

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA E  
GESTÃO DO CONHECIMENTO - PPGECC**

**CÉSAR RAMIREZ KEJELIN STRADIOTTO**

**MÉTODO DE CONSTRUÇÃO DE ONTOLOGIAS  
MULTILINGUES COM ASSOCIAÇÃO DE CONCEITOS A  
OBJETOS EM ESPAÇO 3D**

Tese submetida ao Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento da Universidade Federal de Santa Catarina para a obtenção do Grau de Doutor em Engenharia e Gestão do Conhecimento  
Orientador: Prof. Dr. Hugo César Hoeschl – Post. Doc.

Florianópolis

2011

Catálogo na fonte pela Biblioteca Universitária  
da  
Universidade Federal de Santa Catarina

S895m Stradiotto, César Ramirez Kejelin  
Método de construção de ontologias multilingues com  
associação de conceitos a objetos em espaço 3D [tese] / César  
Ramirez Kejelin Stradiotto ; orientador, Hugo Cesar Hoeschl. -  
Florianópolis, SC, 2011.  
283 p.: il., tabs.

Tese (doutorado) - Universidade Federal de Santa Catarina,  
Centro Tecnológico, Programa de Pós-Graduação em Engenharia  
e Gestão do Conhecimento.

Inclui referências

1. Engenharia e gestão do conhecimento. 2. Ontologias  
(Sistema de recuperação da informação). 3. Agentes  
inteligentes (Software). 4. Web semântica. I. Hoeschl, Hugo  
Cesar. II. Universidade Federal de Santa Catarina. Programa de  
Pós-Graduação em Engenharia e Gestão do Conhecimento. III.  
Título.

CDU 659.2

CÉSAR RAMIREZ KEJELIN STRADIOTTO

**MÉTODO DE CONSTRUÇÃO DE ONTOLOGIAS  
MULTILINGUES COM ASSOCIAÇÃO DE CONCEITOS A  
OBJETOS EM ESPAÇO 3D**

Esta Tese foi julgada adequada para obtenção do Título de “Doutor em Engenharia e Gestão do Conhecimento”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento - PPGE GC

Florianópolis, 12 de Setembro de 2011.

---

Prof. Maurício Selig, Dr.  
Coordenador do Curso

**Banca Examinadora:**

---

Prof<sup>ª</sup>. Tânia Cristina D’Agostini Bueno, Dra. (UFSC /SC/Brasil)  
Examinadora

---

Prof. Rogério Cid Bastos, Dr. (UFSC/SC/Brasil)  
Examinador

---

Prof. Aires José Rover, Dr. (UFSC/SC/Brasil)  
Examinador

---

Prof. Jesús Cardeñosa Lera, PhD, (UPM/Madrid/Espanha)  
Examinador Externo

---

Prof<sup>ª</sup>. Carolina Gallardo Perez, PhD, (UPM/Madrid/Espanha)  
Examinadora Externa

---

Prof. Hugo César Hoeschl, Post-Doc, (UFSC/SC/Brasil)  
Orientador



Dedico este trabalho à minha  
mãe, Nídia Kejelin.

Dedico também ao meu pai,  
Alonso Bonifácio Cáceres  
Gonzáles.

E à minha avó, Júlia Rosalina  
Kejelin, e ao meu avô Osvaldo  
Souza.

Gostaria que eles estivessem  
aqui para que pudessem ver até  
onde consegui chegar.



## AGRADECIMENTOS

Ao Sr. Osny Raffs Júnior, que, com esforço extremo, me auxiliou, e me orientou, com precisão, sobre a redação correta de vários pontos desta tese.

Aos Srs. Pedro A. S. de Carvalho Júnior e Luiz A. P. Athayde Filho, da empresa Centrais Elétricas de Santa Catarina S/A. (CELESC), pelo apoio financeiro durante a realização das pesquisas e redação desta tese.

À Universidade Federal de Santa Catarina, por ter me acolhido como seu aluno, como graduando, mestrando e doutorando.

Aos professores do Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, pelas disciplinas e orientações ministradas.

Aos integrantes do Instituto de Governo Eletrônico, Inteligências e Sistemas – i3G, Srs. Thiago Paulo da Silva de Oliveira, Sonali de Paula Molin Bedin, Ângela Iara Zotti, Claudia de Oliveira Bueno e Vinicius Oliveira Mirapalheta, pela atenção, dedicação e apoio durante esta pesquisa.

Aos professores Jesús Cardeñosa Lera, Roberto Giordano Lerena, Carolina Gallardo Perez, e Igor Boguslavski, pelas novas idéias.

Aos professores Hugo César Hoeschl e Tânia Cristina D’Agostini Bueno, pela extrema paciência, incentivo, apoio, carinho e proteção.

A Nédio Kejelin, meu tio, pelas brincadeiras e companhias na infância. Vou seguir o teu exemplo, e em meu coração sempre permanecer uma criança.

A Névio Kejelin, meu tio, por me mostrar um pouco da vida profissional ainda durante minha graduação.

A Nério Kejelin, meu tio, que me cedeu os primeiros documentos, revistas e projetos, que encaminharam a minha atividade acadêmica e profissional.

À Níria Kejelin, minha tia, que sempre me teve com especial carinho.

A Osvaldo Souza, meu avô.

A Julia Rosalina, minha avó.

A Alonso Bonifácio Cáceres Gonzales, meu pai.

A Nídia Kejelin, minha mãe: O principio de tudo o que fiz até agora.

Tenho um pedaço de cada um de vocês dentro de mim, que vou carregar pelo restante de minha vida...





Aos que acreditam que o mundo é cruel, digo que existem pessoas que estão sempre dispostas a incentivar e socorrer no momento exato de quando precisamos. E não são poucos. Tenho comprovado isto na prática, ao longo do tempo...

O que é mais importante do que ver o sorriso de satisfação e gratidão das pessoas que receberam a tua ajuda?



## RESUMO

Com o crescimento no volume de documentos disponíveis na Web, são necessários métodos para gerenciar o conhecimento disponível nesses documentos. Entre os métodos de gestão, destaca-se a construção e aplicação de ontologias. Torna-se então necessária a aplicação de um meio descritivo mais universal para a apresentação de um conceito e de suas relações semânticas, um meio que seja compreensível por qualquer pessoa, em qualquer idioma. Torna-se necessário também o reuso de duas tecnologias já existentes, na tentativa de universalização de conceitos e relações semânticas: a *Universal Networking Language (UNL)* e a *Unified Modeling Language (UML)*. Portanto, o objetivo deste trabalho é desenvolver um método de construção e representação visual de ontologia multilíngue e modelar um editor usando recursos gráficos em espaços 2D e 3D. A proposta da tese consiste em uma metodologia de construção de ontologias composta de palavras universais, associadas a objetos no espaço 3D, possibilitando a conversão entre instâncias de ontologias em vários idiomas. Para a validação desta tese, foram programados três sistemas gráficos compartilhados de construção de ontologias multilíngües, na forma de redes de relacionamento e envoltórias, utilizando os conceitos de Palavras Universais do projeto UNL como ponte para equivalência entre os conceitos dessas ontologias, e usando critérios de usabilidade durante a construção do editor. Este trabalho utilizou a pesquisa qualitativa como abordagem para análise e interpretação dos dados. As fontes de informações desta pesquisa foram obtidas por intermédio da pesquisa bibliográfica, em que os dados estão disponíveis em livros, publicações e periódicos. Quanto aos objetivos, esta pesquisa caracterizou-se como descritiva. A proposta é seguida de um projeto e programação de um aplicativo de construção de ontologias. Posteriormente, para fins de melhoramento no uso deste aplicativo, são levantados os critérios de usabilidade de software. Após o levantamento, três aplicativos de construção de ontologias são comparados com o uso desses critérios, por meio de entrevistas com usuários: a) O primeiro aplicativo apresenta ontologias com conceitos associados a objetos no espaço 3D, na forma de redes de relacionamento e envoltórias; b) Um aplicativo que apresenta ontologias como redes de relacionamento e envoltórias no espaço 2D; c) Um aplicativo que apresenta ontologias construídas na forma de listas indentadas, formulários e tabelas. Após a comparação dessas ferramentas, foram apontadas vantagens e desvantagens no uso de cada um dos três modelos de editores. Em seguida, foram levantados

os relatos científicos anteriores referentes a comparações entre visões de ontologias, comparando opiniões concordantes e discordantes dos autores. Concluiu-se que os resultados foram bem-sucedidos na representação de conceitos multilíngües associados a objetos, representados tanto no espaço 2D quanto no espaço 3D. Foram observados também resultados bem sucedidos nos modelos matemáticos e estruturais que fundamentaram as aplicações de construção de ontologias.

**Palavras-chave:** Método de construção de ontologias multilíngües, ferramentas de visão e construção de ontologias, Visão de redes de ontologias no espaço 2D e 3D.

