades de ontologias desde os anos 80, ocupou o nicho de mercado aberto pela Ontolingua e, originalmente como ferramenta de aquisição de conhecimento limitada a um sistema especialista para oncologia, foi se modernizando gradativamente e permitindo diversos formalismos e estratégias de inferências, integrar tarefas (aquisição de ontologias e instâncias, ambiente de teste com inferência) num mesmo ambiente, criar automaticamente formulários para entrada de conhecimento, acessar ontologias e a combinar ontologias.

A melhor decisão de projeto do Protégé está em seu modelo de conhecimento extensível, podendo ser adaptado a diversos usos, com outras características como (Noy *et al*, 2000):

³⁹ <http://protege.stanford.edu>

⁴⁰ <http://www.ontoprise.de>

⁴¹ <http://img.cs.man.ac.uk/oil/>

- A linguagem axiomática PAL (*Protégé Axiomatic Language*), que permite a inserção de restrições e axiomas que incidem sobre as classes e instâncias de uma ou mais ontologias;
- A geração de arquivos de saídas alteráveis, permitindo que sejam implementados componentes para traduzir o conhecimento para outros formalismos através das metaclasses;
- Uma excelente interface para entrada de conhecimento, incluindo um gerador automático de formulários para as classes definidas, admitindo ainda a reposição da interface original por componentes mais adequados à aplicações específicas.

Das muitas ferramentas e ambientes disponíveis que podem ser usados para a construção de ontologias, alguns critérios devem ser considerados na escolha de qual usar, tais como, arquitetura do *software* e evolução da ferramenta, interoperabilidade, representação do conhecimento, serviços de inferência e usabilidade. Essas ferramentas utilizam-se de linguagens de representação para construir suas ontologias.

Vemos em Gava e Menezes (2003) que o homem, como ser inteligente, raciocina sobre o que esta ao seu redor, e para isso, cria uma representação do conhecimento envolvido no mundo, e que, a ontologia descreve esses elementos observáveis no mundo (domínio de conhecimento) e as relações entre eles, que pode estar representada das mais diversas formas, de acordo com o nível de formalismo desejado, quer seja lógica de primeira ordem, redes semânticas, *frames*⁴² ou em linguagem natural.

Existem linguagens tradicionais para sua representação, como CYCL, Ontolingua, F-Logic, CML, etc., e linguagens padrão para *Web*, como OIL (*Ontology Inference Layer*), DAML+OIL (*DARPA Agent Markup Language*), RDF(s), XOL (*Ontology Exchange Language*), SHOE (*Simple HTML Ontology Extensions*), XTM (*XML Topic Maps*) e OWL (*Ontology Web Language*). É importante ressaltar que

⁴² Um *Frame* é uma coleção de atributos, em geral chamados *slots*, valores a ele associados e restrições sobre estes valores, que descrevem alguma atividade do mundo. (Minsky, 1975)

existem diferentes conexões entre os componentes da ontologia, seus paradigmas de representação do conhecimento e suas linguagens de representação.

A ferramenta *Protégé 2000* aqui utilizada pode exportar a ontologia em diversas linguagens diferentes, bem como desenvolvê-la diretamente em OWL.

2.5.7.1 Protégé

O Protégé é uma ferramenta de interface gráfica que dá suporte à aquisição de ontologia e conhecimento. O Protégé suporta a geração de interface para o usuário baseada em ontologia. Uma outra característica deste editor é contemplar uma arquitetura modulada, permitindo a inserção de novos recursos, além de possuir código aberto.

O Protégé possui uma interface de fácil utilização para a criação de instâncias e uma quantidade considerável de *plugins* que aumentam o número de suas funcionalidades. A seguir, conforme ilustra a figura 33, descrevemos alguns elementos da tela inicial:

- 1) contém as guias que permitem a alternância na edição das classes, propriedades, formulários, instâncias e metadados. Os formulários são configurações para a exibição das propriedades das classes que se queira editar, permitindo a personalização da interface.
- 2) são exibidos os ícones para ações relacionadas às classes e para a visualização da árvore de classes da ontologia que está sendo criada. A classe básica “owl:Thing” é a superclasse para todas as especializações e está presente desde a criação da nova ontologia.
- 3) é utilizado para o preenchimento do nome e outras informações relevantes a cada classe.

- 4) são exibidas as descrições lógicas da classe.
- 5) São mostradas as propriedades relacionadas com a classe que está em edição.
- 6) São exibidas as classes disjuntas.

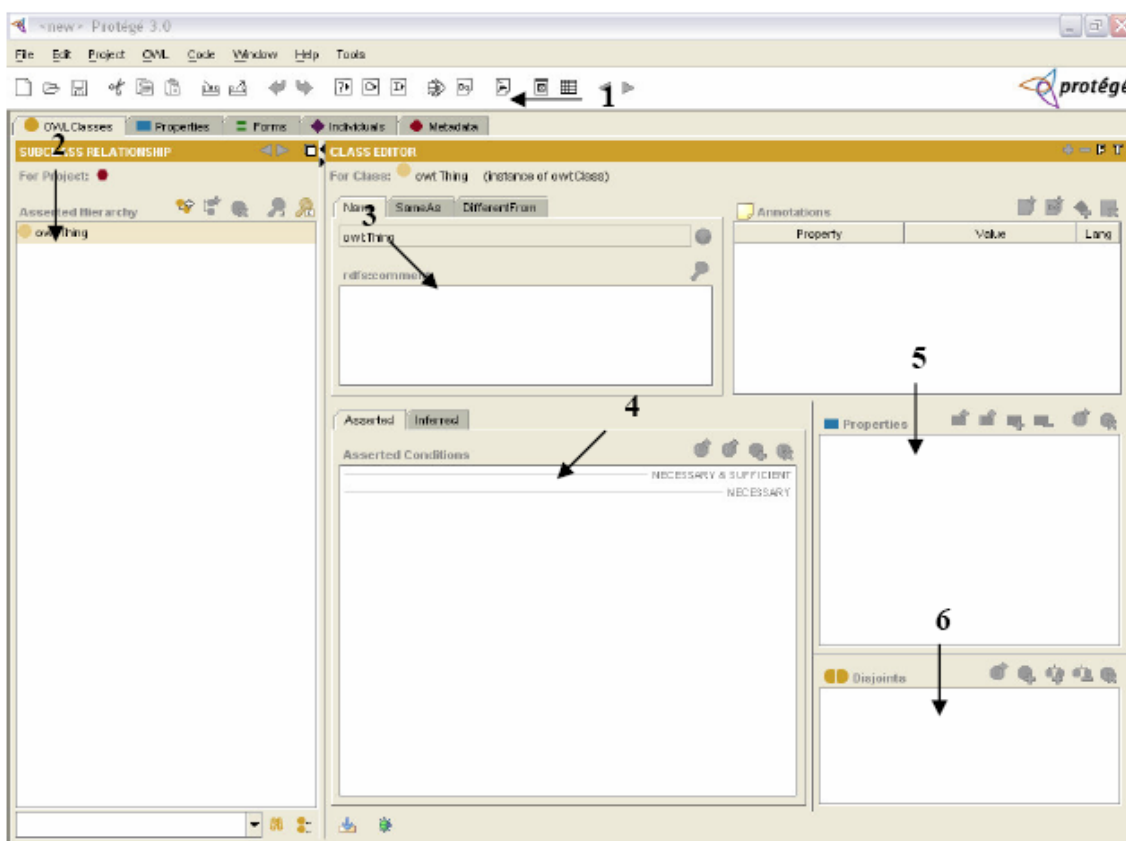


Figura 33 Tela inicial do Protege
 Fonte: <http://protege.stanford.edu/>

2.5.7.1.1 XML

A linguagem *eXtensible Markup Language* (XML) é um padrão de representação que permite a introdução de uma significativa quantidade de metadados para descrever, inclusive com grande nível de detalhamento, o conteúdo temático de um documento, sendo assim, importante na recuperação de informações.

Seu conjunto de regras e convenções de sintaxe pode ser utilizado para a criação de um conjunto personalizado de elementos de marcação, com vistas à descrição do conteúdo de um documento eletrônico. Considerado por diversos autores como a pedra fundamental da Web semântica, tem sido amplamente adotada como instrumento para o desenvolvimento de novas áreas de processamento de informação.

2.5.7.1.2 DTD

Document Type Definitions (DTD) refere-se ao conjunto de regras sintáticas de validação às quais os documentos devem seguir para serem considerados válidos de acordo com o objeto que se pretende representar, foram recomendados inicialmente pelo W3C e tem como desvantagem apenas a verificação sintática dos documentos, não sendo dotada de nenhum tipo de controle semântico.

2.5.7.1.3 XML Schema

Inicialmente proposta pela Microsoft, é uma linguagem de definição de tipos complexos que permite a realização de restrições de vocabulário em documentos XML, simulando a especificação de estruturas de dados, como os nomes de campos e seus respectivos tipos.

O propósito de um XMLSchema ou de uma DTD é a especificação da estrutura de um documento em termos de elementos, atributos e tipos de dados que tal documento pode conter, cuja diferença básica está no fato do DTD não fornecer um controle fino sobre o formato e os tipos de dados dos elementos e seus atributos.

Como sintaxe não é o foco deste trabalho, aqui XML, DTD e XMLSchema são utilizados principalmente para descrever os tipos de dados em OWL, como *xsd:string*, *xsd:integer*, *xsd:boolean*, etc.

2.5.7.1.4 OWL

A OWL (*Web Ontology Language*) é uma linguagem destinada à construção de ontologias que faz parte da crescente lista de recomendações da W3C relacionadas ao desenvolvimento da *Web semântica* e oferece mecanismos para representar explicitamente o significado dos termos e os relacionamentos entre estes termos.

A OWL, independentemente da sintaxe utilizada nos documentos, suporta interoperabilidade representando a semântica dos dados de forma que o conteúdo das informações possa ser processado e seu significado compreendido. Sua sintaxe é baseada no XML, e é uma linguagem compatível com o RDF e RDF Schema (LUSTOSA, 2003).

Além dos recursos oferecidos pela própria OWL, podemos utilizar outros mecanismos como elementos definidos no RDF Schema e XML Schema para que a máquina realize a inferência da semântica dos dados e seja mais expressiva na representação do conhecimento, conforme apresentado na figura 33 abaixo:

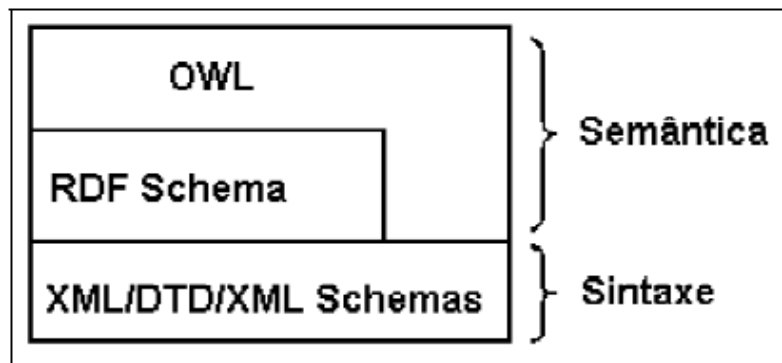


Figura 34 Definição da sintaxe e semântica dos dados
Fonte: Lustosa, 2003.

Os recursos que a linguagem OWL oferece são divididos em três sub-linguagens: OWL Lite, OWL DL e OWL FULL, onde a indicação de cada uma depende da ontologia que se deseja construir.

2.5.7.1.4.1 OWL Lite

A OWL Lite é a sub-linguagem que dá suporte aos usuários que necessitam basicamente de uma hierarquia de classificação e funcionalidades com restrições simples, ou seja, ela apresenta apenas algumas características que a OWL como um todo oferece e impõe algumas limitações aos recursos, como por exemplo, ao uso de cardinalidade que fica restrita a obter somente os valores 0 ou 1.

2.5.7.1.4.2 OWL DL

Esta sub-linguagem é destinada aos usuários que necessitam de um pouco mais de expressividade para formalização de ontologia, ou seja, a OWL DL oferece todos os recursos apresentados pela OWL Lite e também todos os outros mecanismos que a linguagem OWL disponibiliza. Uma característica notada na OWL DL é que, apesar de oferecer todos os recursos da linguagem, ela impõe algumas restrições quanto ao uso destes recursos. Aqui é possível definir classes através da união, interseção e complementos de outras classes, porém não é permitido que uma classe seja instância e uma propriedade ao mesmo tempo.

2.5.7.1.4.3 OWL Full

De todas as três sub-linguagens, esta é a que oferece suporte para ser mais expressiva, pois além de oferecer todos os vocabulários disponíveis pela linguagem OWL, não impõe restrição ao uso dos recursos. Na verdade, a OWL Full utiliza o mesmo vocabulário de OWL DL e diferencia-se na ausência de restrição destes mecanismos.

Esta sub-linguagem não impõe restrição aos valores das classes, às propriedades, às instâncias e aos dados. Com isso, tem-se uma das principais características da OWL Full: a possibilidade de uma classe ser tratada ao mesmo tempo como instância de uma classe.

Assim, os elementos oferecidos pela OWL Full são os mesmos que compõem o vocabulário OWL DL descritos acima, apenas não exigem certas restrições. Essa linguagem é uma extensão de RDF, enquanto OWL Lite e OWL DL são vistas como a extensão de apenas uma parte do RDF. Com isso, todo documento escrito na linguagem OWL é um documento RDF e todo documento RDF é um documento válido em OWL Full. Porém, apesar de todo documento OWL Lite e OWL DL serem visto como um documento RDF, apenas em alguns casos o documento RDF pode ser válido como documento OWL Lite e OWL DL.

2.5.7.1.5 RDF (Resource Description Framework)

O RDF é utilizado para representar os objetos encontrados na *Web Semântica* e também faz parte da lista de recomendação da W3C. É um modelo para representar a informação sobre recursos na Web, mais especificamente os metadados dos recursos, cujo uso possibilita descrever uma expressão que represente a semântica dos dados. Identifica a representação dos seus objetos pelos URIs (*Uniform Resource Identifiers*), que é o esquema que descreve os recursos que estão disponíveis para serem reutilizados ou adaptados para adicionar novas características.

Com isso, o uso do RDF permite a padronização de informações e oferece suporte a interoperabilidade entre os domínios, utilizando-se da linguagem XML para representar a semântica dos dados e dar suporte aos metadados.

O modelo RDF básico é estruturado por um conjunto de tuplas composta por sujeito, predicado e objeto, podendo ser representado através de grafo RDF composto por nós, arcos e retângulos que pode ser construído com os seguintes objetos (NOLETO, 2003):



Figura 35 Representação em grafo
Fonte: Noletto, 2003.

- **Recurso:** refere-se a qualquer coisa que possa ser representada, ou seja, qualquer objeto presente na *Web* que pode ser descrito, lembrando que cada recurso é identificado por um URI. Um exemplo de recurso pode ser uma página Web que tem o endereço <http://www.exemplo.com.br/>.
- **Propriedade:** representa as características ou atributos dos objetos. Aquilo que se refere ao objeto e pode ser representado é dito como sua propriedade. As propriedades podem representar também os relacionamentos entre os recursos. Para o exemplo da página Web, pode-se dizer que o criador desta página é um de seus atributos, ou seja, uma propriedade “criador” pode ser definida para este recurso.
- **Valor:** corresponde ao valor que a propriedade assume. No exemplo citado acima, a propriedade deve assumir um valor específico que se refere ao nome do criador da página. Considera-se que o nome do criador seja “João Silva”; este será o valor que a propriedade irá assumir.
- **Sentença:** é a declaração como um todo, ou seja, o fato de interligar o recurso à propriedade, assim como ao seu valor.

A representação mais complexa suportada por RDF é a expressão de junção (AND), de relacionamentos simples, visto que ele não fornece meios de expressar a negação (NOT) ou a disjunção (OR), como também não fornece mecanismos para descrever as propriedades e todos os relacionamentos entre as elas e outros recursos.

Essa limitação levou a criação do RDF Schema, uma extensão do RDF que permite a descrição de classes, propriedades e outros recursos, com técnicas que permitem a sua reutilização e que são utilizados na definição do significado desses objetos.

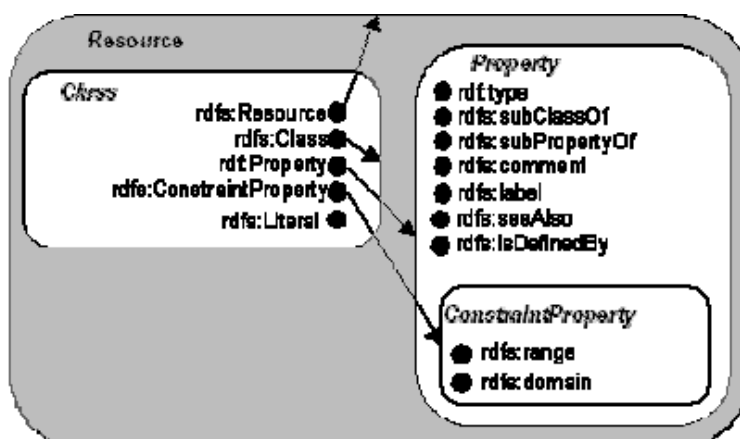


Figura 36 Elementos do RDF Schema
Fonte: Noletto, 2003.

- **Classes:** referem-se aos próprios recursos que devem ser descritos seguindo uma declaração hierárquica. A representação de um determinado objeto é dada pela instância da classe.
- **Propriedade:** o conceito de propriedade é dado pela relação entre o sujeito e o objeto. As propriedades utilizam alguns mecanismos para expressar os relacionamentos entre suas classes, instâncias ou superclasses. (NOLETO, 2003)
- **Restrições:** refere-se às restrições que devem existir ao utilizar as classes e propriedades. Segundo Noletto (2003), estas restrições são os mecanismos utilizados para associar as propriedades identificadas no modelo aos recursos criados para representação do mesmo.