## ABSTRACT

With the growing volume of documents on world wide web, it may be necessary methods for management of the available knowledge on these documents. Among these management methods, it is highlighted the ontology construction and application. So, it became needed the application of a more universal descriptive media for the presentation of a given concepts and its semantic relations, some mean which can be comprehensible by any people, in any language. It also became necessary the reuse of two technologies already existent, on the attempt for universalization of concepts and semantic relations: the *Universal Networking Language (UNL)* and the *Unified Modeling Language (UML)*. So, the objective of this work is the development of a method for construction and visual representation of multilingual ontologies, and the modeling of an editor, using graphic resources on 2D and 3D spaces. The proposal of this doctoral thesis is about a methodology for ontology construction, composed by universal words, associated to objects in a 3D space, turning possible the conversion of ontologies among different languages. For the validation of this thesis, it has been programmed three shared graphic system for construction of multilingual ontology instances, in the form of relationship networks, and envelopes, using the concepts of UNL's Universal Words, as a bridge for the equivalency between these ontologies' concepts, using usability criteria to optimize the construction of these three editors. This work used qualitative research as the approach for data analysis and interpretation. The sources for information were obtained by means of bibliographic research, where information was available in books, publications and periodic. For the objectives, this research has been characterized as a descriptive one. The proposal is followed by a project and programming of an application for ontology construction. After that, for optimization purposes, it there has been surveyed some software usability criteria. After the survey, three application for ontology construction were compared, using these criteria, by means of user interview: a) The first application presents ontology instances with concepts associated to objects on 3D space, on the form of relationship networks and envelopes; b) An application which presents ontology instances as relationship networks and envelopes on 2D space; c) An application which presents ontology instances on the form of indented lists, forms and tables. After the comparison among these three tools, it was pointed advantages and disadvantages on the use of any of the three

ontology editor models. After that, it was surveyed anterior scientific reports about previous comparisons among ontology visualizations, comparing concordant and discordant opinions between authors. It was concluded that the results were well-succeeded on the representation of multilingual concepts associated to objects, represented on 2D and 3D spaces. It was observed also well succeeded results on the mathematical and structural models which grounded the applications for ontology construction.

**Keywords:** Method for multilingual ontology construction, ontology vision and construction tools, Visualization of Ontology Networks on 2D and 3D space



## LISTA DE FIGURAS

Figura 1:Diagrama mostrando a interface de entrada de um documento em linguagem natural, escolha de idiomas de retorno (superior) e listagem de documentos similares agrupados pelos idiomas escolhidos (inferior). .....	10
Figura 2:Diagrama explicando uma visão simplificada do método de procura de documentos multilíngües.....	11
Figura 3:Diagrama detalhando o método de procura de documentos multilíngües. ....	13
Figura 4: <i>Corpi</i> contendo os documentos em vários idiomas, anotados pelas ontologias do modelo de BUENO (2008), com o dicionário de palavras universais ao centro. ....	14
Figura 5:Anotação do texto de entrada do usuário (em Português), pela ontologia feita no seu idioma, e procura de termos equivalentes (em verde), nas ontologias dos outros dois idiomas (Ex: Espanhol e Inglês), com auxílio do dicionário de UW's da UNL.....	15
Figura 6:Etapa de captura de documentos. ....	17
Figura 7:Diagrama representando a etapa de seleção / construção de termos. ..	19
Figura 8:Diagrama representando a etapa de construção de relações semânticas. ....	21
Figura 9:Diagrama representando a etapa de construção de palavras universais. ....	23
Figura 10:Etapa de verificação de equivalência entre a ontologia de UW's e a ontologia em um dado idioma. ....	25
Figura 11:Interface do Sistema IATE de pesquisa de termos em vários idiomas. IATE(2008). ....	30
Figura 12:Página web do Universal Words Dictionary. ....	31
Figura 13:Primeira versão do modelo de domínio. BUENO (2008).....	46
Figura 14:Segunda versão do modelo de domínio. BUENO (2008).....	46
Figura 15:Modelo de domínio aplicado no Editor Multilíngüe. BUENO (2008). ....	46
Figura 16:Interface web da ferramenta OntoKEM para construção de ontologias. RAUTEMBERG et. al. (2008).....	54
Figura 17:Telas de pesquisa e de resultados do Sistema Mongee.....	59
Figura 18:Telas de pesquisa, freqüência de resultados, resultados e legenda do Sistema Ontojuris. ....	59
Figura 19:Telas de pesquisa, freqüência de resultados, resultados e legenda do Sistema Ontoweb.....	60
Figura 20:Telas de pesquisa, freqüência de resultados, resultados e legenda do Sistema SERPI. ....	60
Figura 21:Interface do plug-in OWLViz no Protègè. ....	61
Figura 22:Determinação de condições para as classes, utilizando lógica quantificacional. ....	62



Figura 23:Ferramenta GROWL para visão de ontologias OWL e DL, na forma de redes de relacionamento.....	65
Figura 24:Comparativo entre tarefas de ontologias e métodos de visão. Fonte: KATIFORI et al. (2007). .....	80
Figura 25:Vantagens e desvantagens. Fonte: KATIFORI et al. (2007). .....	81
Figura 26:Página do Cluuz.....	82
Figura 27:Página do Cluuz. ....	83
Figura 28:Redes de relacionamentos de resultados em clusters, na página da máquina Cluuz.....	84
Figura 29:Interface de aplicativo de visão de dados de telefonia e transações financeiras em redes criminais.....	85
Figura 30:Página Web da Prefuse Toolkit, uma biblioteca para criação de soluções de visão de informação.....	86
Figura 31:Visor de ontologias baseado na API Java 3D. Fonte: ONTVIS (2010). .....	87
Figura 32:Modelo de Klein.....	88
Figura 33:Modelo de Disco de Poincaré.....	88
Figura 34:Primeira versão do modelo de domínio proposto para o Editor de Ontologias Multilíngües. ....	97
Figura 35:Segunda versão do modelo de domínio para o Editor de Ontologias Multilíngües.....	99
Figura 36:Modelo estrutural da framework RRC, montada sobre a Borland VCL .....	107
Figura 37:Representação gráfica da primeira versão da classe Aresta.....	112
Figura 38:Classe visual SuperAresta, criada para conter várias instâncias da classe geométrica Aresta.....	113
Figura 39:Sistema de coordenadas da tela e de componentes visuais da Borland VCL .....	114
Figura 40:Nove possibilidades para o cálculo das coordenadas de uma aresta. ....	115
Figura 41:Representação gráfica do cálculo das coordenadas cartesianas tendo como base o eixo da aresta entre dois vértices.....	119
Figura 42:Determinação das coordenadas de um ponto no espaço do eixo entre os dois vértices.....	120
Figura 43:Determinação das coordenadas dos pontos de uma polilinha no espaço do eixo entre os dois vértices. ....	121
Figura 44:Elementos formadores do eixo principal da extremidade da aresta.....	122
Figura 45:Posicionamento de pontos do desenho da extremidade de uma aresta. ....	123
Figura 46:Envoltórias com formato intermediário entre quadrado e octogonal. ....	126
Figura 47:Princípio das equações para os segmentos de reta da envoltória quadrada. ....	127
Figura 48:Segmentos de reta da envoltória octogonal, formados pelo cruzamento das retas do grupo da Equação 10. ....	128

Figura 49:Primeira versão do editor de ontologias. Stradiotto et. al. (2006). ...	130
Figura 50:Aplicativo de construção de ontologias 2D, com o processo de importação de arquivos Microsoft Excel. ....	132
Figura 51:Elementos anexados em vértices e arestas. ....	133
Figura 52:Demonstração de formação de conjuntos de vértices e arestas a partir de arquivos Microsoft Excel. ....	134
Figura 53:Protótipo com demonstração de múltiplas arestas entre vértices. ...	135
Figura 54:Interface do editor mostrando uma janela com vários níveis de domínios. ....	137
Figura 55:Interface de uma das janelas do editor de ontologias multilíngües, mostrando os termos em três idiomas (Português, Espanhol e Inglês, em amarelo), palavras universais (em azul), as relações e as envoltórias representando os domínios. ....	138
Figura 56:Interface exemplificando termos (conectados por palavras universais) dentro de domínios, em vários níveis. ....	139
Figura 57:Efeito do componente <i>heavyweigh</i> do Java 3D sobre componentes <i>lightweigh</i> do Swing. (Sobreposição do ambiente 3D sobre o menu do aplicativo). ....	142
Figura 58:Imagem de protótipo para teste de sobreposição de várias janelas contendo seus respectivos ambientes 3D. ....	143
Figura 59:Resultados do segundo experimento, sobre posicionamento de câmera, navegação, textura, lado visível de uma face, e aplicação de Luz. ....	143
Figura 60:Amostragem de $2 \times 320 \times 320 = 204.800$ quadrados, renderizados em tempo real, pelo Java 3D. ....	144
Figura 61:Teste de desempenho com duas janelas, cada uma com 3200 quadrados com oito texturas diferentes, e 1600 arestas. As arestas foram desenhadas com um degradê entre as cores branca e amarela. ....	145
Figura 62:Protótipo com seis janelas com o mesmo grupo de arestas, com suas respectivas setas, cores e espessuras. ....	146
Figura 63:Interface que permite ao usuário escolher uma imagem presente em um sistema de arquivos, e armazená-la em uma base de dados <i>Postgres</i> . ....	147
Figura 64:Conjunto de vértices com texturas carregadas a partir de base de dados. ....	148
Figura 65:Protótipo do editor mostrando várias instâncias de um mesmo vértice experimental com o seu rótulo. ....	149
Figura 66:Disposição helicoidal dos termos de ontologias ainda criptografados, a partir da base de dados. ....	150
Figura 67:Posicionamento somente de termos, dado o domínio escolhido pelo usuário, carregados no ambiente 3D. ....	151
Figura 68:Tentativa inicial de conectar pares de vértices através de arestas. ...	152
Figura 69:Exemplo de arestas personalizadas, com o posicionamento de seus pontos resultando em um formato com uma quebra central (formato em "v"). ....	153
Figura 70:Demonstração do uso de setas nas extremidades das arestas. ....	154
Figura 71:Múltiplas arestas com o problema da constrição. ....	155