2.5.7.1.6 Recursos do OWL, RDF e RDF Schema

i. Class

As classes correspondem a maior parte da representação dos documentos OWL e são utilizadas para representar os conceitos de certo domínio, sendo definidas como um grupo de indivíduos (instâncias) que possuem propriedades em comum, representadas hierarquicamente, com classes generalizadas (superclasses) e classes especializadas (subclasses). A OWL possui uma classe predefinida chamada “Thing”, em que todo objeto é membro desta classe, sendo assim esta a superclasse de todas as classes.

A OWL apresenta o elemento *owl:Class* para representar uma classe, em que cada classe é identificada por *rdf:ID* que contém o nome da classe. As informações de cabeçalho como *Ontology* e *Imports* não serão tratadas aqui, por não terem sido utilizadas diretamente na pesquisa.

ii. *rdf:Property*

Este elemento da linguagem OWL refere-se às propriedades de uma classe, ou seja, às características ou relacionamento entre os objetos. Além de especificar a classe a qual ela pertence, ou seja, a classe que contém tal característica ou relacionamento, também apresenta o tipo de seu conteúdo, existindo no OWL os tipos de propriedades:

- ii.a. *ObjectProperty*: que fazem referência a uma classe e representa o tipo da imagem da propriedade, ou seja, o tipo do valor que ela irá assumir;
- ii.b. *DatatypeProperty*: Semelhante a propriedade anterior, mas fazendo o relacionamento entre uma classe e um tipo de dado específico utilizando-se dos tipos definidos pelo *XML Schema*. Alguns deles são: booleanos, float, integer, string, entre outros.
- ii.c. *InverseOf*: é utilizado para definir que uma propriedade é o inverso de uma outra propriedade, ou seja, se o objeto X relaciona-se com o objeto Y por meio da propriedade P1, pode ser definida uma outra propriedade, P2, como o inverso de P1. Com isso, pode-se inferir que Y relaciona-se com X por meio de P2.
- ii.d. *TransitiveProperty*: uma propriedade é transitiva quando se pretende demonstrar que se X está contido em Y e Y está contido em Z, então X também está contido em Z. Ou seja, se o objeto X relaciona-se com Y por meio de uma propriedade transitiva P1, e Y relaciona-se com Z também pela propriedade P1, pode-se deduzir que X também se relaciona com Z através desta propriedade.

- ii.e. *SymmetricProperty*: a característica de uma propriedade simétrica é que se X relaciona com Y por meio de uma propriedade simétrica P1, deduz-se então que Y também se relaciona com X por esta mesma propriedade.
- ii.f. *FunctionalProperty*: esse tipo de mecanismo restringe o valor de uma propriedade e garante que apenas um valor seja atribuído à ela, ou seja, determina que a instância de uma classe tenha apenas uma instância desta propriedade. O uso de *FunctionalProperty* permite definir que a propriedade tem cardinalidade mínima igual a 0 e a máxima igual a 1.
- ii.g. *InverseFuncionalProperty*: O valor de uma propriedade do tipo funcional inversa deve valer a apenas uma instância da classe definida para o domínio, ou melhor, se uma classe X relaciona-se com uma classe Y por uma propriedade funcional inversa, então é possível garantir que uma instância de X deve relacionar-se com apenas uma instância de Y.

iii. rdfs:subClassOf

Como foi dito anteriormente, as classes são definidas de forma hierárquica e para suportar essa hierarquia, o recurso *rdfs:subClassOf* é utilizado para determinar classes mais específicas (subclasses) que herdam as características das superclasses. Por exemplo, a classe “Pessoa” pode conter “funcionários” e “clientes”, que apresentam as mesmas características comuns definidas para as pessoas em geral, como também, um “funcionário” pode apresentar características particulares que distingue das especificações de um “cliente”.

iv. rdfs:subPropertyOf

Assim como existe a hierarquia de classes, é possível também criar uma hierarquia de propriedades, definindo-se subpropriedades de outra propriedade. Por exemplo, digamos que a propriedade “*contemConta*” é uma subpropriedade de “*movimentaConta*”, e que se X está relacionado com Y pela propriedade

“*movimentaConta*”, então pode inferir que X também está relacionado com Y pela propriedade “*contemConta*”.

v. `rdfs:domain`

Uma propriedade relaciona uma classe à outra classe ou uma classe a um tipo de dado, contendo o domínio a qual ela pertence, que refere-se ao conjunto de indivíduos a quem a propriedade pode ser aplicada. Assim, se uma propriedade P relaciona um objeto X com um objeto Y e o domínio desta propriedade é representada pela classe Z, então se pode inferir que X é uma instância da classe Z.

As restrições declaradas com *rdfs:domain* são chamadas restrições globais por não dependerem da classe a qual está associada e sim da propriedade em que ela é definida.

vi. `rdfs:range`

É o conjunto de objetos que a propriedade pode ter como valor, ou seja, refere-se à imagem ou valor que a propriedade deve assumir. Seguindo o exemplo anterior, se a classe Z corresponde ao valor da propriedade P, então, a instância de Z é dada pelo objeto Y.

Assim como *rdfs:domain*, *rdfs:range* também é uma restrição global.

vii. `owl:Restriction`

É o conjunto de restrições de propriedades que a OWL Lite permite aplicar, limitando os valores das propriedades que são usadas por instâncias de uma classe, ou seja, é possível restringir os valores que serão obtidos pelas instâncias de uma classe. As restrições que a OWL Lite disponibiliza para os valores da classe são os seguintes:

- vii.a. *allValuesFrom*: é utilizado para restringir a propriedade de uma classe, indicando que a propriedade em particular possui uma restrição quanto a sua imagem, ou seja, se um objeto X de uma classe em particular relaciona-se com Y por meio de uma propriedade P1 e esta propriedade contém uma restrição do tipo *owl:allValuesFrom*, pode-se

deduzir que Y pertence à classe definida pela restrição, ou seja, X somente pode PI por um Y ou Para todo X , o relacionamento PI é uma instância da classe Y .

- vii.b. *someValuesFrom*: semelhante à restrição apresentada na seção anterior, *owl:someValuesFrom* também restringe o valor atribuído a uma propriedade, porém, neste caso, não é obrigatório que todos os valores para a propriedade sejam do tipo determinado, bastando que apenas um dos valores atenda a especificação e com isso, não é possível garantir que todos os valores são instâncias de uma determinada classe.
- vii.c. *minCardinality*: refere-se a quantidade de valores que uma propriedade pode assumir e é utilizado para indicar a cardinalidade mínima de uma propriedade, ou seja, informar a quantidade mínima de valores que a propriedade deve ter. Especificar a cardinalidade mínima de uma propriedade é um processo suportado pela OWL Lite, porém, esta sub-linguagem permite que o elemento receba somente valores 0 ou 1. Esta restrição é superada pela OWL DL, que não determina os valores permitidos ao elemento e a quantidade mínima de ocorrências para uma determinada propriedade, mas deve ser representada por um valor inteiro.
- vii.d. *maxCardinality*: é semelhante a seção anterior, porém indica a quantidade máxima de valores permitida a uma propriedade. De acordo com a restrição imposta pela OWL Lite, em que cardinalidade mínima e máxima devem apresentar apenas os valores 0 e 1, mas, o uso deste elemento por meio da linguagem OWL DL não restringe a quantidade de valores assumidos por uma propriedade e a quantidade máxima de ocorrências pode ser pré-determinada por meio um valor do tipo inteiro.
- vii.e. *cardinality*: é indicado nos casos em que tanto a cardinalidade mínima quanto a máxima têm o mesmo valor, ou seja, tem cardinalidade mínima e máxima igual, obrigando a existência exata desse valor.

vii.f. *hasValue*: Este elemento é utilizado para restringir o valor de uma propriedade a uma determinada classe, este recurso determina, em uma classe, que para ser uma instância desta classe é necessário que uma determinada propriedade tenha um valor específico pré-determinado.

viii. *IntersectionOf*

Em alguns momentos, pode ser interessante definir que uma classe é na verdade a interseção de outros dois indivíduos, podendo este último ser uma classe, uma propriedade ou outra representação contida na ontologia, ou seja, se um indivíduo apresenta características presentes em duas representações diferentes, ele é, na verdade, uma instância da classe definida com interseção destas características.

ix. *oneOf*

Este mecanismo é utilizado em uma classe para descrever uma enumeração de instâncias permitidas a ela, que podem ser pré-definidas por meio do construtor de *owl:oneOf*, criando-se assim um conjunto de objetos possíveis. Não pode ser utilizado em OWL Lite.

x. *disjointWith*

Este construtor é utilizado para definir que classes são disjuntas uma da outra, ou seja, se existem duas classes definidas como sendo distintas, e X é uma instância de uma destas classes, então X não pode ser também instância de outra classe. Não pode ser utilizado em OWL Lite.

xi. *unionOf*

Este elemento especifica que uma classe pode ser definida como a união de outras classes já existentes na ontologia, ou seja, uma classe definida com o elemento *owl:unionOf* determina que suas características referem-se às representadas pelas outras classes as quais faz referência. Não pode ser utilizado em OWL Lite.

xii. `complementOf`

Este construtor é definido em uma classe para especificar os tipos de instâncias que a classe não suporta, ou seja, as instâncias, com exceção daquelas definidas por *owl:complementOf*, podem ser reconhecidas como instância da classe definida por este elemento. Não pode ser utilizado em OWL Lite.

xiii. `rdfs:comment`

Este elemento permite atribuir os comentários desejados sobre a ontologia.

xiv. `rdfs:label`

O uso deste elemento é semelhante a dar um nome para a ontologia. O que diferencia do comentário é que pode conter várias considerações sobre a ontologia. Este elemento também é diferente do URI, pois estes representam a referência da ontologia, não sendo possível acessar uma definição através de *rdfs:label*.

xv. `rdfs:isDefinedBy`

Este elemento é utilizado na definição do sujeito do recurso. Esta propriedade pode ser usada para indicar o vocabulário de RDF em que o recurso é descrito.

xvi. `rdfs:seeAlso`

Semelhante ao item anterior, a diferença é que esta propriedade é utilizada para adicionar mais informações ao sujeito do recurso.

xvii. `versionInfo`

Este construtor é utilizado para especificar a versão atual da ontologia e este elemento é preenchido com um valor literal, pois apresenta apenas informações necessárias para especificar a versão da ontologia.

xviii. `priorVersion`

É utilizado para referenciar uma outra ontologia e fazer o controle de versões. O uso deste mecanismo determina que a ontologia referenciada seja uma versão anterior da ontologia atual.

xix. `backwardCompatibleWith`

Como a ontologia pode apresentar várias versões, este construtor irá especificar se a nova versão é compatível com as versões anteriores. O elemento *owl:backwardCompatibleWith* é uma maneira de informar aos construtores de ontologia que eles podem modificar os documentos para suportar a nova versão. Se *owl:backwardCompatibleWith* não for declarado então a compatibilidade não deve ser suposta.

xx. `incompatibleWith`

Semelhante ao elemento apresentado na seção anterior, este mecanismo também é utilizado para informar a compatibilidade de versões. Porém, o uso deste elemento determina que a versão indicada não seja compatível com a nova versão. Assim, *owl:incompatibleWith* é utilizado para indicar de forma explícita a não compatibilidade com uma versão anterior.

xxi. `DeprecatedClass`

Quando uma nova versão da ontologia for definida e esta apresentar compatibilidade com a versão anterior, pode ser necessário o uso deste elemento. Utilizar *owl:DeprecatedClass* permite determinar que a classe que contém esta identificação está declarada na versão anterior e, apesar de ser compatível à nova versão, sua definição é mantida conforme a versão anterior.

xxii. `DeprecatedProperty`

Este elemento é semelhante ao anterior, porém, ao invés de referenciar e manter as características de uma classe, este mecanismo é feito sobre uma propriedade.

xxiii. Individual

Na definição de uma classe pode ser necessária a identificação de alguns de seus membros (os objetos reais), ou seja, os indivíduos que fazem parte do domínio e quem tem relação com a classe declarada.

xxiv. Igualdade

A OWL Lite apresenta características relacionadas à igualdade e desigualdade e utiliza-se dos seguintes recursos para formalizá-las.

xxiv.a. *equivalentClass*: é um elemento utilizado quando duas classes apresentam as mesmas instâncias, ou seja, classes que apresentam características semelhantes de forma que a instância de uma classe pode ser uma instância de outra classe, sendo assim definidas como equivalentes.

xxiv.b. *equivalentProperty*: é utilizado para definir a equivalência de propriedades. As propriedades definidas como equivalentes apresentam o mesmo valor.

xxiv.c. *sameAs*: oferece a possibilidade de declarar a igualdade de duas instâncias. tem-se que este elemento é usado para definir a igualdade entre qualquer coisa, seja uma classe e um indivíduo, uma propriedade e uma classe, entre outras possíveis.

xxiv.d. *differentFrom*: oferece mecanismos que definem explicitamente que um indivíduo é diferente ou distinto de outro indivíduo.

xxiv.e. *allDifferent*: semelhante ao apresentado por *differentFrom*, suporta a definição para o problema apresentado possibilitando determinar que as instâncias de uma classes são diferentes umas das outras.

PROCEDIMENTOS METODOLÓGICOS

A pesquisa em questão, destinou-se a modelar uma ontologia sobre *log* de eventos do sistema operacional Microsoft Windows Server 2003, com o objetivo de facilitar a recuperação de informações para apoiar o entendimento e a resolução dos mesmos, e na tentativa de atingir os objetivos propostos, optou-se por uma abordagem empírico-teórica, através da realização de uma pesquisa bibliográfica, documental e exploratória.

3.1 Classificação da pesquisa

(GODOY, 1995^a, p.62 *apud* NEVES, 1996, p.1) ressalta a diversidade existente entre os trabalhos qualitativos e enumera um conjunto de características essenciais capazes de identificar uma pesquisa desse tipo:

- O ambiente natural como fonte direta de dados e o pesquisador como instrumento fundamental;
- O caráter descritivo;
- O significado que as pessoas dão às coisas e à sua vida como preocupação do investigador;
- Enfoque indutivo.

A pesquisa bibliográfica foi embasada em autores internacionais e nacionais, em leituras e pesquisas de livros, artigos de revistas, sites da Internet, dissertações de

mestrado e teses de doutorado e, em função de sua própria tipologia qualitativa, baseou-se em investigação da literatura de referência compreendendo representação do conhecimento, ontologias e suas metodologias, a teoria do conceito, recuperação de informações e mecanismos de busca, Protégé, OWL e RDF, com o intuito de recolher informações e conhecimentos prévios acerca da questão de pesquisa e do problema para o qual se procura resposta.

Neste sentido, analisamos as metodologias existentes para elaboração de ontologias e definimos uma alternativa como proposta, já que não existe um modelo único seguido pelos autores.

A pesquisa documental se valeu do levantamento de documentos apresentados nos sites oficiais da Microsoft: *Events and Errors Message Center*⁴³ sobre log de eventos e *Microsoft Windows Server 2003 TechCenter*⁴⁴ sobre estratégias e práticas recomendadas na solução de problemas, bem como, no *Microsoft Computer Dictionary - Fifth Edition eBook*, que ainda não foram utilizados como base de pesquisa na literatura analisada.

Gil (2002), em sua classificação das pesquisas com base em seus objetivos gerais, descreve três grandes grupos: exploratórias, descritivas e explicativas. Esta dissertação se classifica como uma pesquisa exploratória em sua natureza, tendo por objetivo proporcionar maior familiaridade com o problema, com vistas a torná-lo mais explícito e ao esclarecimento de idéias para a construção de hipóteses.

Não sendo definida como pesquisa experimental, devido à ausência de controle das variáveis determinantes de relações de causa e efeito, mas podendo ser enquadrada numa pesquisa quase-experimental, levando-se em consideração a tentativa de se analisar os efeitos da ontologia no entendimento e na resolução de log de eventos, onde seu pré-teste seria o próprio mecanismo de busca atual, e o pós-teste, a busca apoiada na ontologia, com a comparação dos resultados obtidos.

⁴³ http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

⁴⁴ <http://thesource.ofallevil.com/technet/prodtechnol/windowsserver2003/pt-br/library/ServerHelp/4dcfa74e-d61f-4f5b-ba1c-986bc9412b6a.msp?mfr=true>

3.2 Universo da pesquisa

Segundo a Pró-pesquisa⁴⁵, em termos estatísticos, população ou universo de dados é o conjunto dos elementos que tem alguma característica em comum que possa ser contada, medida, pesada ou ordenada de algum modo e que sirva de base para as propriedades a serem investigadas.

A amostra, por sua vez, é um subconjunto representativo ou não da população em estudo, altamente desejada em estatística, que ocorre quando ela apresenta as mesmas características gerais da população da qual foi extraída.

A Microsoft Corporation é uma instituição privada, norte-americana, que mantém uma diversidade de informações: dados estruturados (armazenados em bancos de dados, com modelos de dados formais e manipulados por ferramentas de apoio - *CASE*) e não estruturados (*Web Sites*, documentos internos à empresa, informações de concorrentes, por exemplo), fontes de dados internas e externas, processos formais e informais de troca de informações.

No que tange esta pesquisa, foram recolhidos dados relativos ao processo de entendimento e resolução de *log* de eventos apresentados no site oficial da empresa, que mantém uma infra-estrutura consolidada de informações dos dados advindos de bases de dados estruturadas ou não.

A população foi composta por todos os *log* de eventos existentes no site da Microsoft, para o sistema operacional Microsoft Windows 2003, sejam eles relacionados a sistemas, aplicativos, segurança, serviço de diretório, replicação de arquivos ou servidor DNS.

⁴⁵ <http://www.propesquisa.com.br>

3.2.1 Amostra

Neste trabalho, utilizamos uma amostra intencional objetivando o entendimento e a resolução dos problemas decorrentes do modelo atual utilizado pela Microsoft no seu mecanismo de busca de informações encontrado pelos usuários, que foi apresentado na definição do problema, na **Seção 1.2**.