Figura 72:Construção das múltiplas arestas corrigida. ....	156
Figura 73:Exemplo de arestas conectando arestas. ....	157
Figura 74:Exemplo de arestas conectando arestas com vértices. ....	158
Figura 75:Poliedros de Arquimedes.....	159
Figura 76:Envoltória rombicuboctaédrica transparente (contendo um cubo colorido em seu interior para facilitar a identificação de suas faces e pontos constituintes).....	160
Figura 77:Exemplo bem comportado de envoltória de base rombicuboctaédrica. Sua disposição está ao redor do centro de cada cubo.....	160
Figura 78:Exemplo mal sucedido do mesmo modelo de envoltória. ....	161
Figura 79:Experimento com seis janelas contendo uma envoltória de base rombicuboctaédrica, sobre vértices posicionados aleatoriamente ao redor de um ponto. ....	162
Figura 80:Combinação de envoltórias de base rombicuboctaédrica (representando os domínios) e vértices (representando os termos) no espaço 3D. ....	163
Figura 81:Exemplos de envoltórias paralelepípedicas. ....	164
Figura 82:Resultado do processo de criação de vértice (termo nomeado como "TESTE") nas coordenadas (0, 0, 0). ....	167
Figura 83:Tela do aplicativo com duas janelas mostrando o resultado do processo de criação de arestas.....	168
Figura 84:Janelas lado a lado mostrando a construção de um tronco de cone no 3D Studio MAX, e a inclusão desta geometria no ambiente 3D do editor de ontologias multilíngües. ....	170
Figura 85:Interface do aplicativo VRMLPad, para a edição de geometrias no formato VRML97. ....	171
Figura 86:Conteúdo exemplo de arquivo de configuração de arestas. ....	172
Figura 87:Exemplos de arestas parametrizadas. ....	172
Figura 88:Modelo de domínio sugerido para o editor de ontologias.....	175
Figura 89:Exemplo de representação das relações semânticas do modelo de Bueno (2005, 2008) como relações do Diagrama de Classes da UML.....	176
Figura 90:Representação visual das instâncias de ontologias, com as relações no formato UML, e com domínios. ....	177
Figura 91:Experimentação de carga de arquivo VRML contendo um relevo improvisado da cidade de Florianópolis. ....	179
Figura 92:Exemplo mostrando múltiplos objetos 3D por janela.....	180
Figura 93:Experimento de funcionamento do editor 3D de ontologias multilíngües rodando com <i>Applet</i> em três navegadores Web diferentes: Microsoft Internet Explorer 8, Mozilla Firefox 3.6 e Google Chrome 6. ....	181



## LISTA DE QUADROS

Quadro 1: Exemplo de método para construção de UW's, a partir do conteúdo de um documento da Web. ....	42
Quadro 2: As três premissas da metodologia Engenharia da Mente. Fonte: BUENO (2008). ....	49
Quadro 3: Comparativo de ferramentas de construção colaborativa. Fonte: Noy et. al. 2008. ....	67
Quadro 4: Recursos das ferramentas. Fonte: NOY et. al. (2008). * Recursos disponíveis em algumas versões das ferramentas, mas não na versão usada no evento. ....	67
Quadro 5: Equivalência entre documentos categorizados de arquivo e recursos de ontologia. ....	76
Quadro 6: Tipos de visualização, separadas em 2D e 3D. ....	78
Quadro 7: Visão completa da tabela de Fatores de Design. Fonte: Bresciani et. al. (2008). ....	90
Quadro 8: Fatores de Design: Descrição, Dimensões Similares, Citações de Praticantes, e Implicações de Design. Fonte: Bresciani et. al. (2008). ....	91
Quadro 9: Fatores de Design: Contrapartida, Escala, Exemplo de Alto Valor, Exemplo de Baixo Valor. Fonte: Bresciani et. al. (2008). ....	92
Quadro 10: Dimensões Cognitivas. Fonte: Blackwell et. al. (2001). ....	93
Quadro 11: Formas de representação selecionadas. ....	101
Quadro 12: Descrição das principais classes estruturais da framework. ....	108
Quadro 13: Variáveis usadas para o arraste dos componentes visuais com o uso do Mouse. ....	110
Quadro 14: Variáveis usadas para o desenho de uma aresta isolada. ....	116
Quadro 15: Variáveis usadas para o cálculo dos pontos das extremidades das arestas. ....	124
Quadro 16: Significado das relações do Diagrama de Classes da UML. ....	173
Quadro 17: Critérios de Usabilidade. ....	185
Quadro 18: Observações do entrevistado 1. ....	186
Quadro 19: Observações do entrevistado 2. ....	187
Quadro 20: Observações do entrevistado 3. ....	188
Quadro 21: Observações do entrevistado 4. ....	189
Quadro 22: Observações do entrevistado 5. ....	190
Quadro 23: Avaliação das entrevistas. ....	191



## LISTA DE TABELAS

Tabela 1: Tipos e grupos de relações semânticas para a análise do impacto de redes de ontologias sobre ferramentas de busca (copiado de BEDIN, 2007)....	52
Tabela 2: Visão geral das quatro tabelas de combinação de elementos visuais. .....	102
Tabela 3: Combinações entre quatro elementos geométricos básicos: (espaço, vértice, aresta, envoltória) X (espaço, vértice, aresta, envoltória). .....	103





## LISTA DE EQUAÇÕES

Equação 1: Vetor eixo principal da aresta. ....	119
Equação 2: Módulo do vetor eixo principal da aresta. ....	119
Equação 3: Vetor unitário do eixo principal da aresta. ....	120
Equação 4: O eixo vertical do espaço determinado pela aresta entre os dois vértices. ....	120
Equação 5: Determinação do ponto Q, no sistema de coordenadas da SuperAresta contêiner, dado o ponto P0(X0, Y0), no espaço do eixo determinado pela Aresta. ....	120
Equação 6: Lista de equações para cada ponto Q(X, Y) no espaço de um contêiner SuperAresta, dadas as coordenadas dos pontos da Aresta, no espaço determinado pelo eixo entre Vo e Vd. ....	121
Equação 7: Vetor do eixo formado entre Po e Pd. ....	123
Equação 8: Módulo do vetor D, contido na variável d. ....	124
Equação 9: Coordenadas dx, dy do vetor unitário uD do eixo formado entre os vértices de origem e destino. ....	124
Equação 10: Representação das linhas retas que formarão os segmentos da envoltória octogonal. ....	128



## **LISTA DE ABREVIATURAS E SIGLAS**

EC	- Engenheiro do Conhecimento
IA	- Inteligência Artificial
RRC	- Redes de Relacionamento e Conjuntos
API	- Application Program Interface
HTML	- Hyper Text Markup Language
INFOVIS	- Information Visualization
RDF	- Resource Description Framework
UML	- Unified Modeling Language
UNL	- Universal Networking Language
UW's	- Universal Words
VCL	- Visual Component Library
XML	- Extended Markup Language
WYSIWYG	- What You See Is What You Get



## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>2</b>
1.1 PROBLEMA DE PESQUISA .....	2
1.2 OBJETIVOS .....	3
<b>1.2.1 Objetivo Geral</b> .....	<b>3</b>
<b>1.2.2 Objetivos Específicos</b> .....	<b>4</b>
1.3 JUSTIFICATIVA.....	5
1.4 PROPOSTA DA TESE .....	7
1.5 CARACTERIZAÇÃO DA PESQUISA.....	7
1.6 METODOLOGIA .....	7
<b>2 MOTIVAÇÃO</b> .....	<b>9</b>
2.1 NECESSIDADES OBSERVADAS.....	10
2.2 Método de Busca: Procura e Retorno de Documentos nos Idiomas	11
2.3 MÉTODO DE CONSTRUÇÃO DE ONTOLOGIAS MULTILÍNGÜES .....	16
<b>2.3.1 Captura do <i>corpus</i> de documentos</b> .....	<b>16</b>
<b>2.3.2 Construção/seleção de termos</b> .....	<b>18</b>
<b>2.3.3 Construção de relações semânticas</b> .....	<b>20</b>
<b>2.3.4 Construção de palavras universais</b> .....	<b>22</b>
<b>2.3.5 Verificação de equivalência entre a ontologia em um idioma e a ontologia de palavras universais</b> .....	<b>24</b>
<b>3 REVISÃO DA LITERATURA</b> .....	<b>26</b>
3.1 Web Semântica.....	27
3.2 Modelos De Representação de Conhecimento .....	30
<b>3.2.1 Dicionários</b> .....	<b>30</b>
<b>3.2.1.1 IATE - Inter-Active Terminology for Europe</b> .....	<b>30</b>
3.2.1.2 <i>Universal Words Dictionary</i> .....	31
3.2.1.3 <i>Dicionários on-line e Wordnet</i> .....	31

<b>3.2.2 Ontologias .....</b>	<b>32</b>
<b>3.2.3 RDF - Resource Description Framework.....</b>	<b>35</b>
<b>3.2.4 OWL - Web Ontology Language.....</b>	<b>36</b>
<b>3.2.5 UNL como linguagem de suporte a ontologias .....</b>	<b>37</b>
3.2.5.1 <i>Elementos da UNL .....</i>	40
3.3 Metodologias de Representação do Conhecimento .....	42
<b>3.3.1 RC2D.....</b>	<b>42</b>
<b>3.3.2 Engenharia da Mente.....</b>	<b>43</b>
3.3.2.1 <i>Modelos de domínio da metodologia da Engenharia da Mente</i>	45
3.3.2.2 <i>Premissas .....</i>	47
3.3.2.3 <i>Validação de Ontologias construídas com a metodologia da Engenharia da Mente.....</i>	50
3.4 Ferramentas de Construção Colaborativa de Ontologias .....	53
<b>3.4.1 OntoKEM .....</b>	<b>53</b>
<b>3.4.2 Twine.....</b>	<b>54</b>
<b>3.4.3 Suíte de Engenharia do Conhecimento .....</b>	<b>55</b>
3.4.3.1 <i>MONGEE-Monitoramento On-Line de Gases de Efeito Estufa</i>	55
3.4.3.2 <i>Ontojuris .....</i>	56
3.4.3.3 <i>Ontoweb .....</i>	56
<b>3.4.4 Protègè .....</b>	<b>61</b>
3.4.4.1 <i>Método 101 .....</i>	63
<b>3.4.5 Growl.....</b>	<b>65</b>
<b>3.4.6 Comparação de Outras Ferramentas.....</b>	<b>66</b>
<b>3.4.7 Identificação das necessidades .....</b>	<b>70</b>
<b>3.4.8 Requisitos levantados para ferramentas de construção colaborativa de ontologias .....</b>	<b>73</b>
<b>3.4.9 Editores 3D e Ferramentas de Visão 3D de Ontologias .....</b>	<b>75</b>
3.5 Máquinas de busca.....	82
<b>3.5.1 Clusty .....</b>	<b>82</b>
<b>3.5.2 Cluuz .....</b>	<b>83</b>

3.6 Bibliotecas de Visão de informação .....	84
<b>3.6.1 Alphaworks .....</b>	<b>84</b>
<b>3.6.2 Framework RR .....</b>	<b>85</b>
<b>3.6.3 Prefuse .....</b>	<b>86</b>
<b>3.6.4 Java 3D .....</b>	<b>86</b>
3.7 Critérios de Usabilidade de Software .....	89
<b>4 MÉTODO DE CONSTRUÇÃO DO EDITOR DE ONTOLOGIAS MULTILINGUE .....</b>	<b>95</b>
4.1 Implementação do Protótipo de Visão de Redes de Ontologias.....	95
<b>4.1.1 Requisitos funcionais.....</b>	<b>95</b>
<b>4.1.2 Prototipagem para testes de viabilidade.....</b>	<b>96</b>
<b>4.1.3 Modelos de domínio para a ontologia multilíngüe .....</b>	<b>97</b>
<b>4.1.4 Elementos para representação de componentes das ontologias .....</b>	<b>100</b>
<b>4.1.5 Formas de representação.....</b>	<b>101</b>
<b>4.1.6 Combinação das formas de representação.....</b>	<b>102</b>
4.2 Descrição dos Modelos Estruturais e Matemáticos.....	106
<b>4.2.1 Descrição dos modelos matemáticos e algoritmos .....</b>	<b>109</b>
<i>4.2.1.1 A Borland VCL e a programação orientada a eventos .....</i>	<i>109</i>
<i>4.2.1.2 Eventos do mouse .....</i>	<i>109</i>
4.3 Movimento de Vértices e Conjuntos, com Uso do Mouse .....	110
4.4 Sobre a Classe Superaresta (O Contêiner de Instâncias da Classe Aresta).....	112
4.5 Diferenças entre os Sistemas de Coordenadas do Monitor e de uma Classe SuperAresta.....	114
4.6 Desenho de uma Linha Reta em um Contêiner de Arestas .....	115
4.7 Algoritmo Parcial de Desenho de uma Aresta Isolada .....	116

4.8 Eixo Determinado pelo Segmento de Reta Entre os Vértices Inicial e Final .....	119
4.9 Desenho de Uma Polilinha em um Contêiner de Arestas .....	121
4.10 O Desenho da Geometria da Extremidade de uma Aresta .....	122
4.11 Determinação da Envoltória.....	126
4.12 Princípio Básico de Construção: A Envoltória Quadrada.....	127
4.13 Envoltória Octogonal .....	128
<b>5 RESULTADOS DA APLICAÇÃO DO MÉTODO.....</b>	<b>129</b>
5.1 Primeira Versão do Editor de Ontologias 2D.....	129
5.2 Segunda Versão do Editor de Ontologia 2D .....	131
<b>5.2.1 Prototipagem das formas visuais de representação e suas combinações.....</b>	<b>131</b>
5.2.1.1 Conexão com fontes de dados Excel .....	131
5.2.1.2 Comunicação com bases Postgres, com dados criptografados	132
5.2.1.3 Elementos anexados .....	133
5.2.1.4 Envoltórias .....	134
5.2.1.5 Múltiplas arestas entre dois vértices.....	134
<b>5.2.2 O protótipo da segunda versão do editor de ontologias.....</b>	<b>135</b>
5.3 Terceira Versão do Editor de Ontologias: Visão 3D.....	140
<b>5.3.1 Prototipagem das formas visuais de representação e suas combinações.....</b>	<b>140</b>
5.3.1.1 Elementos básicos .....	140
5.3.1.2 Teste de visão de ambientes 3D com Java 3D em múltiplos documentos (MDI) .....	141
5.3.1.3 Testes de desempenho para vértices .....	144
5.3.1.4 Testes de desempenho para vértices e arestas .....	145
5.3.1.5 Testes de elementos de arestas e espessura de arestas .....	146
5.3.1.6 Testes de carga de imagens da base de dados para o espaço 3D .....	147
5.3.1.7 Rótulos de vértices e efeito de Billboard.....	149
5.3.1.8 Carga de termos de ontologias para o espaço 3D.....	150
5.3.1.9 Dependência do posicionamento das arestas com os seus vértices conectados .....	152



5.3.1.10	<i>Personalização do formato de arestas</i>	153
5.3.1.11	<i>Inclusão de setas</i>	154
5.3.1.12	<i>Múltiplas arestas entre dois vértices</i>	155
5.3.1.13	<i>Arestas conectando arestas entre si e arestas conectando arestas com vértices</i>	157
5.3.1.4	<i>Programação de envoltórias 3D</i>	158
5.3.1.15	<i>Envoltória rombicuboctaédrica sobre vértices</i>	162
5.3.1.16	<i>Combinando envoltórias e vértices para representar domínios e termos</i>	163
5.3.1.17	<i>Simplificações no algoritmo de cálculo da envoltória paralelepipedicas</i>	163
5.3.1.18	<i>Arestas entre envoltórias paralelepipedicas</i>	164
5.3.1.19	<i>Navegação</i>	164
5.3.1.20	<i>Menus e formulários</i>	166
5.3.1.21	<i>Criação de vértices (Termos ou Palavras Universais)</i>	167
5.3.1.22	<i>Criação de arestas (Relações Semânticas)</i>	168
5.3.1.23	<i>Carga de modelos 3D externos</i>	169
5.3.1.24	<i>Parametrização das extremidades das arestas</i>	170
5.3.1.25	<i>Parametrização das arestas</i>	171
5.3.1.27	<i>Modelo de domínio</i>	175
5.3.1.28	<i>Primeiros resultados com simbologia UML</i>	176
5.3.1.29	<i>Primeiro resultado com a carga de domínios, termos e relações, com representação UML</i>	177
5.3.1.30	<i>Tentativa de inclusão de relevos e mapas</i>	178
5.3.1.31	<i>Carga e movimentação de múltiplos objetos 3D por janela</i>	180
5.3.1.32	<i>Interface Web do Editor</i>	181
<b>6</b>	<b>PERCEPÇÃO DO EDITOR PELOS USUÁRIOS ENTREVISTADOS</b>	<b>182</b>
<b>7</b>	<b>DISCUSSÃO</b>	<b>192</b>
<b>8</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>235</b>
<b>9</b>	<b>SUGESTÕES PARA TRABALHOS FUTUROS</b>	<b>238</b>



# 1 INTRODUÇÃO

## 1.1 PROBLEMA DE PESQUISA

Nos últimos vinte anos, numa escala global, a ‘Representação do Conhecimento baseada em Ontologias’ tem sido uma solução para diversas áreas da ciência. Além disso, pela emergência da sua necessidade, está em vertiginoso desenvolvimento em virtude das constantes pesquisas empreendidas. Tais formas de representação têm permitido que as máquinas, quando requisitadas, executem processos de dedução de informações – a partir de um conhecimento básico mínimo, igual ao que faz o ser humano – desempenhando, com isso, tarefas racionais, como a de responder a perguntas específicas e a de oferecer sugestões para resolver um problema.