Desta forma, foi capturado um conjunto completo de todos os *log's* que se encontram no site, que foram impressos conforme consultados, e mantidos encadernados para garantir a confiabilidade da pesquisa. Este procedimento foi efetuado para se garantir que futuras alterações que possam ocorrer no site da Microsoft Corporation não interfiram na pesquisa em questão, antes do término da mesma, como ocorreu ao antigo mecanismo de busca oficial da empresa *Windows Server 2003 Events and Errors*⁴⁶, apresentado na figura 37 abaixo:

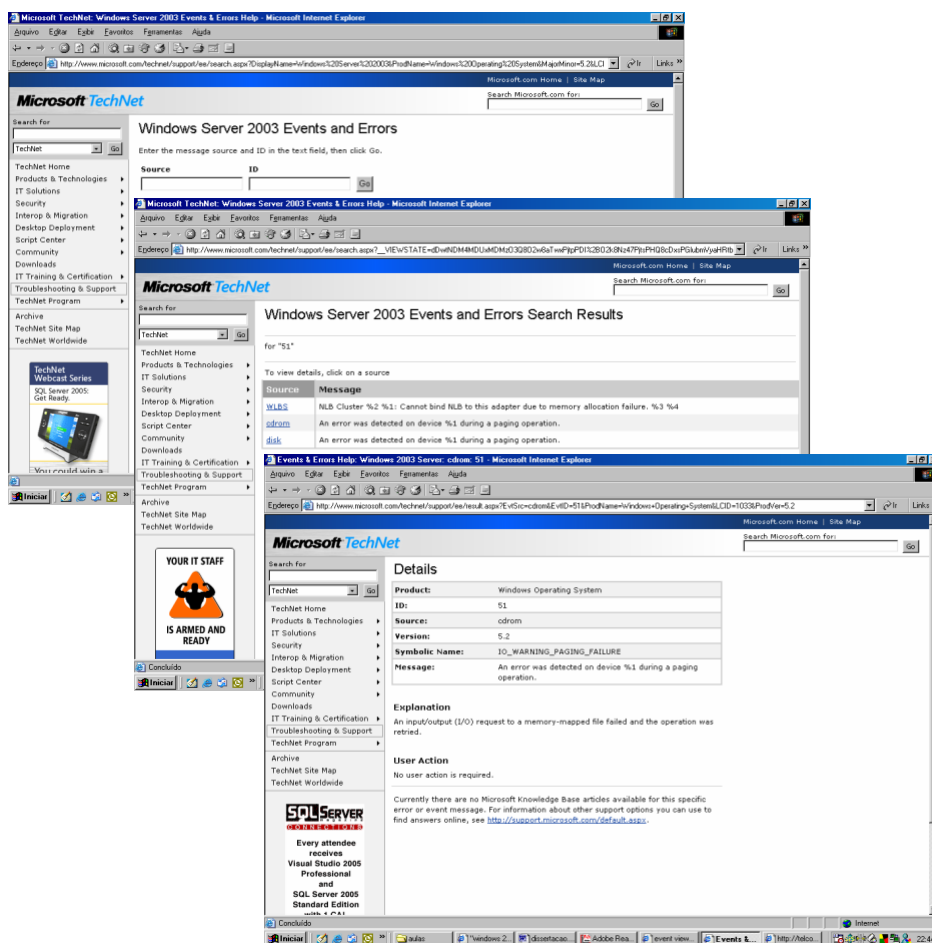


Figura 37 Antigo Site Windows Server 2003 Events and Errors
Fonte: Microsoft Corporation (2005)

⁴⁶ <http://www.Microsoft.com/technet/support/eventerrors.aspx>

Como podemos ver na figura 37 acima, o antigo mecanismo de busca apresentava apenas os campos *Source* e *ID* como entradas ao processo de consulta, o que tornava “pobre” o processo de consulta pelo usuário. O site atual conforme ilustrado na figura 9 incorporou outras informações sobre os *logs* que não constavam na versão anterior. Neste trabalho foi utilizada a nova versão.

A base terminológica, universo da pesquisa utilizado para a extração dos conceitos que gerarão a amostra para a criação de uma ontologia sobre *log* de eventos foi extraída do sistema operacional Microsoft Windows Server 2003 Enterprise Edition seguindo os passos abaixo:

3.2.1.1 Instalação do Virtual PC 2004

Inicialmente, para se manter o rigor científico, com a possibilidade de reprodução integral da pesquisa realizada, foi feita a instalação do programa Microsoft Virtual PC 2004 versão 5.3.582.27 apresentado na figura 38, cujo objetivo é emular uma máquina virtual em cima de outro sistema operacional e, instalar nele um sistema operacional Windows Server 2003 Enterprise Edition.

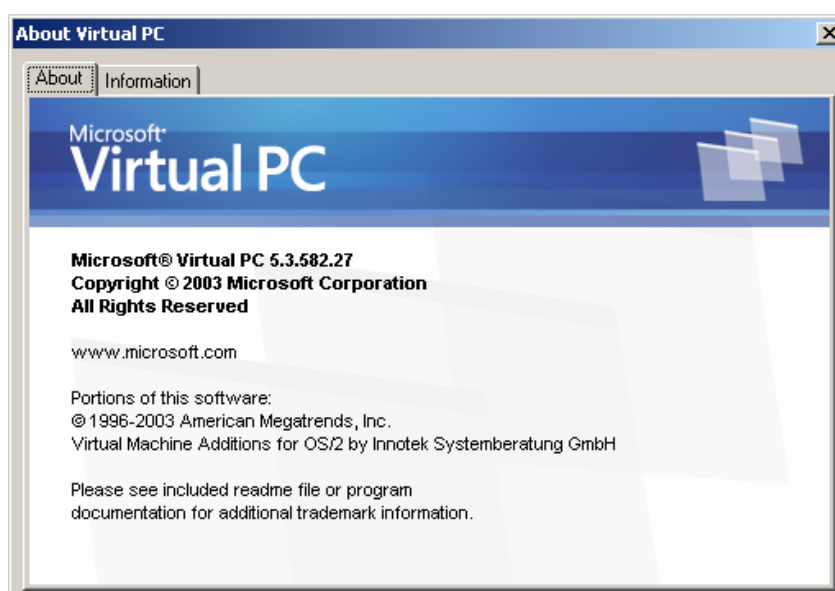


Figura 38 Microsoft Virtual PC 2004
Fonte: tela sobre (*about*) do *software* instalado

3.2.1.2 Instalação do sistema operacional Windows Server 2003



Figura 39 Microsoft Windows Server 2003 Enterprise Edition
Fonte: tela inicial do sistema operacional

Após a instalação do Virtual PC, foram criadas oito novas máquinas virtuais com o sistema operacional Microsoft Windows Server 2003 Enterprise Edition, figura 39 acima. A figura 40 ilustra a seqüência do procedimento de instalação a partir de duas máquinas inicialmente instaladas (1 e 2) como servidor membro, anteriormente já citado, na versão em português (1) e na versão em inglês (2). Após essas instalações, fizemos quatro cópias completas das mesmas, sendo duas de cada uma das versões instaladas. Assim, duas das cópias (3 e 4) foram promovidas a controlador de domínio⁴⁷ para que os *log* de eventos do serviço de diretório, do serviço de replicação de arquivos e do servidor DNS fossem instalados e mantidas assim para análise.

Outras duas cópias (5 e 6) foram então atualizadas completamente utilizando-se o *Windows Update*⁴⁸. As duas últimas (7 e 8) foram completamente copiadas das

⁴⁷ Em um ambiente de domínio do Windows, computador que executa o Active Directory e gerencia o acesso do usuário a uma rede, o que inclui o logon, a autenticação e o acesso ao diretório e aos recursos compartilhados. (Ajuda do Windows)

⁴⁸ O Windows Update é um site da Microsoft que fornece atualizações para o software do sistema operacional Windows e para o hardware baseado no Windows, é uma extensão on-line do Windows que ajuda a manter o computador atualizado. As atualizações abordam questões conhecidas e ajudam a proteger seu computador contra ameaças de segurança conhecidas, como as de segurança e outras atualizações críticas, para ajudar a proteger seu computador contra novos vírus e ameaças à segurança que possam se espalhar pela Internet ou por uma rede. (Ajuda do Windows)

máquinas virtuais já como controladores de domínio (3 e 4) e também atualizadas completamente utilizando-se o *Windows Update*:

Microsoft Windows Server 2003 Enterprise Edition			
Sem Atualização		Atualizado	
Servidor Membro	Controlador de Domínio	Servidor Membro	Controlador de Domínio
1.Português	3.Português	5.Português	7.Português
2.Inglês	4.Inglês	6.Inglês	8.Inglês

Figura 40 Seqüência utilizada nas instalações das máquinas virtuais
Fonte: Autoria nossa.

3.2.1.3 Busca nos registros internos do Windows Server 2003

Na seqüência, foi feita uma busca nos registros internos dos sistemas operacionais Windows Server 2003 Enterprise Edition instalados, nas oito máquinas virtuais diferentes e no servidor de produção de uma grande rede atualmente em funcionamento no Estado de Mato Grosso do Sul, utilizando-se o programa (regedit.exe ou regedt32.exe) apresentado na figura 41, em busca das fontes (*Sources*) dos eventos pré-registrados nessas instalações.

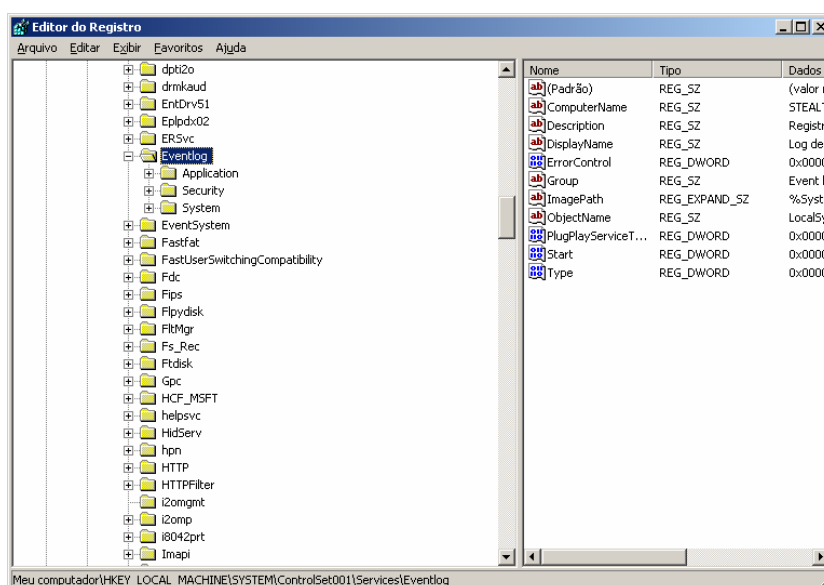


Figura 41 Registry do Windows

Fonte: tela do programa regedit.exe ou regedt32.exe

3.2.1.4 Busca no site oficial da Microsoft

De posse dessas informações, foi então consultada a página oficial do **atual** mecanismo de busca da Microsoft (2006), figura 42, informando as fontes (*Sources*) de eventos extraídos anteriormente nos registros para a obtenção dos *log* de eventos registrados.

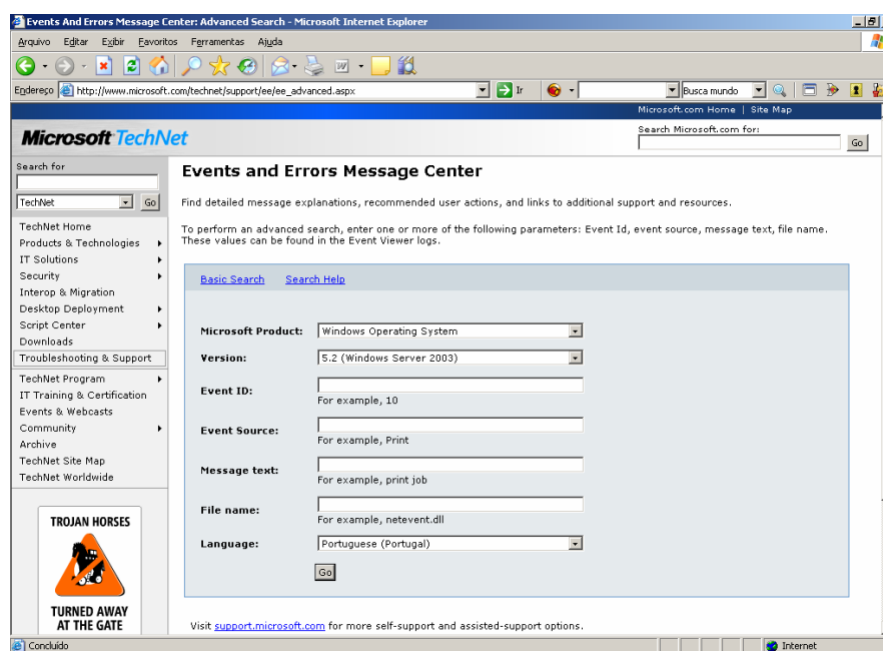


Figura 42 Mecanismo de busca oficial da Microsoft

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Como exemplo, se procurarmos a *Source: Print* no site oficial, utilizando a linguagem: *english*, obtemos o resultado da figura 43:

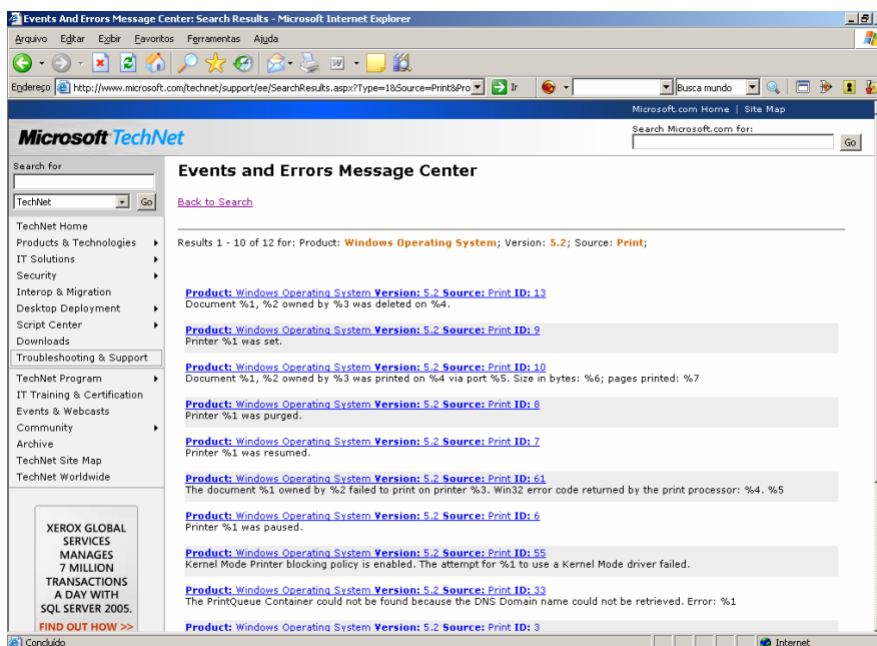


Figura 43 Resultado do mecanismo de busca oficial da Microsoft

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Neste exemplo, ao selecionar o primeiro *link Product: Windows Operating System Version: 5.2 Source: Print ID: 13*, obtém-se o resultado da figura 44:

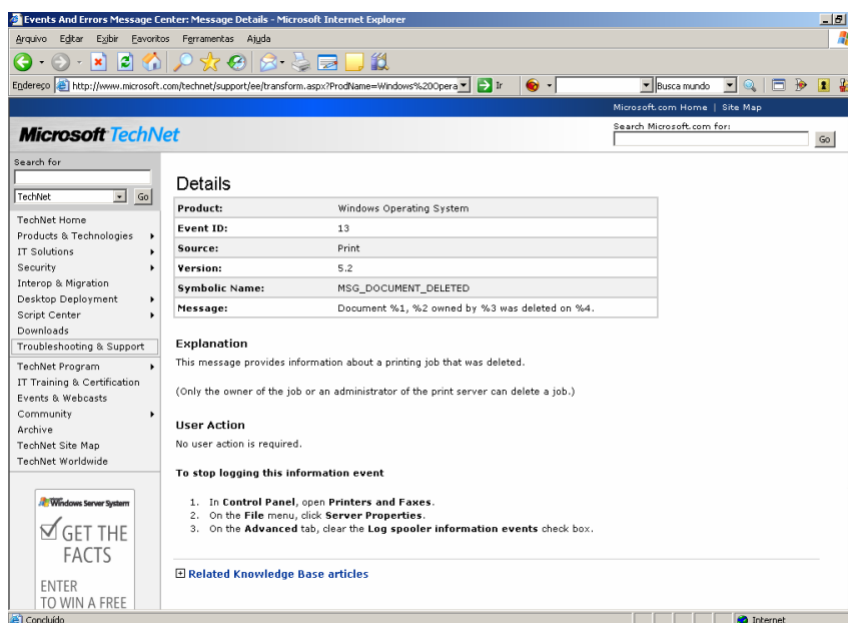


Figura 44 Resultado do *link* selecionado no mecanismo de busca oficial

Fonte: http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Do conjunto de dados retornados pelo site oficial *Events and Errors Message Center* da empresa, selecionamos para nossa amostra, apenas dois termos, que na

ontologia conterão um conjunto de conceitos relacionados, os quais mais se identificavam com os elementos descritos na **Seção 1.2**, definição do problema, no intuito de modelar uma ontologia que ajudasse no entendimento e na resolução dos mesmos.

Assim, a amostra obtida pelos passos anteriores, tratará apenas de alguns eventos representando-os de forma significativa, mas, desde já podemos pressupor que o resultado obtido por eles, se enquadrará também, nos demais resultados que seriam obtidos por quaisquer outros eventos que venham a ser tratados dentro do universo de estudo.

3.2.2 Roteiro para a construção da ontologia

Baseado na adaptação das metodologias de Noy e McGuinness (2001) e Uschold e King (2001), descritas na **Seção 2.5.6** da revisão da literatura, inicialmente definimos o que pretendíamos alcançar com a ontologia construída, estabelecendo para isso, o domínio e o escopo da mesma, respondendo algumas questões básicas e de competência estabelecidas na revisão de literatura e, na seqüência, efetuamos pesquisas sobre ontologias que pudessem estar em meios eletrônicos nas bibliotecas de ontologias disponíveis na Internet.