Um importante motivo para a construção de ontologias é a possibilidade da reutilização, por sistemas computacionais, das informações por elas representadas, sem a necessidade de reprogramar esses sistemas toda vez que uma nova área de conhecimento deva ser representada. Desta forma – graças às ontologias, na Web Semântica – qualquer conhecimento pode estar disponível para qualquer pessoa, em qualquer tempo e lugar, e qualquer processo automatizado pode coletar informações de locais virtuais e processá-las, sendo capaz de responder a perguntas específicas feitas por usuários.

Um segundo motivo relevante para a construção de ontologias é a sua capacidade de representar e comunicar o conhecimento em diferentes línguas, denominadas ‘ontologias multilíngües’, em que o conhecimento armazenado em uma instância de uma ontologia em um idioma pode ser reaproveitado por sistemas desenvolvidos em outro idioma.

Sendo as ontologias bases de conhecimento que servem como um ponto de partida para que os sistemas computacionais possam inferir sua informação, elas devem ser construídas por especialistas. Entretanto, essa construção depende de métodos e ferramentas que facilitem o trabalho.

Não obstante, uma grande limitação nos processos de construção de ontologias e nos processos nos quais se deseja observar os efeitos do uso de ontologias refere-se à oferta das suas formas de visão. Para facilitar a gestão de grandes volumes de conhecimentos, vários métodos e ferramentas foram desenvolvidos com a intenção de permitir aos especialistas ver de maneira clara a ontologia construída. Formas comuns de representação visual de ontologias são as tabelas, para representação seqüencial de dados; as listas indentadas, para representação de hierarquias; ou as redes de relacionamento, para descrever relacionamentos semânticos múltiplos entre conceitos. Mesmo assim, devido ao tamanho crescente de uma instância de ontologia em construção, é difícil ver e diagnosticar a utilidade ou impacto dos conceitos, regras, relacionamentos semânticos, e domínios, sobre uma futura inferência nesta ontologia.

Nesse contexto, a construção de ontologias segue uma tendência de trabalho que é o compartilhamento das tarefas entre vários profissionais. Isso se deve ao fato de que os dados referentes às ontologias serem produzidos em grande quantidade, além de possuírem alta complexidade, já que servem como insumo para inferência de informações em sistemas inteligentes. Além disso, como já mencionado, nas ontologias

produzidas estão representadas informações em vários idiomas. Em virtude do tamanho crescente das ontologias, são exigidos artifícios de visão e controle de seus dados.

Diante desta situação, para gerar soluções que satisfaçam a visualização dos dados crescentes em volume e complexidade, são necessárias ferramentas de visão de informação sobre ontologias multilíngües.

Atualmente, existe uma série de editores de ontologias que apresentam um conjunto de visão de dados sobre as instâncias construídas. Dependendo da informação que se deseja sobre a instância, é possível apresentar visões de listagem, de hierarquia, na forma de scripts e na forma de redes de relacionamento, todas elas presentes no espaço bidimensional.

Apesar da versatilidade da apresentação da informação no espaço 2D, a apresentação de informação nesse espaço contém duas limitações: a) A sobreposição de documentos que contenham as diversas visões da ontologia construída, em que caso o usuário deseje ver vários documentos de uma vez, ele deve escolher o documento que deve estar ativo para que possa ver suas informações e possa trabalhar sobre ele. b) A reduzida capacidade de mostrar relacionamentos de equivalência entre diferentes diagramas que apresentem visões de um mesmo elemento, em que caso um elemento de trabalho esteja sendo mostrado em dois diagramas diferentes, de maneira simultânea, e este elemento esteja sofrendo modificações, os dois diagramas são atualizados de maneira síncrona; mas, a priori, antes de iniciar a modificação do elemento, o usuário não percebe que dois diagramas possam conter este mesmo elemento.

Neste trabalho, um modelo para visualização de conceitos e construção de ontologias multilíngües foi elaborado para uma metodologia específica. O problema de pesquisa, então, é apresentado: *É possível a comunicação multilíngüe de conceitos utilizando objetos no espaço virtual tridimensional compartilhado?*

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

O objetivo geral desta tese é desenvolver um método de construção e representação visual de instancias de ontologias multilíngües e modelar um editor, usando recursos gráficos em espaços 2D e 3D.

### 1.2.2 Objetivos Específicos

- a)Apresentar modelos de representação de conhecimento;
- b)Apresentar metodologias de representação de conhecimento;
- c)Apresentar ferramentas de construção de ontologias, e máquinas de busca, suas evoluções/tendências;
- d)Apresentar bibliotecas de visão de informação, e suas evoluções/tendências;
- e)Descrever os critérios de usabilidade de software para executar a comparação entre os modelos tabular, bidimensional e tridimensional;
- f)Definir uma metodologia para o desenvolvimento de um modelo gráfico de visão de ontologias;
- g)Apresentar um modelo de domínio sobre o qual será desenvolvido este modelo gráfico;
- h)Criar modelos matemáticos para representar os formatos de visualização de informações, na forma de redes de relacionamento e conjuntos (RRC);
- i)Criar protótipos para testes de viabilidade de representação gráfica das ontologias em ambiente 2D e 3D;
- j)Comparar e avaliar os modelos tabular, bidimensional e tridimensional de visão de elementos de ontologias;
- k)Apresentar uma proposta de construção de ontologias multilíngües e um método de busca multilíngüe sobre as quais serão aplicadas às ontologias construídas.

### 1.3 JUSTIFICATIVA

Verifica-se, empiricamente, que o multilingüismo é uma tendência dos buscadores. Nessa perspectiva, a Universal Networking Language (UNL), há mais de 20 anos, persegue este objetivo: criar uma linguagem universal para o computador. Muitos trabalhos acadêmicos estão sendo desenvolvidos 'nesta linha. Um exemplo disso é o projeto Ontojuris (Bueno et. al. 2009): um consórcio formado por quatro países (Brasil, Argentina, Chile e Espanha) para a recuperação da informação multilíngüe de legislação, cujos argumentos foram empregados neste trabalho.

Além disso, o engenheiro do conhecimento (EC) enfrenta obstáculos para construir ontologias de modo síncrono e compartilhado, em virtude da dificuldade de ver e acompanhar a produção de outro EC. Este impedimento também é verificado em outros editores, tais como: o Protege, OntoKEM e UNL, como será demonstrado neste trabalho.

Diante dessas circunstâncias e como alternativa de solução para apresentação visual de instâncias de ontologias, propõe-se uma visão de todos os dados referentes às instâncias de ontologias trabalhadas em um espaço tridimensional. Como primeira vantagem dessa forma de representação, um editor 3D pode apresentar documentos no espaço, mostrar as conexões entre os elementos componentes desses documentos como também mostrar as conexões entre um mesmo elemento em suas diferentes visões.

Outra vantagem da apresentação 3D de dados de ontologias é a utilização da profundidade do espaço na apresentação de instâncias que são apresentadas em espaços bidimensionais. Desta forma, as instâncias de ontologias podem ter suas redes separadas em camadas, na qual o usuário pode afastar os conceitos que sejam menos interessantes. Ou ainda, referente à dimensão da profundidade do espaço 3D: o usuário pode utilizar esta terceira dimensão para demonstrar medidas diferentes na construção da ontologia, em que o eixo da profundidade pode servir como uma medida contínua de tempo de construção, número discreto de utilizações do conceito, métricas de votação sobre a utilidade ou coerência do conceito, ou mesmo essas medidas podem ser usadas para as relações semânticas construídas entre os conceitos.

Uma terceira vantagem da utilização do espaço 3D é a possibilidade de focar uma determinada parte da instância de ontologia, deslocando os demais elementos que não interessam para uma análise. Algo muito semelhante ao processo de afastar um grupo de informações para fora da região visível do

espaço bidimensional, com a diferença de que o usuário terá uma dimensão a mais para posicionar seus subgrupos de dados.