Executou-se uma pesquisa em bibliotecas de ontologias disponíveis na Internet⁴⁹, na busca de uma ontologia com o mesmo objetivo da proposta neste trabalho, e nada foi encontrado.

No domínio dos *log* de eventos, algumas das possíveis questões de competência são:

- Que tipos de fontes de eventos eu poderia encontrar nos diversos tipos de *log*'s?

⁴⁹ DAML (banco com cerca de 200 ontologias, em DAML+OIL), Ontolingua Server (50 ontologias em Ontolingua), Universal Repository (50 ontologias para educadores e escolas, classificadas por áreas de conhecimento).

- A mesma fonte **Print** é de Aplicativo ou de Sistema?
- A configuração do Protocolo IP funciona sem a configuração do DNS?
- Qual a melhor configuração de *gateway* para minha rede?

A julgar pela lista de questões, a ontologia deve incluir informações de vários *log* de eventos, suas características e tipos, classificações, a escolha apropriada de tarefas para resoluções e recomendações.

Após determinar o escopo da ontologia e delinear os objetivos sobre o que se pretende desenvolver, a próxima etapa consistiu na consulta sobre a existência de ontologias e outras estruturas desenvolvidas no domínio de interesse.

Nossa ontologia cobrirá o universo dos *log* de eventos apresentados pelo sistema operacional Microsoft Windows Server 2003 Enterprise Edition, objetivando a melhoria do entendimento e da resolução dos mesmos por parte dos usuários, provendo respostas para questões relacionadas aos *log* de sistemas, aplicativos, segurança, serviço de diretório, replicação de arquivos ou servidor DNS, e deverá ser mantida por especialistas da área de redes de computadores. Com relação às questões de competência, as mesmas foram respondidas à medida que a modelagem e a análise da ontologia foram sendo realizadas, e serão descritas abaixo.

Criamos então uma lista de termos que consideramos mais importantes analisando inicialmente a base estrutural dos dados, e não o conteúdo dos mesmos, apresentados na **Seção 2.1**, **subseções 2.1.1**, **2.1.2** e **2.1.3**, referentes à *log* de eventos, ao programa aplicativo *Event Viewer* (Visualizar Eventos), aos sistemas operacionais da Microsoft Corporation e ao site *Events and Errors Message Center*, em seguida, identificamos os termos independentes que se tornariam candidatos a classes da ontologia que classificamos como de domínio e de tarefa, seguindo a estratégia *top-down*, pela sua característica estruturalista semelhante a uma base de dados, e descartamos termos desnecessários e modelamos a ontologia contendo os termos e definições relevantes em uma descrição semi-informal e com um vocabulário

artificialmente definido pela linguagem técnica utilizada, que deu suporte à instanciação dos *log* de eventos apresentados.

Definimos então, nossos relacionamentos, atributos, restrições e instâncias, suficientes para demonstrar, mesmo que sucintamente, a utilidade da ontologia no âmbito do objetivo proposto neste trabalho, e finalizamos, implementando as etapas acima executadas, com a utilização do programa *Protégé* utilizando-se da linguagem *OWL* recomendada pela W3C para a representação de ontologias e conteúdos da *Web semântica*.

Essa modelagem inicial, totalmente independente de qualquer aplicação, seguiu alguns padrões que estabelecemos para manter uma ordem nos termos selecionados, conforme apresentamos abaixo:

- As classes e instâncias foram criadas com seus termos no singular, iniciados com a primeira letra em maiúsculo e o restante do termo em minúsculo. Ex: *Serviço, Diretório, Log, Aplicativo* etc. As exceções ocorreram em termos compostos, que mantiveram em maiúsculo cada termo simples contido no termo composto. Ex: *Serviço de Diretório, Log do Aplicativo* etc. E também ocorreram exceções em termos aparentemente plurais, mas designativos de substantivos ou nomes próprios. Ex: *Log de Eventos, Visualizar Eventos* etc.
- Os relacionamentos foram criados com seus termos compostos no singular, iniciados com a primeira letra em minúsculo e com seus termos simples integrantes iniciados com a primeira letra em maiúscula. Ex: *éParteDe, éCompostoPor, éConectadoAo* etc.
- As propriedades das classes foram criadas com seus termos totalmente no singular. Ex: *data, hora, nome, computador* etc.

Esses padrões foram utilizados durante todo o processo de modelagem, o que totalizou um conjunto inicial de 46 classes (*Serviço, Evento, Log de Eventos, Log do Aplicativo, Log de Segurança, Log do Sistema, Log do Serviço de Diretório, Log do Serviço de Replicação de Arquivos, Log do Servidor DNS, Erro, Aviso, Informação,*

atualizado pelo *Windows Update*, um Controlador de Domínio sem qualquer atualização e um Controlador de Domínio atualizado pelo *Windows Update*.

Após a instalação dos mesmos com iguais configurações, utilizamos o programa *regedt32.exe* apresentado na figura 41 para capturar as fontes (*sources*) pré-registradas nessas instalações. Fizemos ainda a captura das fontes (*sources*) em um Servidor de Produção de uma grande rede atualmente em funcionamento no Estado de Mato Grosso do Sul, dados estes que foram tabulados e apresentados no tabela 2 abaixo:

Tabela 2 Resumo das fontes pré-registradas encontradas

Sistema operacional	Instalado como	Status	seq	Log de	quantidade de fontes (<i>sources</i>) encontrada
Windows Server 2003 - Português (Brasil) - Inglês	servidor membro	Sem atualização	01	Aplicativo	067
			02	Segurança	007
			03	Sistema	191
		Com atualização	04	Aplicativo	067
			05	Segurança	007
			06	Sistema	193
	controlador de domínio	sem atualização	07	Aplicativo	068
			08	Segurança	007
			09	Sistema	191
			10	Serviço de Diretório	017
			11	Serviço de replicação de arquivos	002
			12	Servidor DNS	002
		com atualização	13	Aplicativo	068
			14	Segurança	007
			15	Sistema	193
			16	Serviço de Diretório	017
			17	Serviço de replicação de arquivos	002
			18	Servidor DNS	002
Servidor de Produção	controlador de domínio	-x-	19	Aplicativo	082
			20	Segurança	007
			21	Sistema	196
			22	Serviço de Diretório	017

			23	Serviço de replicação de arquivos	002
				Servidor DNS	002

Fonte: Autoria nossa.

Conforme a tabela 2 acima, verificamos que os *log* de eventos dos Aplicativos apresentaram apenas uma alteração com o acréscimo da fonte: *NTFRSPerf* quando no programa Virtual PC promovemos o Servidor Membro para Controlador de Domínio, e apresentaram uma significativa mudança em relação ao Servidor de Produção pela presença de quinze novas fontes: *CA_LIC*, *CA_LIC_CLNT*, *CA_LIC_SRVR*, *CertEnterprisePolicy*, *EvntAgnt*, *Group Policy Management*, *HostMIBAgent*, *IADsTools*, *McLogEvent*, *Microsoft H.323 Telephony Service Provider*, *Ntbackup.ini*, *NTFRSPerf*, *Replication Monitor*, *W3SVC-WP* e *WinsCtrs* que devem ser decorrentes de novos programas aplicativos instalados no mesmo, totalizando um conjunto de 82 fontes (*Sources*) diferentes.

Já com relação aos *log* de eventos do sistema, os mesmos apresentaram duas alterações no programa Virtual PC com o acréscimo das fontes: *NtServicePack* e *Schannel* quando fizemos as atualizações *Windows Update* do servidor, e também uma significativa mudança em relação ao Servidor de Produção pela presença de dezoito novas fontes: *4mmdat*, *aatimeo*, *Automatic Updates*, *cda1000*, *EL90XBC*, *IAS*, *MSFTPSVC*, *PCI_HAL*, *Removable Storage Service*, *SNMP*, *SNMPTRAP*, *Software Restriction Policy*, *stdatw2k*, *TermServLicensing*, *vga*, *W3SVC*, *Windows Installer 3.1*, *WindowsMedia*, totalizando um conjunto de 211 fontes (*Sources*) diferentes.

Os *log* de eventos de Segurança não apresentaram nenhuma alteração em nenhum caso e totalizaram um conjunto de 7 fontes (*Sources*) diferentes, assim como, os *log* de eventos do serviço de diretório e do serviço de replicação de arquivos que aparecem somente em Controladores de Domínio, e do servidor DNS, que também se mantiveram inalterados em todos os casos, totalizando um conjunto de 17, 2 e 2 fontes (*Sources*) diferentes respectivamente.

De posse das informações acima descritas, criamos uma lista em ordem alfabética para cada tipo de *log* de eventos (*Sistema*, *Aplicativo*, *Segurança*, *Serviço de Diretório*, *Serviço de Replicação de Arquivos* e *Servidor DNS*) contendo o conjunto

união das diversas fontes (*sources*) encontradas nos sistemas operacionais instalados nas versões em português (Brasil), em Inglês e no Servidor de Produção, bem como, enquanto Servidor Membro ou Controlador de Domínio, estando ou não completamente atualizados pelo *Windows Update*.

Na seqüência, consultamos o mecanismo de busca oficial da Microsoft apresentado nas figuras 42, 43 e 44, cada uma das fontes (*Sources*), selecionando-se como Produto: *Windows Operation System*, como versão: *Windows Server 2003* e como Língua: *English* procurando conhecer e quantificar os eventos atualmente registrados nessa busca que poderão ser utilizados na construção ou instanciação da ontologia, resultando no seguinte:

Dos 82 *log* de eventos dos Aplicativos consultados, apenas 21 obtiveram algum resultado, com 147 registros retornados cujas fontes estão apresentadas na tabela 3 abaixo:

Tabela 3 Eventos dos Aplicativos que obtiveram resultados na busca.

seq	Fonte (Source)	qtd	Seq	Fonte (Source)	qtd
001	Application Error	004	012	Ntbackup	005
002	Application Hang	002	013	Perflib	004
003	Application Management	001	014	PerfNet	002
004	AutoEnrollment	002	015	SceCli	004
005	Chkdsk	001	016	Userenv	041
006	Ci	017	017	VSS	017
007	DrWatson	001	018	Windows Product Activation	001
008	ESENT	010	019	Winlogon	003
009	EventSystem	001	020	WinMgmt	013
010	EvntAgnt	010	021	WMIAdapter	001
011	LoadPerf	007			

Fonte: Aatoria nossa.

Dos 07 *log* de eventos de Segurança consultados, apenas 01 obteve resultado, cuja fonte (*Source*) foi *Security* com 40 registros retornados.

Dos 17 *log* de eventos do serviço de diretório consultados, apenas 01 obteve resultado, cuja fonte (*Source*) foi *NTDS KCC* com 09 registros retornados.

Dos 02 *log* de eventos do serviço de replicação de arquivos consultados, apenas 01 obteve resultado, cuja fonte (*Source*) foi *NtFrs* com 04 registros retornados.

Dos 02 *log* de eventos do servidor DNS consultados, apenas 01 obteve resultado, cuja fonte (*Source*) foi *DNS Server* com 11 registros retornados.

Dos 211 *log* de eventos do sistema consultados, apenas 48 obteve algum resultado, com 325 registros retornados cujas fontes estão apresentadas na tabela 4 abaixo:

Tabela 4 Eventos de Sistema que obtiveram resultados na busca.

seq	Fonte (Source)	qtd	Seq	Fonte (Source)	qtd	Seq	Fonte (Source)	qtd
001	acpi	003	017	KDC	002	033	Save Dump	001
002	acpiec	002	018	Kerberos	004	034	Server	032
003	Application Popup	002	019	LDM	002	035	Service Control Manager	014
004	atapi	003	020	LDMS	015	036	Setup	004
005	BITS	014	021	LsaSrv	002	037	SideBySide	001
006	Browser	003	022	MrxSmb	010	038	SNMP	003
007	cdrom	003	023	NetBT	005	039	Srv	002
008	DCOM	007	024	Netlogon	010	040	System	001
009	Dhcp	008	025	Ntfs	009	041	System Error	001
010	Disk	008	026	Parvdm	001	042	Tcpip	004
011	dmboot	005	027	PlugPlayManager	009	043	TermDD	001
012	dmio	027	028	Print	012	044	Ultra	001
013	Dnsapi	002	029	RasMan	002	045	USER32	007
014	eventlog	004	030	Redbook	001	046	VolSnap	029
015	ftdisk	001	031	RemoteAccess	019	047	W32Time	017
016	IPNATHLP	007	032	Removable Storage Service	004	048	Windows File Protection	001

Fonte: Aatoria nossa.

As informações acima obtidas nos mostram que parte das buscas efetuadas apresentou resultados, o que nos indica que possivelmente não ocorreram eventos para essas fontes (*Sources*) ou que as mesmas apresentam-se apenas em versões do sistema operacional Windows diferente da que foi estudada.

3.2.3 Teste da ontologia

Foi realizado pela consulta de alguns *log* de eventos apresentados pelo programa aplicativo *Event Viewer* (Visualizar Eventos) no *link* existente na descrição do evento que leva ao site da Microsoft e no site oficial *Events and Errors Message Center* da empresa, que foram comparados com o entendimento e a resolução obtidos com a consulta feita na interface de busca simulada apoiada na ontologia. Finalmente, analisamos e descrevemos qualitativamente os resultados obtidos nas duas formas de consulta e na interface simulada, procurando verificar se houve melhoria no entendimento dos termos e, em seguida, se os relacionamentos entre os conceitos ajudarão a compreender melhor e até mesmo a solucionar os problemas que nos é apresentado nos *log* de eventos. Os resultados apresentados estão na **Seção 4** a seguir:

ANÁLISE E RESULTADOS

Neste capítulo, abordamos os resultados alcançados pelos objetivos propostos neste trabalho, analisando e comparando os mecanismos de busca da Microsoft com a simulação da interface de busca apoiada na ontologia que foi modelada, e apresentando os problemas e soluções encontradas durante os mesmos.

4.1 Análise das ferramentas da Microsoft Corporation

A análise mais detalhada das ferramentas foi realizada na etapa inicial da pesquisa e encontra-se descrita nas **Seções 1.2 e 2.1** da dissertação.

Uma análise do programa aplicativo *Event Viewer* (Visualizar Eventos) feita nas versões do sistema operacional em português e em inglês nos mostra que os mesmos apresentam como recurso disponível, além da descrição do evento, apenas um *link* para o site da Microsoft, em ambos os casos, a ser utilizado na busca de informações mais detalhadas sobre o que está ocorrendo.

Na versão em português, em nenhum dos casos testados, ocorreu uma resposta satisfatória, apresentando-se sempre uma desculpa informando não haver mais informações adicionais, conforme foi mostrado na figura 2 da definição do problema.

Já uma análise do site oficial *Events and Errors Message Center* da empresa, apresentou os recursos descritos na seção 2.1.3 da revisão de literatura, que, em nossa análise apresenta limitações relacionadas à *estrutura dos registros que não permite a captura da semântica, a falta de resolução para todos os logs, o site em inglês*

constituindo uma barreira para muitos usuários, a dificuldade na resolução pela falta de compreensão comum sobre os logs, o custo de resolução em casos recorrentes, a falta de interoperabilidade dos sistemas computacionais em relação aos logs.

4.2 Modelagem da ontologia

Iniciamos nossa análise, considerando a modelagem da ontologia um processo altamente trabalhoso, desgastante e passível de erros, onde verificamos que a sua realização deveria ser feita por um conjunto de profissionais, especialistas no domínio em questão, para que a mesma não possua vizes decorrentes de interpretações, ou de análises incompletas feitas por um profissional nem sempre preparado para uma tarefa de tamanha grandeza.

Verificamos que o processo de construção da ontologia apresentou uma vantagem e uma desvantagem: *a modularidade*. Podemos vê-la como vantagem quando nos permite acrescentar novos termos e conceitos ao domínio em questão, mas, podemos vê-la como desvantagem, pelo fato de que os novos termos acrescentados podem, pelos seus possíveis novos relacionamentos, prover novas conclusões, o que pode tornar as conclusões anteriores incoerentes ou até mesmo incorretas. Ex: inicialmente podemos dizer que “todo *Dispositivo* é um tipo de *Hardware*” e que “o *Hardware* compõe um *Computador*”, então, por dedução chegamos a “todo *Dispositivo* compõe um *Computador*”. Mas, se dissermos “o *Hardware* inclui qualquer *Dispositivo* que seja conectado ao *computador*“, então, não podemos mais afirmar nossa dedução.

Assim, como acontece nos sistemas especialistas, podemos induzir o usuário a não chegar a uma decisão, se os fatos que dispõe não forem suficientes, ou então, a chegar a uma conclusão errada em função dos fatos que encontrar, e este ponto deve ser seriamente analisado, quando nos referimos as pequenas alterações que a modularidade e o processo de construção da ontologia nos permite fazer. Outro exemplo é o próprio termo *Usuário*, que inicialmente foi construído na ontologia como uma classe, depois foi passado a propriedade da classe *Evento*, e retornou novamente a classe, por pertencer agora a uma classe *Conta de Usuário*.

Outra dificuldade encontrada na modelagem da ontologia, diz respeito ao fato de que as teorias encontradas não deixam claro quando devemos tomar a decisão de colocar um determinado termo como *classe* ou como *instância*, e com isso, nos deparamos em diversas situações em que tomamos essa decisão intuitivamente, como, por exemplo, *Erro*, *Alerta* e *Informação*, enquanto tipo de evento.

Neste trabalho tentamos usar o maior número possível de conceitos abstratos na nossa ontologia, como *serviço*, *evento*, *log* etc, por terem uma maior fidelidade na representação semântica, não contendo simplificações e existindo independente de um determinado objeto em si, e instanciar os objetos como *impressoras*, *monitores* etc que normalmente representam objetos reais.