Considerando que as ontologias podem ser organizadas em grupos de conceitos ou domínios, a existência de envoltórias poliédricas pode mostrar tais domínios no espaço 3D. Tais envoltórias podem diferenciar grupos de conceitos e relações através de cores ou mesmo de formatos, em que cada cor e formato podem possuir um significado quanto ao grupo de conceitos que envolvem, tal como uma quantidade de conceitos que pertence a um domínio, ou a quantidade de usuários que esteja trabalhando neste domínio.

A quinta vantagem em se trabalhar com ontologias em um espaço 3D é a liberdade que o usuário tem de dispor os nós da rede de relacionamentos que representa uma instância de ontologia em várias formas geométricas: Disposição cilíndrica; Disposição Esférica; Disposição Semi-esférica (calota); Disposição Cúbica ou Paralelepípedica; Disposição Espiralada; Disposição Hierárquica em Árvore.

A sexta vantagem de se trabalhar com ontologias em um espaço tridimensional é a possibilidade que o usuário tem de utilizar navegação alternativa entre os elementos de uma ontologia. Existem vários equipamentos de navegação dentro de espaços virtuais no mercado (displays montados na cabeça (HMD's), luvas digitais, acelerômetros), os quais podem ser utilizados para dar liberdade adicional na construção de instâncias de ontologias.

A sétima vantagem é a possibilidade de associar de maneira direta conceitos ontológicos que definem o espaço, com os próprios fatos espaciais que ocorrem na tridimensionalidade: os conceitos e relações semânticas podem ser evidentemente e diretamente associados com lugares geográficos, ou com lugares geométricos, facilitando em muito a visão dos relacionamentos dos conceitos com os seus fatos espaciais descritos.

Por último, como justificativa para esta pesquisa, estão os 'Motivos de Adesão da Tese' ao Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento (PPGEGC- UFSC): a) A necessidade da criação de métodos para gestão de conhecimento que possam ser reusados em projetos internacionais (BUENO, 2008, STRADIOTTO, 2009). b) A tese propõe pesquisas sobre métodos de representação da informação; sobre um método de construção de ontologias em vários idiomas e de recuperação e visão de informação em vários idiomas; c) A tese propõe um modelo de representação do conhecimento.

Portanto, esta tese trás embutida a cultura repassada pelo Departamento da Engenharia e Gestão do Conhecimento.

## 1.4 PROPOSTA DA TESE

A proposta da tese consiste em uma metodologia de construção de ontologias composta de palavras universais, possibilitando a conversão entre instâncias de ontologias em vários idiomas.

Para a validação desta tese, foi programado um sistema gráfico compartilhado de construção de ontologias multilíngües, baseado no modelo de Bueno (2008), na forma de redes de relacionamento e envoltórias, utilizando os conceitos de Palavras Universais do projeto UNL como ponte para equivalência entre os conceitos dessas ontologias, e usando critérios de usabilidade durante a construção do editor.

## 1.5 CARACTERIZAÇÃO DA PESQUISA

Esta pesquisa utilizou a abordagem qualitativa para análise e interpretação dos dados. A análise qualitativa está voltada para os dados produzidos pelas interações interpessoais provenientes da co-participação das situações em que os informantes estejam envolvidos. Segundo Chizzotti (1998), o conhecimento não se reduz a um rol de dados isolados que conectados por uma teoria explicativa, pois o sujeito-observador é parte integrante do processo de conhecimento e interpreta os fenômenos atribuindo-lhes significado. O objeto não é um dado inerte e neutro, mas está possuído de significados e relações que os sujeitos concretos criam em suas ações. Nas metodologias qualitativas, os sujeitos de estudo não são reduzidos a variáveis isoladas ou a hipóteses, mas vistos como parte de um todo, em seu contexto natural, habitual.

Quanto às fontes de informação, esta pesquisa obteve parte de suas informações por intermédio da pesquisa bibliográfica, em que os dados e as informações foram obtidos de fontes secundárias: livros, publicações e periódicos, ou seja, fontes que já apresentam algum tipo de organização e tratamento. “A pesquisa bibliográfica procura explicar um problema a partir de referências teóricas já publicadas (em livros, revistas, etc.)” (GIL, 2007).

Quanto aos objetivos, esta pesquisa caracterizou-se como descritiva, já que descreveu as características de determinado fenômeno estudado utilizando-se de técnicas de coleta de dados.

## 1.6 METODOLOGIA

A metodologia utilizada nesta tese correspondeu às seguintes etapas:



1. Revisão da literatura sobre modelos e métodos de representação do conhecimento, abordando a Web Semântica, dicionários, ontologias, RDF, OWL, UNL e seus elementos, e a RC2D.
2. Revisão da literatura sobre métodos de construção e validação de ontologias.
3. Revisão da literatura sobre ferramentas de construção de ontologias
4. Revisão da literatura sobre ferramentas e bibliotecas de visão de informação
5. Revisão da literatura sobre critérios de usabilidade de software revisão da literatura sobre editores 3D de ontologias e ferramentas de visualização de ontologias
6. Posteriormente foi descrito o método utilizado para a construção das várias versões do editor de ontologias multilíngües, desde a implementação de protótipos, com seus requisitos de projeto, e com seus modelos matemáticos descritos por equações, modelos estruturais e de comunicação com o usuário, e os resultados das aplicações destes métodos: as duas versões do editor de ontologias multilíngües em espaço 2D, e a versão final do editor de ontologias multilíngües em espaço 3D, com interação com objetos tridimensionais, carga de base de dados, testes de performance, personalização de elementos, navegação e manuseio, e representação de elementos com simbologia UML.
7. Após a prototipagem dos três aplicativos de construção de ontologias, foi feito um teste de comparação de usabilidade entre três ferramentas de construção de ontologias, baseando-se em critérios de usabilidade consultados na bibliografia. O teste de comparação foi feito através de entrevistas com usuários durante o uso das três ferramentas.
8. Logo após foi feita uma comparação destas ferramentas, e seu impacto no método de construção de ontologias, baseada nestes critérios de usabilidade, tendo seus aspectos comentados e comparados com depoimentos de outros pesquisadores do ramo
9. Finalmente foram dadas as conclusões a respeito da comparação feita, do método de entrevista, e foram feitas sugestões para pesquisas em trabalhos futuros.

## 2 MOTIVAÇÃO

Neste capítulo serão mostrados os métodos de busca de documentos e construção de ontologias multilíngües, anteriores ao método proposto nesta tese, que serviram como motivadores para a pesquisa. Aqui, esta tese descreve oito métodos diferentes:

1. O método de busca de documentos multilíngües do Ontosinapse (BUENO, 2005, 2008), que é o objetivo final da construção das ontologias multilíngües
2. Três métodos de construção destas ontologias multilíngües:
  - a. O método de construção destas ontologias multilíngües, descrito também no projeto Ontosinapse
  - b. O método descrito nesta tese, de construção de ontologias multilíngües, na forma de redes de relacionamentos e envoltórias, em espaço 2D
  - c. O método proposto nesta tese, de construção de ontologias multilíngües, na forma de redes de relacionamentos e envoltórias, com associação a objetos, em espaço 3D
3. Três métodos de construção de ferramentas para edição de ontologias:
  - a. A primeira versão do editor 2D de ontologias
  - b. A segunda versão do editor 2D de ontologias
  - c. O editor 3D de ontologias
4. O método de comparação de usabilidade entre três ferramentas de edição de ontologias: Ontosinapse, Segunda versão do editor 2D, e o Editor 3D de ontologias.

Aqui, se deve tomar o cuidado de não confundir um método para CONSTRUÇÃO DE ONTOLOGIAS, com um método para CONSTRUIR UMA FERRAMENTA para construção de ontologias. Por último, vale lembrar, resumidamente, que todo o processo de pesquisa partiu de uma necessidade de otimização de método de busca: Do método de busca, veio a necessidade da ferramenta de busca. Da ferramenta de busca veio a necessidade da construção da ontologia, pois originalmente ela não existe. Para a construção da ontologia foi necessário um método para esta construção. Para automatizar e compartilhar este método de construção de ontologia veio a necessidade da ferramenta de construção de ontologia, e finalmente, para concretizar a ferramenta de construção de ontologia, veio a necessidade do método de construção desta ferramenta para construção de ontologias.

## 2.1 NECESSIDADES OBSERVADAS

Observadas as necessidades de busca de informação multilíngüe de algumas páginas de procura (Google, Bing e Yahoo), e a funcionalidade da página de procura de termos em vários idiomas (IATE, 2008), criou-se o sistema Ontosinapse (BUENO, 2005, 2008) que disponibiliza ao usuário a seguinte funcionalidade: supondo que o usuário entre com um texto **T<sub>0</sub>**, em linguagem natural de seu idioma **I<sub>0</sub>** (e não apenas palavras isoladas) e que ele escolha os idiomas **I<sub>j</sub>**, cujo sistema deve retornar os documentos similares a essa entrada, este sistema de busca deve retornar para esse usuário uma lista de documentos similares ao documento de entrada, nos idiomas escolhidos, ordenados pela similaridade e agrupados pelos idiomas escolhidos.

O usuário entra com um texto em linguagem natural, no seu idioma, e escolhe os idiomas dos documentos resultantes da busca. O sistema retorna os documentos por ordem de similaridade, semelhantes ao documento (texto de entrada em linguagem natural), agrupados pelos idiomas escolhidos (Figura 1):

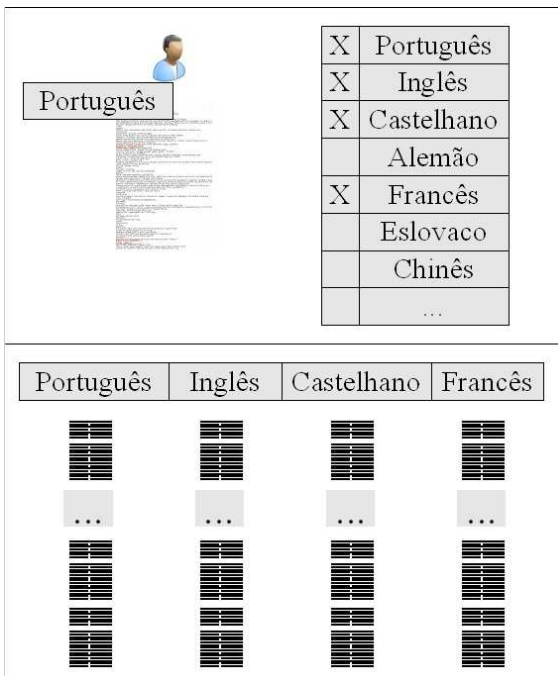


Figura 1: Diagrama mostrando a interface de entrada de um documento em linguagem natural, escolha de idiomas de retorno (superior) e listagem de documentos similares agrupados pelos idiomas escolhidos (inferior).

## 2.2 MÉTODO DE BUSCA: PROCURA E RETORNO DE DOCUMENTOS NOS IDIOMAS

O texto de entrada em português é indexado por uma ontologia = T Português, que tem seus termos associados a uma ontologia multilíngüe. As demais ontologias, em seus respectivos idiomas, que indexam seus respectivos documentos, também possuem seus termos associados a esta ontologia multilíngüe (Figura 2).

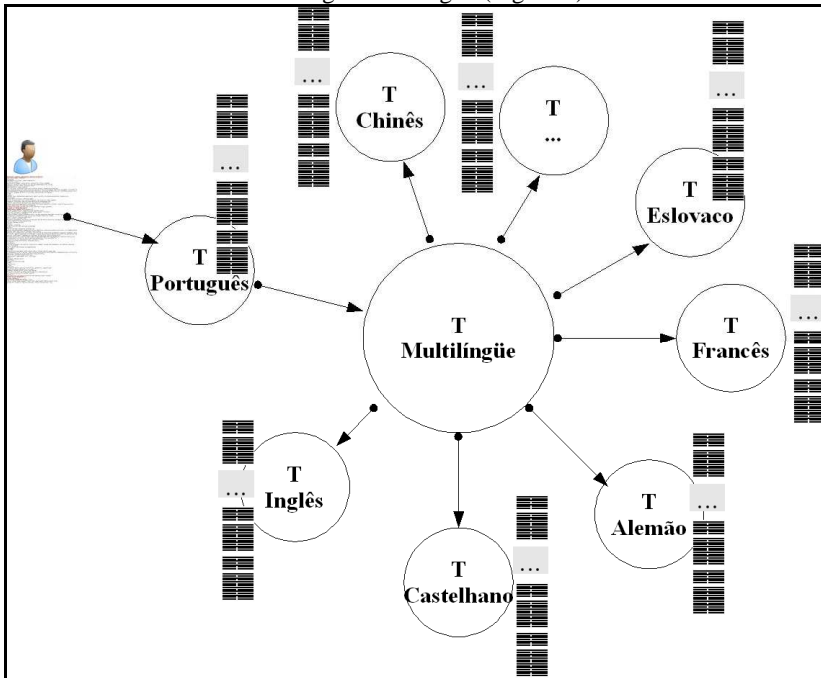


Figura 2: Diagrama explicando uma visão simplificada do método de procura de documentos multilíngües.

Detalhando a visão do sistema, o texto entrado pelo usuário e os textos da base do idioma português são anotados por uma ontologia de termos, relações e domínios do idioma português. Os documentos do idioma inglês são indexados pela ontologia no idioma inglês. Idem para os demais idiomas (Figura 3).

Os termos de cada ontologia têm uma equivalência com as palavras universais do projeto UNL (representadas na **Erro! Fonte de referência não encontrada.**, no lugar do transformador T Multilíngüe, no centro da **Erro! Fonte de referência não encontrada.**). Dessa forma, os termos de um idioma podem ser convertidos em termos de outro idioma.

Uma visão da interface gráfica do sistema, contendo as ontologias, com termos e relações, combinada com a visão dos documentos a serem indexados por essas ontologias é mostrada na **Erro! Fonte de referência não encontrada.**

O texto de entrada do usuário, escrito em linguagem natural, também é indexado pela ontologia construída no seu idioma de entrada (supondo que o usuário escreva textos em linguagem natural no idioma português) (

Figura 5: Anotação do texto de entrada do usuário (em Português), pela ontologia feita no seu idioma, e procura de termos equivalentes (em verde), nas ontologias dos outros dois idiomas (Ex: Espanhol e Inglês), com auxílio do dicionário de UW's da UNL.

).



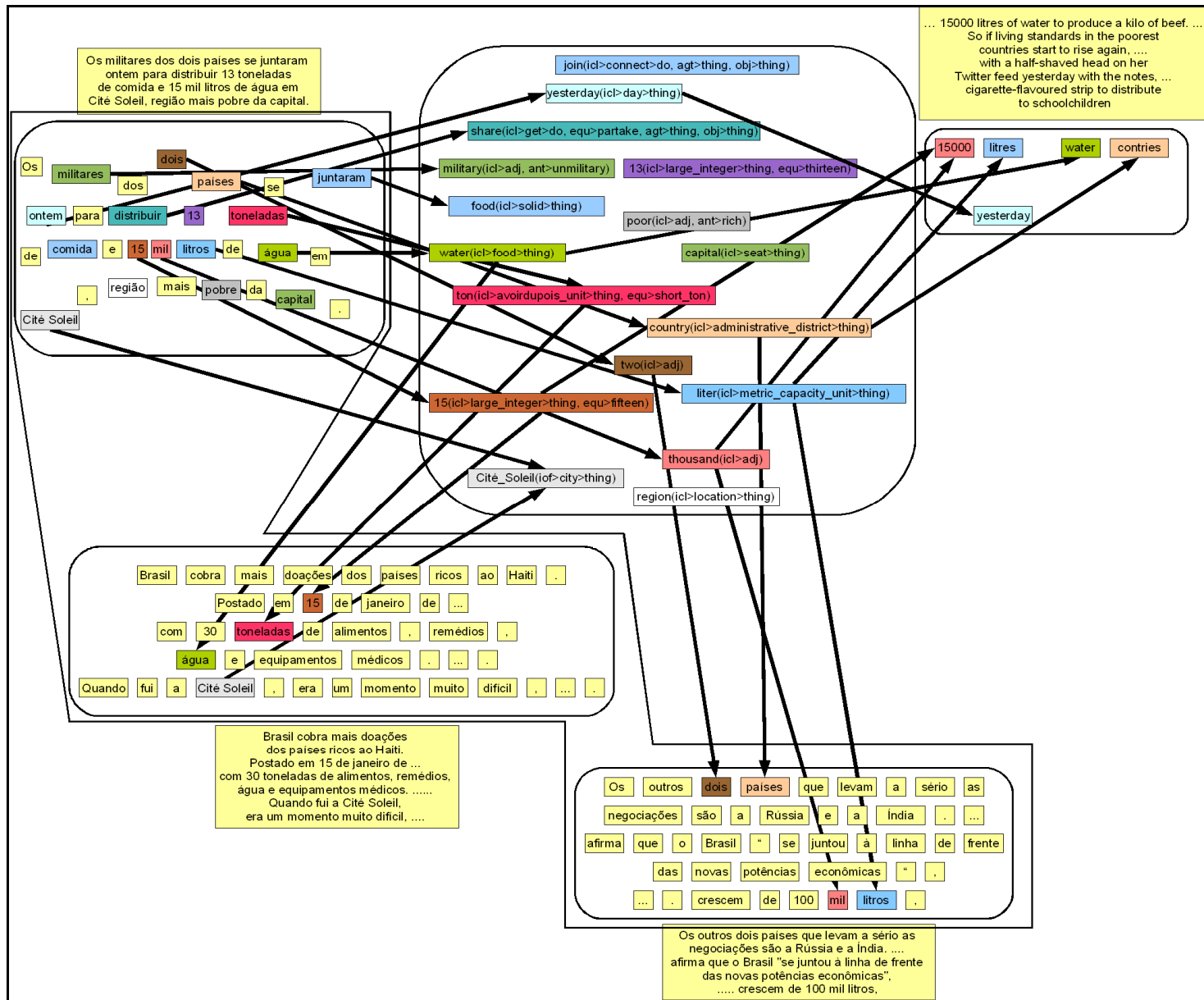


Figura 3:Diagrama detalhando o método de procura de documentos multilíngües.