Nossa construção inicial apresentou relacionamentos hierárquicos básicos, como *hasSubClass* e *hasInstance* que são mantidos em inglês pela própria ferramenta Protégé, bem como, termos sinônimos, como: *erro* e *mensagem de erro*, *log* e *log de eventos*, *usuário* ou *nome do usuário* etc, conforme ilustrado na figura 46:

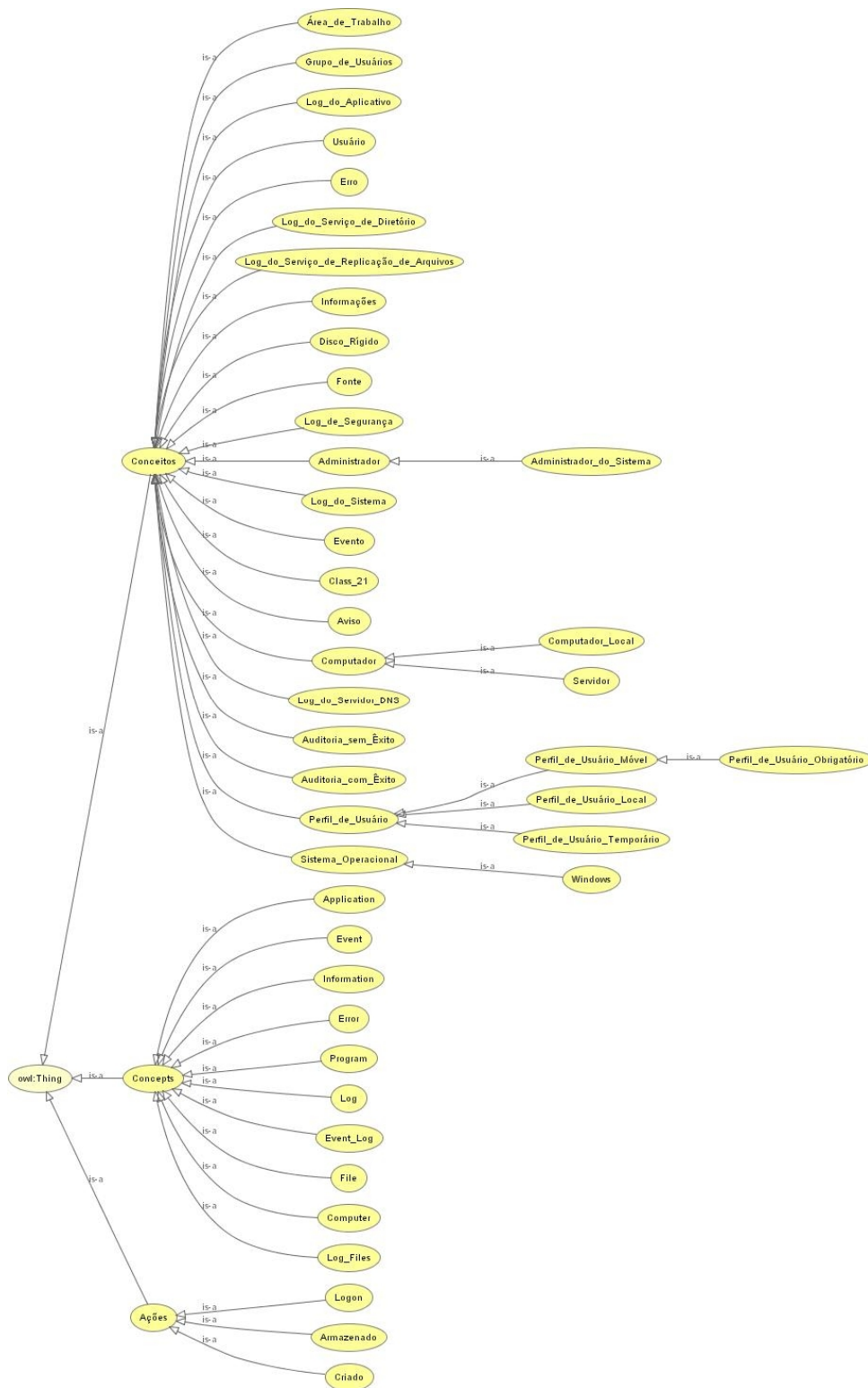


Figura 46 Modelagem inicial da ontologia de domínio
Fonte: OWLViz

Uma das falhas mais problemáticas que encontramos no ponto de vista metodológico, refere-se à falta de uma análise terminológica e lingüística que sentimos necessária durante a fase de seleção e montagem dos termos da ontologia, e para

podermos efetuar uma análise nesse sentido, apresentamos na tabela abaixo algumas das classes modeladas na ontologia de domínio:

Tabela 5 Resultados na busca no site oficial da Microsoft.

Termo	Definição
Serviço	PROGRAMA, ROTINA ou PROCESSO que executa uma função específica do SISTEMA para dar suporte a outros programas, especificamente em um nível baixo (próximo ao HARDWARE). (Microsoft)
Evento	é qualquer ocorrência significativa no SISTEMA ou em um APLICATIVO que exija que os USUÁRIOS sejam notificados ou que uma entrada seja adicionada a um LOG. (Microsoft)
Log de Eventos	é o ARQUIVO que contém INFORMAÇÕES e MENSAGENS DE ERRO para todas as ATIVIDADES do COMPUTADOR. (Microsoft)
Log do Aplicativo	contém EVENTOS registrados por APLICATIVOS ou PROGRAMAS. (Microsoft)
Log de Segurança	registra EVENTOS como tentativas de LOGON válidas e inválidas, assim como eventos relacionados ao uso de RECURSOS, como criar, abrir ou excluir arquivos ou outros objetos. (Microsoft)
Log do Sistema	contém EVENTOS registrados pelos COMPONENTES DE SISTEMA do WINDOWS. (Microsoft)
Log do Serviço de Diretório	contém EVENTOS registrados pelo SERVIÇO DE DIRETÓRIO do WINDOWS. (Microsoft)
Log do Serviço de Replicação de Arquivos	contém EVENTOS registrados pelo SERVIÇO DE REPLICAÇÃO DE ARQUIVOS do WINDOWS. (Microsoft)
Log do Servidor DNS	contém EVENTOS registrados pelo SERVIÇO DNS do WINDOWS. (Microsoft)
Erro	um PROBLEMA SIGNIFICATIVO, como a PERDA DE DADOS ou FUNCIONALIDADE. (Microsoft)

Aviso	Um EVENTO que não é necessariamente significativo, mas pode indicar um possível PROBLEMA FUTURO. (Microsoft)
Informação	um EVENTO que descreve a operação com êxito de um APLICATIVO, DRIVER ou SERVIÇO. (Microsoft)
Auditoria com Êxito	uma tentativa de acesso de segurança que passou por AUDITORIA e obteve ÊXITO. (Microsoft)
Auditoria sem Êxito	uma tentativa de acesso de segurança que passou por AUDITORIA e FALHOU. (Microsoft)
Data	a DATA na qual o EVENTO ocorreu. (Microsoft)
Hora	a HORA LOCAL na qual o EVENTO ocorreu. (Microsoft)
Usuário	o NOME DE USUÁRIO em nome do qual o EVENTO ocorreu. (Microsoft)
Computador	o NOME DO COMPUTADOR onde o EVENTO ocorreu. (Microsoft)
Identificação do Evento	um número que identifica o TIPO DE EVENTO específico. (Microsoft)
Fonte	o SOFTWARE que registrou o EVENTO, que pode ser um NOME DE PROGRAMA, como o "SQL Server," ou um COMPONENTE DO SISTEMA ou de um grande PROGRAMA, como um NOME DE DRIVER. (Microsoft)
Tipo	uma classificação da GRAVIDADE DO EVENTO: ERRO, INFORMAÇÕES ou AVISO nos LOGS DO SISTEMA ou do APLICATIVO, AUDITORIA COM ÊXITO ou AUDITORIA SEM ÊXITO no LOG DE SEGURANÇA. (Microsoft)
Categoria	uma CLASSIFICAÇÃO DO EVENTO definida pela FONTE DO EVENTO. (Microsoft)
Descrição	é freqüentemente a parte mais útil das INFORMAÇÕES, indicando o que acontece ou o SIGNIFICADO DO EVENTO.

	(Microsoft)
Dados Opcionais	são DADOS BINÁRIOS, que podem ser exibidos em BYTES ou em PALAVRAS, gerados pelo PROGRAMA que gerou o registro de EVENTO, exibidos no FORMATO HEXADECIMAL e cujo significado pode ser interpretado por um TÉCNICO DE SUPORTE familiarizado com o PROGRAMA de origem. (Microsoft, adaptado)
Sistema Operacional	PROGRAMA que gerencia o armazenamento e recuperação das INFORMAÇÕES no COMPUTADOR. (Microsoft)
Windows	É o SISTEMA OPERACIONAL criado pela Microsoft Corporation.
Explicação	EXPLICAÇÃO detalhada da DESCRIÇÃO.
Ação do Usuário	Procedimentos recomendados.
Recursos e Suporte Adicional	ARTIGOS relatados na BASE DE CONHECIMENTO.

Fonte: Eventos de Sistema

Uma análise dos conceitos apresentados acima nos indica aqueles que não devem ser implementados na ontologia como classes, por ocorrerem pontualmente no momento em que o evento é registrado em determinado computador, como por exemplo, *Data*, *Hora* e *Computador*.

Continuando a análise, encontramos sinônimas, consideradas no OWL como equivalentes, mas que não conseguimos implementar na prática, como por exemplo, *Descrição* ou *Significado do Evento*, *Fonte* ou *Fonte do Evento*, *Mensagem de Erro* ou *Erro*, etc. encontramos também termos homonímicos, ambíguos e polissêmicos, como por exemplo, *Fonte* (evento) e *Fonte* (Tipo de Letra), termos homógrafos que complementamos com uma palavra ou frase qualificadora, seguindo as diretrizes dos tesouros monolíngües.

Diversas siglas em inglês foram encontradas, como DNS (*Domain Name System*), DHCP, WINS etc. Nossa análise inicial não encontrou nenhum termo acrônimo.

Conforme apresentado acima, os conceitos poderão ser utilizados em processamento de linguagem natural desde que incluídos os sinônimos necessários, bem como, um mapeamento para o uso em mais de uma língua, conforme tratado neste trabalho nas classes *DomainConcept* em inglês e *Conceitos do Domínio* em português, ajudando assim, a eliminar a barreira existente para muitos usuários.

No âmbito da terminologia, esses mesmos mecanismos permitem a sistematização dos conceitos e, conseqüentemente, a elaboração de definições consistentes, o que nos leva a próxima etapa do nosso projeto, o levantamento dos *log* de eventos existentes para o desenvolvimento parcial da ontologia de tarefa.

4.3 Interface simulada de busca

Selecionar uma representação significa fazer um conjunto de compromissos ontológicos, o que nem sempre atende satisfatoriamente a totalidade da necessidade do usuário, por isso, e por não ter sido possível levantar as ferramentas visuais já disponíveis de plugin para o Protégé, propomos desde já uma reavaliação da interface com a criação de uma outra mais amigável ao usuário, devido a complexidade que o mesmo apresenta ao usuário, o que muitas das vezes, acaba por dificultar ao invés de facilitar o entendimento.

Apesar de usarmos o *plugin* do Jambalaya na construção da interface simulada, esses compromissos, citados acima, filtram a representação e determinam o que pode ser visto, enfocando alguma parte do mundo em detrimento de outras, e apesar de parecer uma falha, é uma das mais importantes contribuições devido a esmagadora complexidade do mundo real, conforme mostra a figura 47 abaixo:

Um dos benefícios confirmados neste trabalho com a busca simulada apoiada na ontologia foi ajudar as pessoas a raciocinar e entender o domínio em questão, servindo de referência de um consenso, eliminando realmente contradições e inconsistências, pela especificação não ambígua e pela possibilidade ilimitada de reuso.

Conforme figura 48, apresentamos nossa a seguir o funcionamento da nossa interface simulada de busca apoiada em uma ontologia:

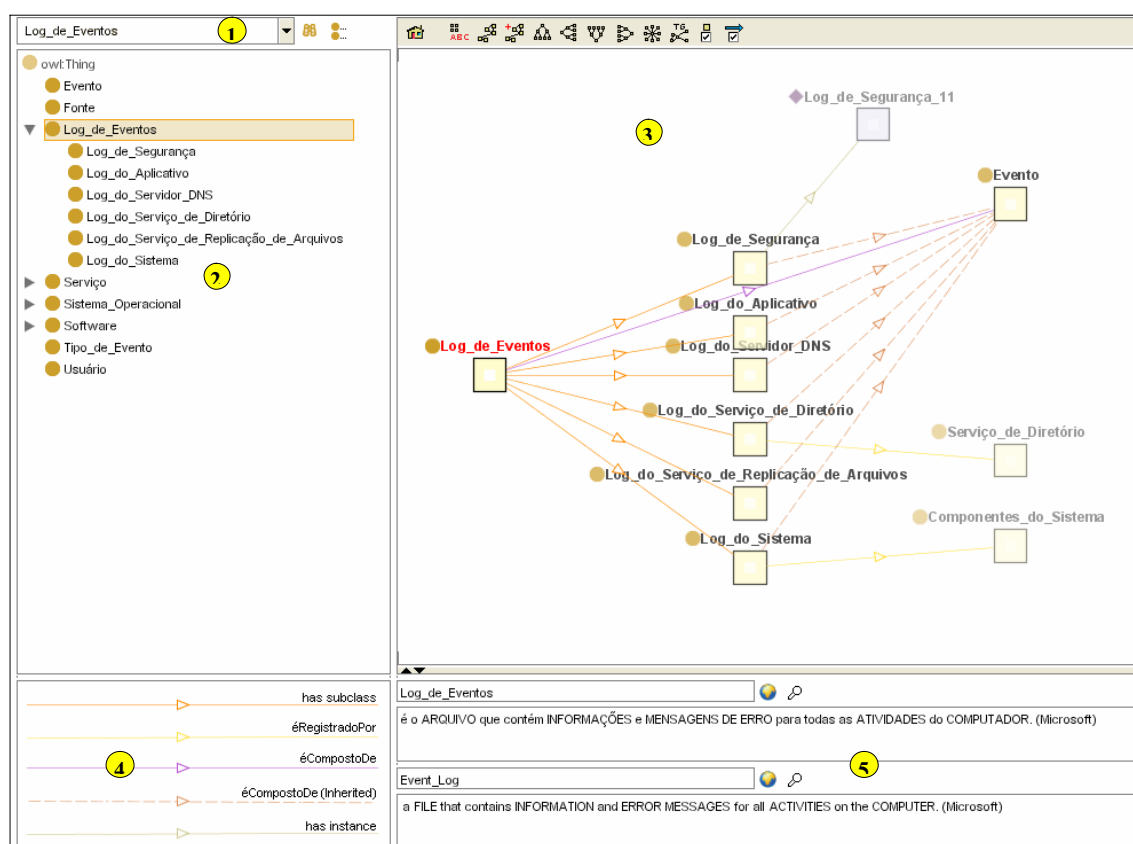


Figura 48 interface de busca apoiada em uma ontologia (simulação)

No campo 1, o usuário informa o termo, sigla, ou qualquer palavra que deseje encontrar um melhor entendimento ou uma solução relacionado a determinado evento apresentado no programa aplicativo Event Viewer (Visualizar Eventos) da Microsoft, ao lado, existe um símbolo de um binóculo, que o usuário irá clicar para iniciar o processo de busca apoiada na nossa ontologia.

Caso a ontologia seja criada com termos sinônimos, então, será possível efetuar a busca em linguagem natural e o mecanismo de busca encontrar por aproximação todas as palavras procuradas.

No campo 2, o mecanismo apresenta de forma hierárquica e taxonômica, todas as classes relacionadas ao termo que foi buscado, destacando o mesmo, conforme apresentado na figura 48 acima, e também outros termos existentes na ontologia, o que pode facilitar a busca e o entendimento do termo que esta sendo procurado.

O símbolo encontrado ao lado do binóculo, com dois pontos, se pressionado, filtrará o mecanismo de busca e apresentará o número de níveis pré-configurados.

No campo 3, é apresentado um *plugin* do Jambalaya, onde podemos visualizar todas as classes, baseando-se no termo central procurado, e também o número de níveis pré-configurados, bem como as instâncias e os relacionamentos existentes entre os conceitos e instâncias, este é o principal local de apoio ao usuário em sua busca, pois possibilita a interação e o entendimento dos conceitos e relacionamentos existentes, bem como facilitar a resolução de eventos ocorridos.

Ainda no campo 3, na parte superior, apresenta-se um pequeno menu gráfico, onde podemos selecionar qual a forma que desejamos visualizar a ontologia apresentada, seja ela na horizontal, vertical, centralizada ou outra disponível pelo *plugin*, no intuito de facilitar a visualização e o entendimento por parte do usuário. O primeiro elemento, uma casinha, centraliza a apresentação na tela, independente do formato escolhido.

No campo 3 também, na parte inferior do mesmo, encontramos duas setas, que são responsáveis pela configuração do que será apresentado pelo *plugin*. Nestas opções podemos selecionar quantos níveis iremos apresentar, bem como, quais os relacionamentos, instâncias etc irão ser apresentados no *plugin*.

No campo 4, apresentamos uma legenda explicativa apresentando os relacionamentos existentes no gráfico do *Jambalaya*, também neste campo, podem aparecer TODOS os eventos atualmente registrados no site oficial da Microsoft, fonte desta pesquisa, de acordo com o termo selecionado.

No campo 5, apresentamos os termos selecionados, com suas respectivas descrições, apresentadas atualmente na língua inglesa e portuguesa, o que auxiliará os usuários com dificuldade no entendimento de outras línguas. Lembramos que podemos configurar a ontologia para que seja possível a inclusão de outras línguas, abrangendo um número maior de usuários que possam vir a utilizar o mecanismo de busca.

Também no campo 5, apresentamos um “globinho”, que se pressionado, efetua a busca em mecanismo de busca da internet, previamente configurado como *Google*, já a “lupinha” retornará as informações contidas diretamente do site em inglês relacionado aos *log* de eventos da própria Microsoft.

4.4 Análise das buscas

Para ultrapassarmos o problema encontrado na tentativa de captura semântica, podemos seguir os exemplos da *Web semântica*, especificando padrões para os registros e preenchimento dos *log* de eventos e especificando seus termos inseridos numa estrutura significativa e definindo formatos de intercâmbio e esquemas de representação do conhecimento.

Assim, iniciamos fazendo uma análise dos *log* de eventos capturados, buscando entender seu comportamento, e comparando alguns deles com a interface de busca simulada, no intuito de verificar se esta interface avalia o entendimento e resolução destes *log* de eventos.

Iniciamos nossa análise das buscas pelo *log* de eventos do sistema, em fontes como *acpi* (id: 5 e 4), que apresentam a mesma explicação e ação do usuário, e uma descrição tão semelhante, que o usuário fica confuso sem saber se não se trata do mesmo evento, conforme figura 49 abaixo:

Details

Product:	Windows Operating System
Event ID:	5
Source:	Acpi
Version:	5.2
Symbolic Name:	ACPI_ERR_AMLI_ILLEGAL_I[REDACTED] FATAL
Message:	%2: ACPI BIOS is attempting to [REDACTED] an illegal IO port address (%3), which lies in the %4 protected address range. This could lead to system instability. Please contact your system vendor for technical assistance.

Explanation

The system BIOS attempted to incorrectly access hardware resources concurrently with the operating system. This type of access cannot be synchronized and can cause system instability.

User Action

Contact the system vendor for an updated BIOS and follow the vendor's instructions for installation.

Details

Product:	Windows Operating System
Event ID:	4
Source:	Acpi
Version:	5.2
Symbolic Name:	ACPI_ERR_AMLI_ILLEGAL_I[REDACTED] FATAL
Message:	%2: ACPI BIOS is attempting to [REDACTED] an illegal IO port address (%3), which lies in the %4 protected address range. This could lead to system instability. Please contact your system vendor for technical assistance.

Explanation

The system BIOS attempted to incorrectly access hardware resources concurrently with the operating system. This type of access cannot be synchronized and can cause system instability.

User Action

Contact the system vendor for an updated BIOS and follow the vendor's instructions for installation.

Figura 49 Eventos Acpi (5 e 4)

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Neste caso, a ontologia traria o esclarecimento necessário ao usuário pela definição dos termos que estão mudando o entendimento semântico da descrição acima.

Fontes como *atapi* (id: 26) e *ultra* (id: 26), apresentam uma descrição e explicação vagas, e uma ação do usuário que nos remete a um contato externo com o fabricante do hardware em questão, conforme figura 50 abaixo:

Details

Product:	Windows Operating System
Event ID:	26
Source:	atapi
Version:	5.2
Symbolic Name:	IO_WRN_BAD_FIRMWARE
Message:	The driver has detected that device %1 has old or out-of-date firmware.

Explanation

Many errors on the device have been logged.

User Action

Contact your hardware support provider for updated firmware for the drive.

Figura 50 Eventos atapi (26)

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Neste caso, a ontologia tratando a comunicação e a interoperabilidade entre as empresas, traria um entendimento e a uma resolução que só se apresenta possível com um contato com o suporte do *hardware* em questão.

Já a figura 51 abaixo, nos apresenta a ontologia esclarecendo o fato acima citado, onde verificamos que *driver* é um *software* utilizado pelo dispositivo (*device*) em questão, que é antigo e precisa sofrer uma atualização do seu *firmware*, que é o programa interno de controle do dispositivo.

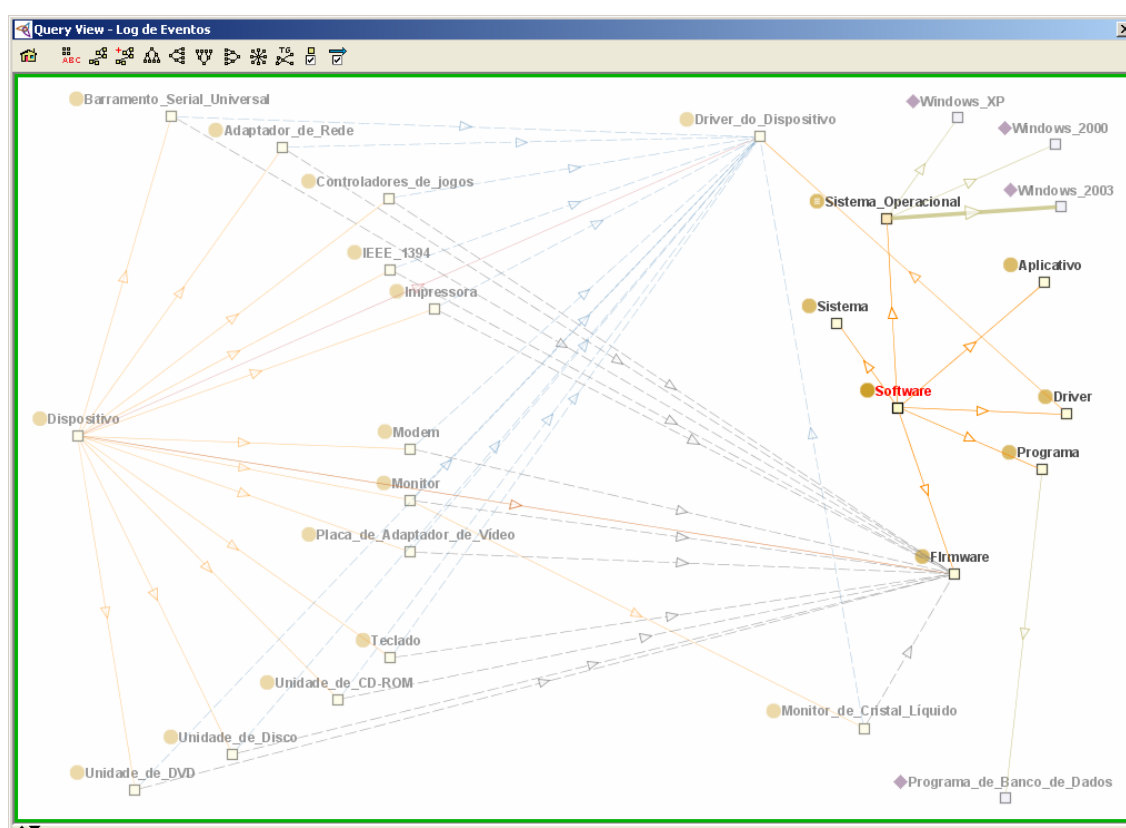


Figura 51 Ontologia parcial de Software
Fonte: Nossa autoria (Jambalaya)

Fontes como *cdrom* (id: 7, 11 e 51) e *disk* (7 e 11), apresentam uma descrição, uma explicação e uma ação do usuário onde a ontologia nos permite um melhor entendimento e resolução do que está acontecendo, conforme figura 52 abaixo:

Details

Product:	Windows Operating System
Event ID:	7
Source:	cdrom
Version:	5.2
Symbolic Name:	IO_ERR_BAD_BLOCK
Message:	The device, %1, has a bad block.

Explanation
The device has a bad block of memory, which Windows attempted to read. The data might be missing or corrupted.

User Action
Do one or all of the following:

- Retry the operation.
- Clean the physical hard drive.
- Use newer media.

Details

Product:	Windows Operating System
Event ID:	11
Source:	cdrom
Version:	5.2
Symbolic Name:	IO_ERR_CONTROLLER_ERROR
Message:	The driver detected a controller error on %1.

Explanation
This problem is typically caused by a failing cable that connects the drive to the computer.

User Action
Replace the cable.

Details

Product:	Windows Operating System
Event ID:	51
Source:	cdrom
Version:	5.2
Symbolic Name:	IO_WARNING_PAGING_FAILURE
Message:	An error was detected on device %1 during a paging operation.

Explanation
An input/output (I/O) request to a memory-mapped file failed and the operation was retried.

User Action
No user action is required.

Figura 52 Eventos cdrom (7, 11 e 51)

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Neste caso, novamente temos um Dispositivo (device) apresentando *bad block*, ou seja, trilhas do disco apresentando defeito, o que pode requerer a troca do dispositivo (7), falha de paginação, que é a troca entre os dados da memória e do arquivo gravado no disco rígido (51) e erro na controladora, o que se supõe ser um problema no cabo, mas pode ser um problema físico interno no dispositivo (11), conforme apresenta figura 53 abaixo:

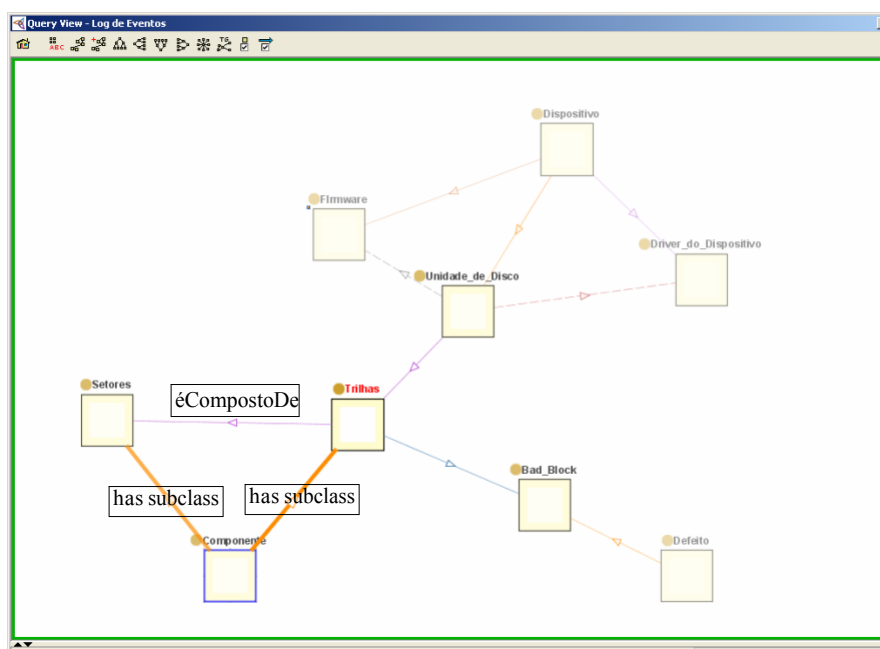


Figura 53 Ontologia parcial de Trilhas

Fonte: Nossa autoria (Jambalaya)

Fontes como *disk* (id: 11 e 11) e *ntfs* (id: 55 e 55), apresentam uma descrição, uma explicação e uma ação do usuário diferente, mas, um *Event ID* iguais, uma análise mais aprofundada nos mostra que esses eventos referem-se à *Version* diferentes (5.2 e 5.2.3790.1830), o que se refere a alguma atualização ocorrida no sistema operacional, a qual não existia no *release* anterior, conforme figura 54 abaixo:

Details

Product:	Windows Operating System
Event ID:	11
Source:	disk
Version:	5.2
Symbolic Name:	IO_ERR_CONTROLLER_ERROR
Message:	The driver detected a controller error on %1.

Explanation
This problem is typically caused by a failing cable that connects the drive to the computer.

User Action
Replace the cable.

Details

Product:	Windows Operating System
Event ID:	11
Source:	disk
Version:	5.2.3790.1830
Message:	Disk - The driver has detected a controller or cable error.

Explanation
This event indicates that I/O failures have occurred on the volume.

Cause
Possible causes include:

- A hardware failure that prevents communication with a disk (for example, a loose cable, a loose disk controller card, or a cable failure).

User Action
Do one or more of the following:

- Check the status of your hardware for any failures (for example, a disk, controller card, or cable failure).
- Check Event Viewer for additional events from lower-level storage drivers that might indicate the cause of the failure.
- Restart the computer.
- Contact Microsoft Customer Service and Support.

Figura 54 Eventos disk (11 e 11)

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Neste caso, a ontologia apresenta o entendimento e a possível resolução, já que os conceitos são independentes do aplicativo, programa ou software onde estão sendo aplicados, mas não resolve o problema relacionado ao *Event ID*, que deveria ter a função de identificar os eventos de forma única.

Fontes como *dmboot* (id: 2 e 1), *dmboot* (id: 6, 5 e 4), *dmio* (12, 9, 8, 7, 6, 5, 4, 37, 36, 10, 35, 32, 31, 30, 29, 28, 26, 25, 21, 20, 19, 18, 24, 23 e 22), *LDMS* (1500, 1000, 3016, 3011, 3007), *LDMS* (3006, 305, 3004, 3003), *ntfs* (id: 37 e 36), *Server* (id: 4004, 4003, 4002, 4001),

Server (id: 2511 e 2507), *Server* (id: 2027, 2026 e 2025), *Server* (id: 2009 e 2001), *Server* (id: 4013, 4012 e 4011), *Server* (id: 4007 e 4005), *VolSnap* (id: 9, 8, 6, 5, 4, 21) e *VolSnap* (id: 41, 38, 36, 35), apresentam uma descrição, uma explicação e uma ação do usuário totalmente iguais, conforme figura 55 abaixo:

Details		Details	
Product:	Windows Operating System	Product:	Windows Operating System
Event ID:	2	Event ID:	1
Source:	dmboot	Source:	dmboot
Version:	5.2.3790.1830	Version:	5.2.3790.1830
Message:	LDM - Volume started in failed redundancy mode	Message:	LDM - Volume started in failed redundancy mode
Explanation		Explanation	
This event indicates that the Logical Disk Manager has detected that a mirrored or RAID-5 volume started in failed redundancy mode due to configuration errors.		This event indicates that the Logical Disk Manager has detected that a mirrored or RAID-5 volume started in failed redundancy mode due to configuration errors.	
Cause		Cause	
Possible causes include:		Possible causes include:	
<ul style="list-style-type: none"> • The volumes in a mirror set are not synchronized. • The volume is fault tolerant, and multiple disk I/O errors have occurred. • A volume in the stripe set is missing. 		<ul style="list-style-type: none"> • The volumes in a mirror set are not synchronized. • The volume is fault tolerant, and multiple disk I/O errors have occurred. • A volume in the stripe set is missing. 	
User Action		User Action	
Do one or more of the following:		Do one or more of the following:	
<ul style="list-style-type: none"> • Check the status of your hardware for any failures (for example, a disk, controller card, or cable failure). • Ensure all disks are present. • Rescan disks to ensure that Plug and Play can detect all disks. • Check if any disks have failed with bad sectors and need to be replaced. • Restart the system. • Contact Microsoft Customer Service and Support. 		<ul style="list-style-type: none"> • Check the status of your hardware for any failures (for example, a disk, controller card, or cable failure). • Ensure all disks are present. • Rescan disks to ensure that Plug and Play can detect all disks. • Check if any disks have failed with bad sectors and need to be replaced. • Restart the system. • Contact Microsoft Customer Service and Support. 	

Figura 55 Eventos dmboot (2 e 1)

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Neste caso, verificamos a ocorrência de uma inconsistência na base de conhecimento da Microsoft, e ontologia apenas apresentaria o entendimento e a possível resolução do evento repetido.

De forma semelhante ao que foi apresentado na figura 50, as fontes como *save dump* (id: 1001), apresentam uma descrição, uma explicação e uma ação do usuário vagas, mas, aqui, nos remete a um contato externo com a própria Microsoft Corporation e o envio do relatório gerado pelo erro conforme figura 56 abaixo:

Details

Product:	Windows Operating System
Event ID:	1001
Source:	Save Dump
Version:	5.2
Symbolic Name:	EVENT_BUGCHECK_SAVED
Message:	The computer has rebooted from a bugcheck. The bugcheck was: %1. A dump was saved in: %2.

Explanation

The computer has restarted from a bugcheck. The event log contains details about the cause of the bugcheck.

User Action

If an error report was generated for this error, you might be able to obtain more information about the error by sending the report to Microsoft for analysis.

What to expect

Microsoft actively analyzes all error reports and prioritizes them based on the number of customers affected by the error signature contained in the report. When a report is submitted, Microsoft will try to determine the cause of the error, categorize it according to the type of issue encountered, and send you relevant information when such information is available. However, because error reports do not always contain enough information to positively identify the source of the issue, Microsoft may need to collect a number of similar reports from other customers before a pattern is discovered. Although Microsoft might not be able to provide a solution for your particular error at this time, all information submitted is used to further improve the quality and reliability of our products.

Figura 56 Eventos Save Dump (1001)

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Como vemos nesta fonte, mesmo a ontologia não é capaz de esclarecer ou resolver os termos acima citados, já que a própria empresa não registra informações suficientes para suporte ao usuário.

Fontes como *ntfs* (id: 50) e *w32time* (id: 14), apresentam uma descrição, uma explicação e uma ação do usuário, mas nos remete a uma busca dentro dos próprios *log* de eventos apresentados no programa aplicativo *Event Viewer* (Visualizar Eventos), conforme figura 57 abaixo:

Details		Details	
Product:	Windows Oper.	Product:	Windows Operating System
Event ID:	50	Event ID:	14
Source:	ntfs	Source:	w32time
Version:	5.2	Version:	5.2
Symbolic Name:	IO_LOST_DELU	Symbolic Name:	MSG_NO_DC_LOCATED
Message:	{Delayed Write file %1. The data of your computer was saved to this file elsewhere}	Message:	The time provider NtpClient was unable to find a domain controller to use as a time source. NtpClient will try again in %1 minutes.
Explanation	Windows could not save data from memory or with hardware. The data has been lost.	Explanation	Windows Time Service is configured to use the domain hierarchy to locate its time source. It could not locate a domain controller that is a suitable time source. The time service will continue to search for an acceptable domain controller. If the time service cannot locate a time source after the maximum number of attempts, the Win32Time 49 message will be logged.
User Action	If the data is still in memory, try to save it again. To determine the root cause of the problem, look in Event Viewer for messages that indicate problems with your disk or network connection.	User Action	If you are a domain administrator, it is recommended that you review the event logs on the domain controllers for any time-related messages.

Figura 57 Eventos ntfs (50) e w32time (14)

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Neste caso, a ontologia além de esclarecer os conceitos necessários, também apresentaria os relacionamentos existentes entre os logs, o que evitaria a necessidade de busca no programa aplicativo *Event Viewer* (Visualizar Eventos), em busca de causa raiz do problema.

Fontes como *Rasman* (id: 20159), apresentam uma descrição, uma explicação e uma ação do usuário concisas, mas relacionados a mais de um dispositivo ou elemento, que necessita de um maior entendimento para a sua resolução, conforme figura 58 abaixo:

Details	
Product:	Windows Operating System
Event ID:	20159
Source:	Rasman
Version:	5.2
Symbolic Name:	ROUTERLOG_DISCONNECTION_OCCURRED
Message:	The connection to %1 made by user %2 using device %3 was disconnected.

Explanation
A Point-to-Point Protocol (PPP) connection was disconnected. Possible causes include normal user-initiated disconnection or a hardware error.

User Action
Check the PPP and Rasman logs for any related events. If no related events are found, then this was probably a user-initiated disconnection and no user action is required.

Figura 58 Eventos Rasman (20159)

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Como vemos nesta fonte acima, a ontologia nos mostra que um problema ocasionado em um dispositivo, pode acabar ocasionando problemas em outros dispositivos do sistema, pois os mesmos podem estar interconectados por protocolos de rede, conforme figura 59 abaixo:

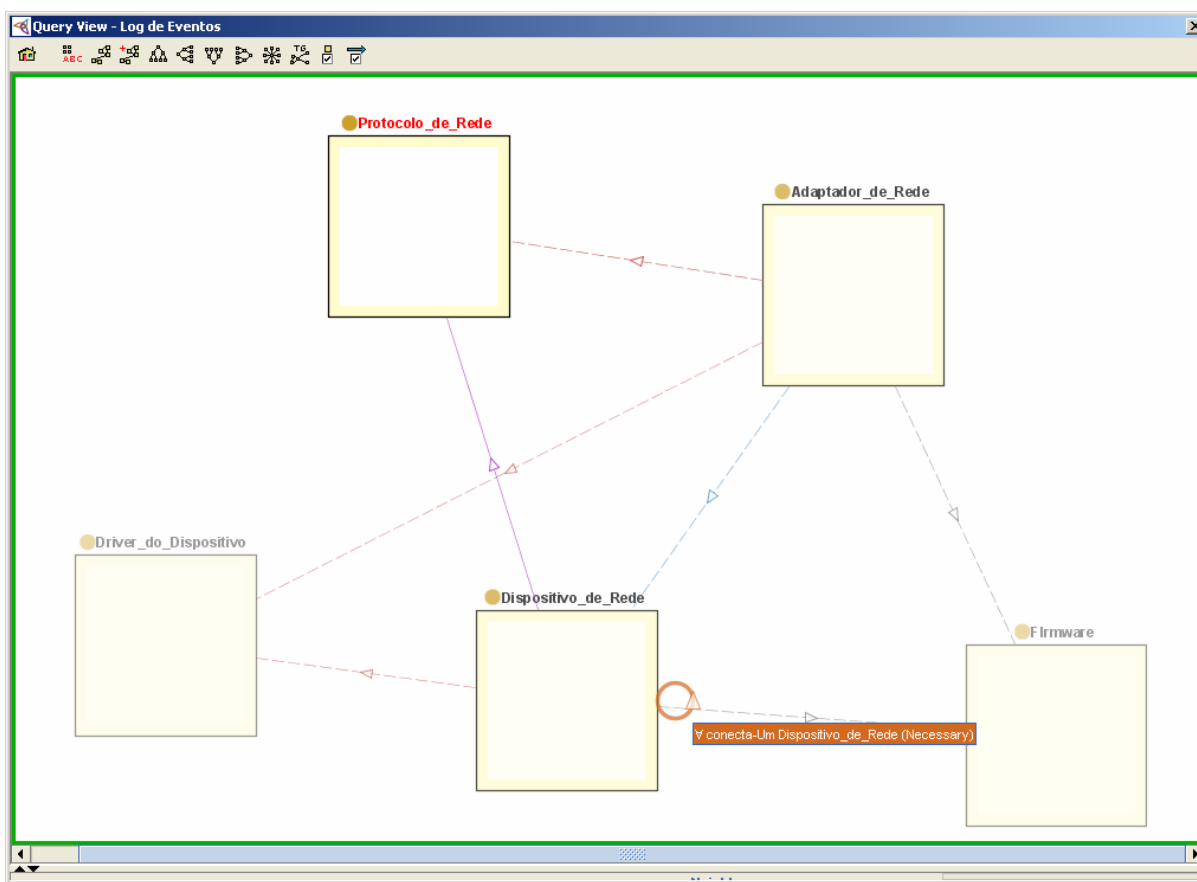


Figura 59 Ontologia parcial de Dispositivo de Rede
Fonte: Nossa autoria (Jambalaya)

Fontes como *Server* (id: 8022, 8021, 8015, 7027, 3100, 14503) e *VolSnap* (id: 11, 7, 1, 3, 26, 25, 23, 22, 20, 2, 16, 15, 14, 13, 12), apresentam uma descrição, mas não uma explicação e nem uma ação do usuário, conforme figura 60 abaixo:

Details Product: Windows Operating System Event ID: 8022 Source: server Version: 5.2.3790.1830 Message: The browser was unable to retrieve a list of domains from the browser master %1 on the network %2. The data is the error code.	Details Product: Windows Operating System Event ID: 7027 Source: server Version: 5.2.3790.1830 Message: Windows 2000 could not be started as configured. A previous working configuration was used instead.
Details Product: Windows Operating System Event ID: 8021 Source: server Version: 5.2.3790.1830 Message: The browser was unable to retrieve a list of servers from the browser master %1 on the network %2. The data is the error code.	Details Product: Windows Operating System Event ID: 3100 Source: server Version: 5.2.3790.1830 Message: The Microsoft IPv6 Developer Edition driver was started.
Details Product: Windows Operating System Event ID: 8015 Source: server Version: 5.2.3790.1830 Message: The browser has forced an election on network %1 because a Windows 2000 Server (or domain master) browser is started.	Details Product: Windows Operating System Event ID: 14503 Source: server Version: 5.2.3790.1830 Message: Dfs could not create reparse point for share %1 mapped to directory %2. The return code is in the record data.

Figura 60 Eventos Server (8022, 8021, 8015, 7027, 3100 e 14503)

Fonte:

http://www.Microsoft.com/technet/support/ee/ee_advanced.aspx

Neste caso, a ontologia esclarece o entendimento dos conceitos relacionados, informando o usuário nos casos em que o evento for apenas uma informação, mas, não apresenta resolução para outros casos.

Alguns dos eventos aqui analisados, não foram foco do entendimento ou da resolução pela ontologia devido ao tamanho da mesma, mas, fizeram parte do processo de análise durante a fase de coleta de dados.

As análises apresentaram que a ontologia em todos os eventos selecionados ajudou no entendimento dos conceitos a eles relacionados, mas, não ajudou na solução de todos, principalmente daqueles que mostraram problemas em sua estrutura ou inconsistência na base de conhecimento da empresa, relacionado aos *log* de eventos.

CONSIDERAÇÕES FINAIS e SUGESTÕES DE PESQUISA

A crescente importância que os *log* de eventos vêm assumindo nas organizações requer, cada vez mais, a disponibilidade de informações que permitam captar e solucionar com rapidez a extrema variedade de situações que eles apresentam.

A análise do conjunto de dados disponibilizado sobre logs de eventos contribuiu para o entendimento da realidade atualmente existente nos sistemas operacionais da empresa Microsoft Corporation, especificamente no Windows Server 2003 Enterprise Edition, que foi objeto de estudo coberto pela ontologia neste trabalho, e de seus processos, cuja finalidade foi a identificação das carências e o estabelecimento de classes e relacionamentos de conceitos utilizados parcialmente como base terminológica para a construção da ontologia sobre *log* de eventos que apoiou a recuperação de informações para o entendimento e a resolução dos mesmos.

Em relação às ferramentas analisadas - *site* oficial [Events and Errors Message Center](#) e o programa aplicativo *Event Viewer* (Visualizar Eventos) - os recursos atualmente disponíveis não são suficientes para o entendimento completo dos *log* de eventos pelo usuário. A informação encontra-se dispersa, ou seja, os conceitos abordados na descrição de um *log* de evento não se encontram relacionados e, em muitos casos não há informação disponível. Outro problema identificado diz respeito ao número de identificação do evento, que cumpre a função de identificá-lo de forma única. Diferentes números de identificação levaram a um mesmo conjunto de informações disponível no *site*, demonstrando uma inconsistência dessas informações.

Para maior utilização das informações sobre *log* de eventos por agentes inteligentes ou por sistemas de resolução de problemas, seria interessante, por exemplo, que este conteúdo fosse disponibilizado em XML, adotando um padrão de representação mundialmente aceito e possibilitando melhor organização e descrição das informações, numa sintaxe passível de ser

compreendida por estes agentes e sistemas. Além disto, em alguns casos, como demonstra a análise de resultados, a resolução de *logs*, implica a necessidade de recorrer a recursos externos aos disponíveis nas bases de conhecimento da Microsoft. A utilização do XML poderia auxiliar na interoperabilidade entre os recursos da Microsoft e os de outras empresas.

Quanto à modelagem da ontologia, foi possível construir uma ontologia de *log* de eventos capaz de fornecer elementos para analisar a simulação de uma interface de busca apoiada por ontologia. Na construção dessa ontologia, um dos grandes problemas identificados diz respeito à quebra do paradigma procedural, dirigido à prática, já que a ontologia encontra-se no paradigma declarativo, explicando as relações entre os conceitos necessários para a modelagem de conceitos, modelo que acaba limitado por nossa capacidade de abstração na construção efetuada.

Outra dificuldade diz respeito à possibilidade de criação de um *Checklist* procedural na ontologia, utilizando-se da noção temporal, necessária à resolução da recorrência de problemas, verificado comumente nas empresas que utilizam tecnologia.

O fato da ontologia ser modular implica a necessidade de um levantamento inicial exaustivo dos conceitos relativos a um domínio ou tarefa. A inclusão de novos conceitos, após a estruturação inicial revela que estes conceitos podem alterar por completo conceitos anteriormente definidos segundo uma linguagem semi-formal bem como a cadeia de relacionamentos já implementada.

Pela experiência de modelagem da ontologia, pode ser observado que há uma lacuna entre a teoria apresentada pela literatura e a prática da construção da ontologia, visto que as metodologias ainda são muito recentes e, portanto, existem poucas experiências de seu uso relatadas na literatura. Essas metodologias não apresentam um detalhamento dos procedimentos a serem seguidos e, somente na prática surgem dificuldades para as quais não se encontra sugestão de solução.

Nas metodologias de construção de ontologias analisadas, por exemplo, é dada pouca atenção aos aspectos conceituais propriamente ditos, isto é, não são abordadas questões relativas à análise conceitual e aos princípios a serem seguidos no estabelecimento de relacionamentos entre conceitos. Assim sendo, concluímos que deva ser útil a qualquer

construção ontológica que venha a ser desenvolvida, a busca inicial pelos termos e definições que dão suporte à estrutura ontológica, a partir dos quais outros conceitos serão identificados no sentido de mapear o domínio a ser modelado.

No tocante às interfaces gráficas do Protegé, ferramenta utilizada na modelagem da ontologia, foi possível observar que, devido ao conjunto de informações que essas interfaces são capazes de apresentar, acabam por dificultar o entendimento da estrutura conceitual da ontologia. Apesar deste trabalho ter utilizado apenas uma das interfaces - o Jambalaya – outras foram testadas no decorrer da pesquisa. Para a utilização desses modelos de apresentação numa interface final para o usuário, há necessidade de torná-los mais amigáveis, para que não chegue a ponto de excesso de informação se tornar desinformação.

Na simulação do uso da ontologia numa interface de busca de informações sobre log de eventos, de maneira geral, foi possível demonstrar que a ontologia pode auxiliar no entendimento desses *logs*, uma vez que pode nos fornecer, por exemplo, a possibilidade de identificar que componentes do sistema, hardware ou software, mais apresentaram avisos, erros ou problemas, e em quais tipos de equipamentos isso mais aconteceu. Este é um passo inicial necessário tanto para a resolução dos *log* de eventos, como para a tomada de decisão enquanto administrador de uma empresa. No entanto, para apoiar efetivamente a resolução dos mesmos, a ontologia não é suficiente.. A resolução dos logs de eventos prescinde em muitos casos de um relacionamento entre eventos antecedentes e precedentes, que nem sempre correspondem a uma estrutura conceitual abstrata, mas sim a uma situação específica da realidade. Para transpor esta realidade a um modelo ontológico seria necessário implementar um nível mais profundo de descrição ontológica que possa ser utilizado por um sistema especialista, possibilitando, dessa forma, um apoio mais efetivo à resolução dos logs de eventos.

4.1 Sugestão de trabalhos futuros

Buscando-se automatizar o processo de resolução de problemas, vemos que, uma vez modelada uma ontologia de *log* de eventos, cuja completeza demandaria uma considerável quantidade de tempo pelo seu porte, agentes de software e/ou robôs dotados de inferência,

seriam capazes de analisar e corrigir os *log* existentes nos computadores, facilitando o trabalho dos usuários e administradores dos mesmos. Neste sentido, são sugeridos os seguintes trabalhos futuros:

- Ampliar a abrangência conceitual da ontologia e transformá-la numa ontologia formal.
- Propor um modelo ontológico em XML ou RDF a ser utilizado pelas instituições para a comunicação e interoperabilidade entre seus sistemas de informação.
- Desenvolver um sistema especialista que utilize o conhecimento modelado na ontologia com o objetivo de resolver logs de eventos de maneira automática.

REFERÊNCIAS

ALMEIDA, M.; BAX M. P. *Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção*. Ci. Inf., Brasília, v. 32, n. 3, p. 7-20, set./dez. 2003. Disponível em:

<http://www.ibict.br/cionline/include/getdoc.php?id=223&article=36&mode=pdf>. Acesso em 20 Out. 2005.

ALENCAR, M. S. M. *Mecanismos de busca na Web: Uma análise da metodologia de estudos comparados*. Dissertação (Mestrado em Ciência da Informação). UFRJ/ECO-MCT/IBICT, Rio de Janeiro, 2001. Disponível em:

http://orbita.starmedia.com/~simone_alencar/arquivos/Dissertacao_Simone.pdf. Acesso em 02 Nov. 2005.

ALVARENGA, L. *A teoria do conceito revisada em conexão com ontologias e metadados no contexto das bibliotecas tradicionais e digitais*. Ci. Inf., DataGramaZero, v.2, n.6, dez/01.

Disponível em: http://www.dgz.org.br/dez01/F_I_art.htm. Acesso em 10 Mai. 2006.

_____. *Definição de publicações oficiais*. Revista da Escola de Biblioteconomia da UFMG, v.22, n.2, p.213-238, julho/dezembro 1993.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (ABNT). *NBR 6023: Elaboração de referências*. Rio de Janeiro: ABNT, 2000.

_____. *NBR 10520: Apresentação de citações em documentos*. Rio de Janeiro: ABNT, 2002.

_____. *NBR 14724: Apresentação de trabalhos acadêmicos*. Rio de Janeiro: ABNT, 2002.

BAEZA-YATES, R.; RIBEIRO-NETO B. *Modern Information Retrieval*. Addison-Wesley, 1999.

BERGMAN, M. K. *The deep Web: surfacing hidden value*. Journal of the Electronic Publishing. v.7, n.1, Aug. 2001. Disponível em: <http://www.press.umich.edu/jep/07-01/bergman.html>. Acesso em 18 Set. 2005.

BENJAMINS V. R.; PÉREZ A. G. *Knowledge-System Technology: Ontologies and Problem-Solving Methods*. 1999. Disponível em: <http://hcs.science.uva.nl/usr/richard/pdf/kais.pdf>. Acesso em 01 Nov. 2005.

BHATNAGAR, R. K.; KANAL, L. N. *Handling Uncertain Information: A Review of Numeric and Non-numeric Methods*. In: KANAL, L. N.; LEMMER, J. F. (editors), *Uncertainty in Artificial Intelligence*. Amsterdam: Elsevier Science, 1986.

BORST, P. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. PhD thesis, Tweente University, 1997. Disponível em: <http://purl.org/utwente/fid/1392> ou <http://www.ub.utwente.nl/webdocs/inf/1/t0000004.pdf>. Acesso em 25 Out. 2005.

BRACHMAN, R.; LEVESQUE, H. *Knowledge Representation and Reasoning*. Morgan Kaufmann. Maio de 2004.

BRANDÃO, W. C. *Método para Produção de Representações Conceituais a partir de Literatura Especializada utilizando a abordagem Analítico-Sintética*. Universidade Federal de Minas Gerais, 2005.

BUCHANAN, B. *Theory of library classification*. London: Clive Bingley, 1979.

CABRÉ, M. T. *La terminología: teoría, metodología, aplicaciones*. Barcelona: Antártida. 1993.

_____. *Una nueva teoría de la terminología: de la denominación a la comunicación*. Actas do VI Simposio Iberoamericano de Terminologia. Cuba (no prelo). 1998.

_____. *Una nueva teoría de la terminología: de la denominación a la comunicación*. In: _____. *La terminología: representación y comunicación*. Barcelona: Universitat Pompeu Fabra. 1999.

CAMPOS, M. L. A. *Modelização de domínios de conhecimento: uma investigação de princípios fundamentais*. Universidade Federal Fluminense. *Ci. Inf.* v. 33, n. 1. 2004. Disponível em: <http://www.ibict.br/cionline/viewarticle.php?id=77&layout=html>. Acesso em 01 Jan. 2006.

CHANDRASEKARAN, B.; JOSEPHSON, J. R. *What Are Ontologies, and Why Do We Need Them?* In: *IEEE Intelligent Systems*, 14(1):20-6, 1999. Disponível em: http://www.infofusion.buffalo.edu/conferences_and_workshops/ontology_wkshop_2/ont_ws_2_working_materials/ChandrasekaranRoleofOntology.pdf. Acesso em 29 Out. 2005.

CHAUI, M. *Convite à Filosofia*. São Paulo: Editora Ática, 1995.

CORAZZON, R. *What is ontology?*, 2002. Disponível em: http://www.formalontology.it/section_4.htm. Acesso em 03 Jun. 2005.

DAHLBERG, I. *Teoria do conceito*. *Ci. Inf.*, Rio de Janeiro, v. 7, n. 2, p. 101-107, 1978.

_____. *A referent-oriented, analytical concept theory of interconcept*. *International Classification*, v.5, n.3, p.122-151, 1978.

DAVIS, R.; SHROBE, H.; SZOLOVITS, P. *What is a knowledge representation?* *AAAI Journal. AI Magazine* (1993); 14 (1): 17-33.

- DING, Y.; FOO, S. *A review of ontology generation*. In: *Ontology Research and Development*. parte 1. Disponível em: http://homepage.uibk.ac.at/~c703205/download/01jis01_final_revision.pdf. Acesso em 16 Ago. 2005.
- _____; _____. *A review of ontology mapping and involving*. In: *Ontology Research and Development*. parte 2. Disponível em: <http://jis.sagepub.com/cgi/reprint/28/5/375>. Acesso em 22 Ago. 2005.
- DUARTE, K. C. FALBO, R. A. *Uma Ontologia de Qualidade de Software*. Mestrado em Informática, Espírito Santo Vitória. UFES. 2000. Disponível em: <http://www.inf.ufes.br/falbo/download/pub/Wqs2000.pdf>. Acessado em 29 Dez. 2005.
- FALBO, R. A. *Integração de Conhecimento em um ambiente de desenvolvimento de software*. Tese de doutorado, COPPE/UFRJ, Rio de Janeiro, Brasil, 1998.
- FÁVERO, A. J. *Sistemas Especialistas*. 2004. Disponível em: <http://www.din.uem.br/ia/especialistas/index.html>. Acesso em 29 Dez. 2005.
- FEIGENBAUM, E. A.; BARR, A. *The Handbook of Intelligence*. Vol. I, 1981.
- FEITOSA, A. L. G. *A integração entre sistemas legislativos, terminologia e Web semântica na organização e representação da informação legislativa*. Tese de Doutorado. Universidade de Brasília, 2005.
- FELBER, H. *Terminology Manual*. Paris: Unesco, 1984. 375 p.
- FENSEL D.; MOTTA E.; DECKER S.; ZDRAHAL Z. *Using Ontologies For Defining Tasks, Problem-Solving Methods and Their Mappings*. 1997. Disponível em: <http://citeseer.ist.psu.edu/rd/56441690%2C224174%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/10003/ftp:zSzzSzftp.aifb.uni-karlsruhe.dezSzpubzSzmikezSzdfzSzpaperzSzto.pdf/fensel97using.pdf>. Acesso em 28 Dez. 2005.
- FERNÁNDEZ-LÓPEZ, M. *Overview of methodologies for building ontologies*. In: *Workshop on Ontologies and Problem-Solving Methods: lessons learned and future trends*. 1999. [S. l. : s. n.], 1999. p. 4-1, 4-13. Disponível em: <http://www.lsi.upc.es/~bejar/aia/aia-web/4-fernandez.pdf>. Acesso em 12 Fev. 2006.
- FERNEDA, E. *Recuperação de Informação: Análise sobre a contribuição da Ciência da Computação para a Ciência da Informação*. Tese de doutorado. São Paulo, 2003.
- FREITAS, F. L. G. *Ontologias e a Web Semântica*. UniSantos, Santos, SP. 2003.
- GASSER, L. *Boundaries, Identity and aggregation: plurality issues in multiagent systems*. In: Eric Werner e Yves Demazeau, editors, *Decentralized Artificial Intelligence*, p.199-212, Elsevier Science, Amsterdam, NL, 1992.
- GAVA, T. B. S.; MENEZES, C. S. *Uma ontologia de domínio para a aprendizagem cooperativa*. In: XIV Simpósio Brasileiro de Informática na Educação, UFRJ, 2003.

GENESERETH, M. R.; NILSSON, L. *Logical foundation of AI*. San Francisco, Los Altos, Califórnia : Morgan Kaufman, 1987.

GEY, F. *Models in Information Retrieval*. Folders of Tutorial Presented at the 19th ACM Conference on Research and Development in Information Retrieval (SIGIR), 1992.

GIL, A. C. *Pesquisa social*. São Paulo: Atlas, 1999.

GINSBERG, M. *Essentials of Artificial Intelligence*. In: Computational Intelligence Research Laboratory (CIRL) at the University of Oregon. Morgan Kaufmann, San Mateo, California, 1993. Disponível em: <http://cirl.uoregon.edu/research/kr.html>. Acesso em 07 Mar. 2006.

GÓMEZ-PÉREZ, A.; JURISTO, N.; PAZOS, J. *Evaluation and assessment of knowledge sharing technology*. In: N. J. Mars (editor), Towards Very Large Knowledge Bases. KBKS 95. IOS Press, Amsterdam, 1995. pp., 289-296.

GRUBER T. R. *Toward principles for the design of ontologies used for knowledge sharing*. Int. J. Humam-Computer Studies, v.43, n. 5/6, 1993.

_____. *What is an Ontology?* 1996. Disponível em: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>. Acesso em 09 Jul. 2005.

GUARINO, N.; GIARETTA, P. *Ontologies and KBs, towards a terminological clarification*. 1995. Disponível em < <http://www.ladseb.pd.cnr.it/infor/Ontology/Papers/KBKS95.pdf>>. Acesso em 22 Jul. 2005.

_____. *Formal ontology, conceptual analysis and knowledge representation*. GUARINO, Nicola, POLI, Roberto (Eds). International Journal of Human-Computer Studies, v. 43, n. 5/6, 1995b. Número especial.

_____. *Formal ontology and information systems*. 1998. Disponível em: <http://www.loa-cnr.it/Papers/FOIS98.pdf>. Acesso em 01 Mar. 2006.

_____; WELTY, C.; PARTRIDGE, C. *Towards ontology-based harmonization of Web content standards*. In: Proceedings of the ER-2000 Workshop on Conceptual Modeling Approaches for e-business, 2000. Padova, Itália. Disponível em: <http://citeseer.ist.psu.edu/cache/papers/cs/26864/http:zSzzSzwww.ladseb.pd.cnr.itzSzinforzSzOntologyzSzPaperszSzeCOMO2000.pdf/guarino01towards.pdf>. Acesso em 01 Nov. 2005.

IKEDA M.; SETA K.; KAKUSHO O.; MIZOGUCHI R. *Task ontology: Ontology for building conceptual problem solving models*. Disponível em: <http://www.ei.sanken.osaka-u.ac.jp/pub/ikeda/ikeda-KAIS98.pdf>. Acesso em 01 Mar. 2006.

ISCOE, N.; TAM, K.; LIU, Z. *A Framework for Understanding and Discussing Domain Modeling*. In: Proceedings of the ICSE 1991 Domain Modeling Workshop. Austin Laboratory for Software Engineering and Computer Science. May, 1991.

ISO. *Terminologia – princípios e métodos*. ISO WD 704.1:1996. 46p.

- KENSKI, V. M. *Tecnologias e Ensino Presencial e a Distância*. Campinas: Papyrus, 2003.
- KIM, H.; SEO, J. *A reliable indexing method for a practical QA system*. In: 19th International Conference on Computational Linguistics, 2002. Disponível em: <http://www.isi.edu/~cyl/wsqa-coling2002/program.html>. Acesso em 21 Fev. 2006.
- KRIEGER, M. G. *Terminology revisited*. DELTA. [online]. vol.16, nº.2, p.209-228. 2000. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-44502000000200001&lng=en&nrm=iso. Acesso em: 09 Jan. 2006.
- LÉVY, P. In: *Do Caos à Inteligência Artificial*. São Paulo: Editora Unesp, 1993
- LIMA-MARQUES, M. *Ontologias*. 2004.
- LUSTOSA, V. G. *O estado da arte em inteligência artificial*. Revista digital da CVA-Ricesu. V.2, n.8, set., 2004. Disponível em: http://www.ricesu.com.br/colabora/n8/artigos/n_8/pdf/id_03.pdf. Acesso em 01 Dez. 2005.
- LUSTOSA, P. A. *OWL e Protégé: estudo e aplicação de conceitos para exemplificação da definição da camada de esquema da Web Semântica em um determinado domínio*. Trabalho de Conclusão de Curso. Curso de Sistemas de Informação. Centro Universitário Luterano de Palmas, Palmas – TO, 2003.
- MAULDIN, M. L. *Conceptual Information Retrieval: A case study in adaptive parsing*. Kluwer, Norwell, 1991.
- MANER, M. *The Research Process: a complete guide and reference writers*. 2 ed., McGraw-Hill, 2001.
- MARCONDES, D. *Dicionário oxford de filosofia*. Tradução D. Murcho et al. Rio de Janeiro : Jorge Zahar, 1997.
- MCGUINNESS, D. L.; WRIGHT, J. *Conceptual Modeling for Configuration: A Description Logic-based Approach*. Artificial Intelligence for Engineering Design, Analysis, and Manufacturing - special issue on Configuration. 1998.
- MEDEIROS, M. B. B. *Tratamento Automático de Ambigüidades na Recuperação da Informação*. Universidade de Brasília. Brasília. 1999.
- MERRIAM-WEBSTER; GOVE, P. B. *Webster's Third New International Dictionary*. Unabridged. New York: Merriam-Webster, 2002. 2.783p.
- MICROSOFT. *Microsoft Computer Dictionary*. Microsoft Corporation, 15 ed., 2002.
- MINSKY, M. A. *A Framework for Representing Knowledge*. McGraw-Hill, New York, 1975.
- MOREIRA, A.; ALVARENGA, L.; OLIVEIRA, A. P. *O nível do conhecimento e os instrumentos de representação: tesouros e ontologias*. DataGramZero. Ci. Inf., v.5, n.6, dez, 2004.

MUSEN, M. A. *Dimensions of knowledge sharing and reuse*. Computers and Biomedical Research 25: 435-467. 1992.

NEVES, J. L. *Pesquisa Qualitativa - características, usos e possibilidades*. Caderno de pesquisas em administração, São Paulo, v.1, n.3, 2 sem., 1996.

NEWELL, A. *The Knowledge Level*. Artificial Intelligence. 18, p.87-127, Jan, 1982.

NOLETO, D. A. *Utilização da API JENA e do Padrão RDF para a estruturação de um portal para manipulação de Imagens*. Trabalho de Conclusão de Curso. Curso de Bacharel em Sistemas de Informação. Centro Universitário Luterano de Palmas – CEULP/ULBRA, Palmas, 2003.

NOY, N. F.; McGUINNESS, D. L. *Ontology development 101: a guide to creating your first ontology*. In: Technical Report SMI-2001-0880. Stanford Medical Informatics, Stanford University, Stanford, CA, 2001. Disponível em: http://protege.stanford.edu/publications/ontology_development/ontology101.pdf. Acesso em 01 Ago. 2005.

OGDEN, C. K.; RICHARDS, I. A. *The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism*. Londres: Routledge & Kegan Paul, 1960.

OLIVEIRA, K. *Modelo para construção de ambientes de desenvolvimento de software orientados a domínio*. 1999. 222 f. Tese (Doutorado em Engenharia de Sistemas e Computação). COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro.

OWL WEB ONTOLOGY LANGUAGE OVERVIEW. Janeiro de 2006. Disponível em: <http://www.w3.org/TR/owl-features/>. Acesso em: 01 Mai. 2006.

OWL WEB ONTOLOGY LANGUAGE – XML Presentation Syntax. Janeiro de 2006. Disponível em: <http://www.w3.org/TR/owl-xmlsyntax/#subsec-struct>. Acesso em: 07 Mai. 2006.

OWL WEB ONTOLOGY LANGUAGE USE CASES AND REQUIREMENTS. Janeiro de 2006. Disponível em: <http://www.w3.org/TR/2004/REC-webont-req-20040210/#onto-def>. Acesso em: 12 Jan. 2006.

OWL WEB ONTOLOGY LANGUAGE GUIDE. Janeiro de 2006. Disponível em: <http://www.w3.org/TR/owl-guide/>. Acesso em: 12 Jan. 2006.

OWL WEB ONTOLOGY LANGUAGE REFERENCE. Janeiro de 2006. Disponível em: <http://www.w3.org/TR/owl-ref/>. Acesso em: 12 Jan. 2006.

PAVEL, S.; NOLET, D. *Précis de Terminologie*. Travaux publics et services gouvernementaux. Canadá. 2001.

PC Week. 2000. *DAML could take search to a new level*. USA. Fevereiro, 2000.

- PEARL, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc., 1988.
- POLI, R. *Framing ontology*. 2001. Disponível em: <http://www.formalontology.it/Framingfirst.htm>. Acesso em 14 Set. 2005.
- PRÓ-PESQUISA. <http://www.propesquisa.com.br>. Acesso em 15 Set. 2005.
- PROTÉGÉ (2000). *The Protege Project*. <http://protege.stanford.edu>. Acesso em 23 Set. 2005.
- QUILLIAN, R. *Semantic Memory*. In: M. Minsky (ed) *Semantic Processing*. MIT Press, Cambridge, MA, 1968.
- RDF VOCABULARY DESCRIPTION LANGUAGE 1.0: RDF SCHEMA*. Janeiro de 2006. Disponível em: <http://www.w3.org/TR/rdf-Schema/>. Acesso em: 12 Jan. 2006.
- RESOURCE DESCRIPTION FRAMEWORK (RDF): CONCEPTS AND ABSTRACT SYNTAX*. Janeiro de 2006. Disponível em: <http://www.w3.org/TR/rdf-concepts/>. Acesso em: 12 Jan. 2006.
- RICH, E. *Inteligência Artificial*. São Paulo. McGraw-Hill. 1988.
- RIJSBERGEN, C. J. *Information Retrieval*. Department of Computing Science. University of Glasgow. 1979.
- RIOS, J. A. *Ontologias: alternativa para a representação do conhecimento explícito organizacional*. In: Proceedings CINFOM - Encontro Nacional de Ciência da Informação VI, Salvador, Bahia. 2005.
- RONDEAU, G. *Introduction à la terminologie*. 2.ed. Québec, Gaëtan Morin. 1984.
- ROY, D. *Conhecimento versus Representação do Conhecimento*. 2004.
- RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A modern approach*. New Jersey: Prantice Hall, 1995.
- SAGER, J. C. *Curso práctico sobre el procesamiento de la terminología*. Madrid: Fundación G.S. Ruipérez. 1993.
- SANTANA, F. *Visualizar Eventos*. 2005. <http://www.juliobattisti.com.br/fabiano/artigos/visualizadoreventos.asp>. Acesso em 01 Jun. 2005.
- SCHOEN, E. *Active Assistance for Domain Modeling*. In: Proceedings of the ICSE 1991 Domain Modeling Workshop. Austin Laboratory for *Software Engineering and Computer Science*. Pp. 154-161. May, 1991.

SCHUWARTZ, C. A. *Restructuring serials management to generate news resources and services: with commentaries on restructurings at three institutions*. Coll. Res. Libr., Chicago, v.59, p.115-124, 1998.

SMITH B. C. *Reflection and Semantics in a Procedural Language*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1982.

SOWA, J. F. *Building, sharing and merging ontologies*. Tutorial. [S.l. : s. n.], 1999. Disponível em: <http://users.bestweb.net/~sowa/ontology/ontoshar.htm>. Acesso em 25 Set. 2005.

_____. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks Cole Publishing Co., Pacific Grove, CA, 2000.

_____. *Semantic networks*. 2002. Disponível em: <http://ww.jfsowa.com/pubs/semnet.htm>. Acesso em 25 Set. 2005.

TATIBANA, C.; KAETSU, D. *Página de Redes Neurais*. 2002. Disponível em: <http://www.din.uem.br/ia/neurais>. Acesso em 29 Set. 2005.

USCHOLD, M.; GRUNINGER, M. *Ontologies: principles, methods and applications*. In: The Knowledge Engineering Review, v. 11(2):93--136, jun. 1996. Disponível em: <http://citeseer.ist.psu.edu/rd/13915933%2C40296%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/3214/http:zSzzSzwww.cm.cf.ac.ukzSzUserzSzJ-C.PazzagliazSzReferenceszSzart:Uschold-96.pdf/uschold96ontologie.pdf>. Acesso em 14 Dez. 2005.

USCHOLD, M.; JASPER, R. *A Framework for Understanding and Classifying Ontology Applications*. In: Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods(KRR5), Stockholm, Sweden, 1999. Disponível em: <http://citeseer.ist.psu.edu/rd/68716753%2C257410%2C1%2C0.25%2CDownload/http://citeseer.ist.psu.edu/cache/papers/cs/11273/http:zSzzSzsunsite.informatik.rwth-aachen.dezSzPublicationszSzCEUR-WSzSzVol-18zSz11-uschold.pdf/uschold99framework.pdf>. Acesso em 23 Out. 2005.

WELTY, C. A. *An Integrated Representation for Software Development and Discovery*. Dissertação, Set., 1996. Disponível em: <http://www.cs.vassar.edu/faculty/welty/papers/phd/>. Acesso em 14 Out. 2005.

WOODS. W. A. *What's in a link: Foundations for semantic network*. In: Bobrow, D. G.; Collins, A. (eds.) Representation and Understanding: Studies in Cognitive Science. Academic Press, New York. 1975.

WOOLDRIDGE, M. *Intelligent agents*. In: Weiss, G., ed., Multiagent Systems - A modern approach to Distributed Artificial Intelligence. The MIT Press, 1999.

WRIGHT, S. E.; BUDIN, G. (organizadores) *Handbook of Terminology Management*. Vol. I. Basic aspects of Terminology Management. Amsterdam/Philadelphia: John Benjamins Publishing Company. 1997. 370 p.

WÜSTER, E. *Introducción a la teoría general de la terminología y a la lexicografía terminológica*. Barcelona: Universitat Pompeu Fabra, 1998.

VICKERY, Brian Campbell. *Facet classification: a guide to the construction and use of special schemes*. London: Aslib, 1960. _____ . *Classification and Indexing in Science*. London: Butterworths, 1975.

YIN, R. *Case Study Research: Design and Methods, Newbury Park*. SAGE, 1994.