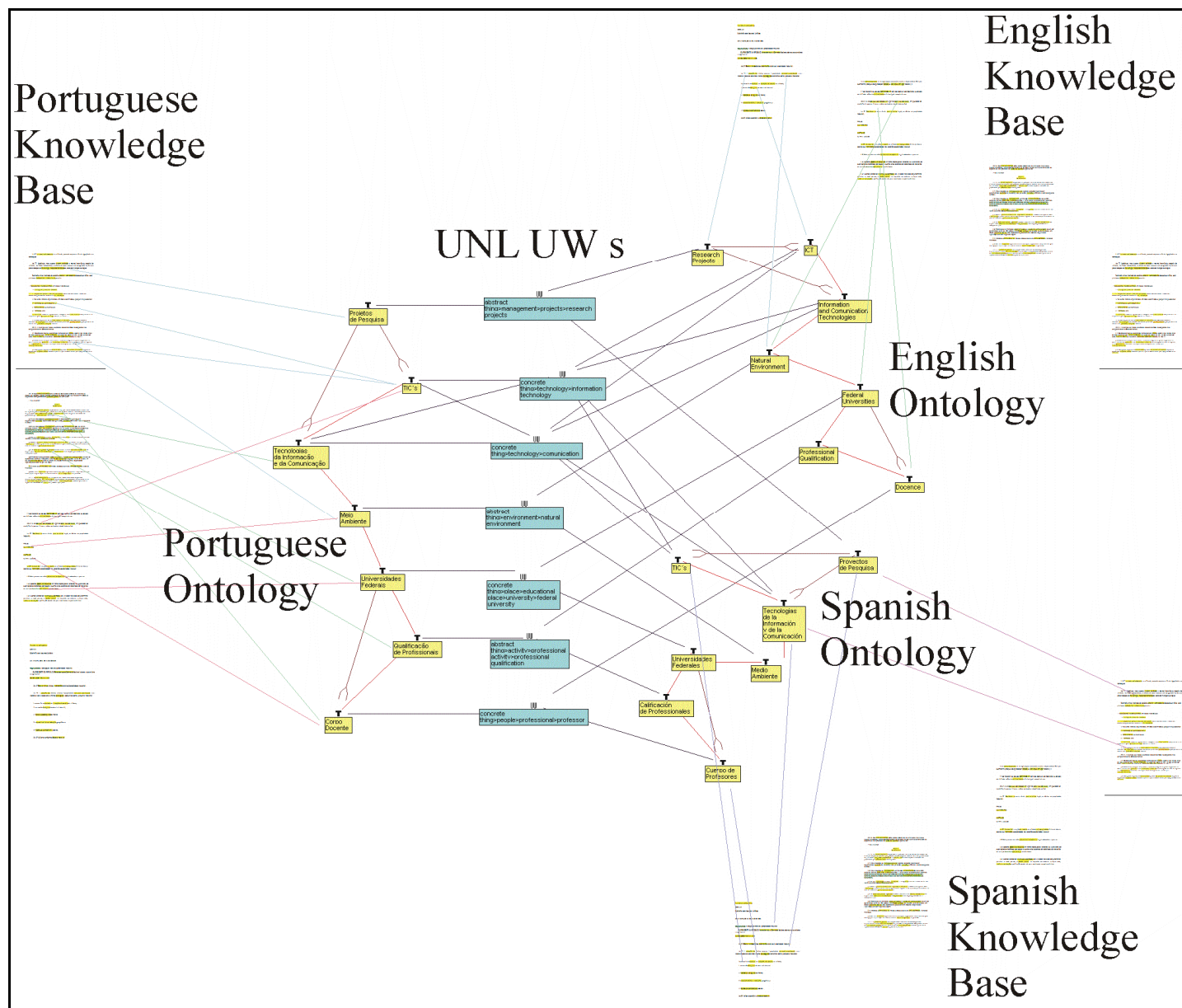


Figura 4: *Corpi* contendo os documentos em vários idiomas, anotados pelas ontologias do modelo de BUENO (2008), com o dicionário de palavras universais ao centro.



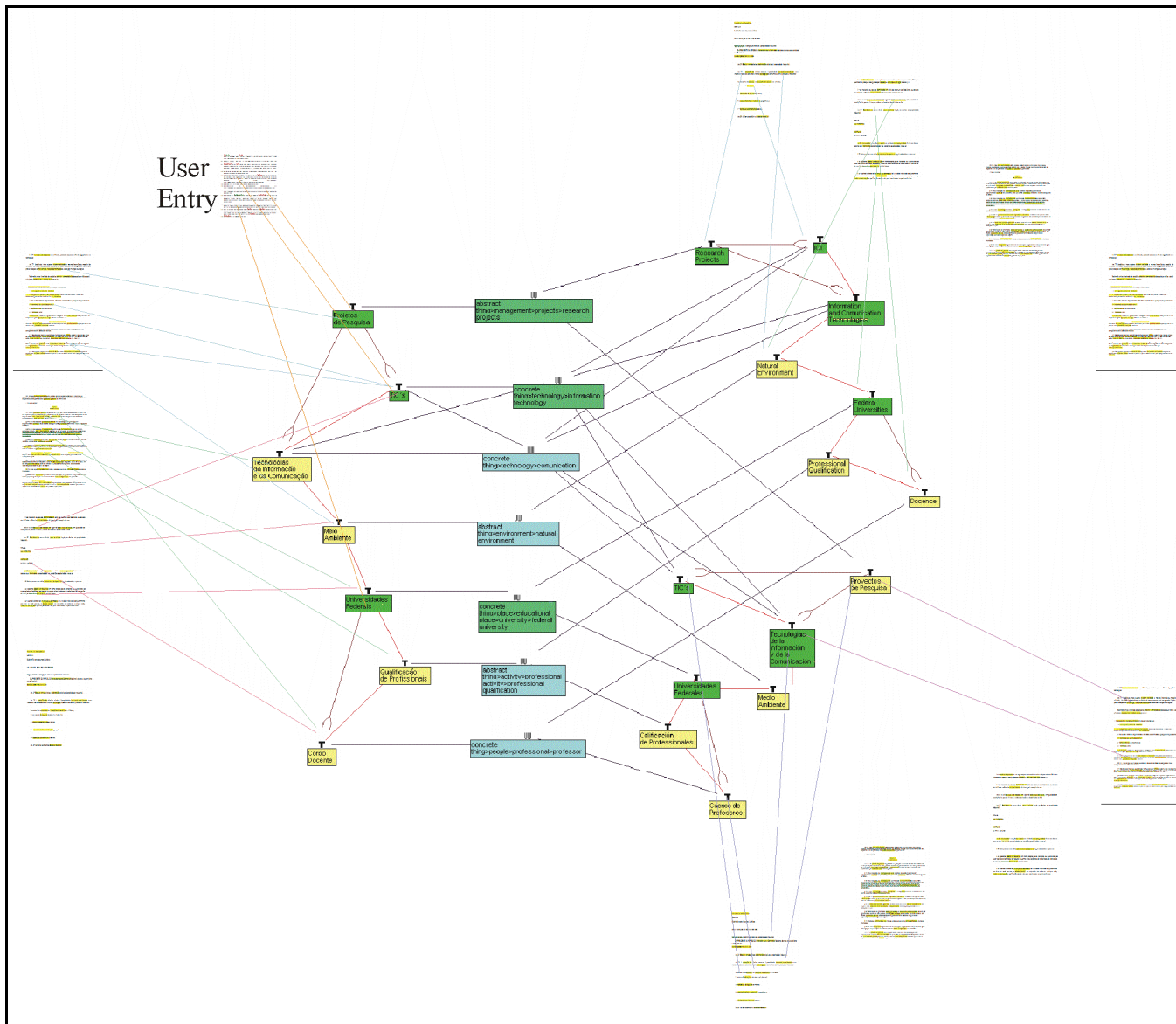


Figura 5: Anotação do texto de entrada do usuário (em Português), pela ontologia feita no seu idioma, e procura de termos equivalentes (em verde), nas ontologias dos outros dois idiomas (Ex: Espanhol e Inglês), com auxílio do dicionário de UW's da UNL.

## 2.3 MÉTODO DE CONSTRUÇÃO DE ONTOLOGIAS MULTILÍNGÜES

Este método de construção das ontologias partiu do modelo proposto por Bueno (2008), composto por cinco etapas:

1. Captura do *corpus* de documentos;
2. Construção/seleção de termos;
3. Construção de relações semânticas;
4. Construção de palavras universais;
5. Verificação de equivalência entre a ontologia composta de palavras universais e a ontologia construída em um idioma específico.

Também foram realizadas entrevistas com engenheiros de ontologias e especialistas de domínio e pesquisa bibliográfica.

### 2.3.1 Captura do *corpus* de documentos

Para garantir representatividade e foco sobre o domínio, vocabulário e das relações semânticas entre os termos do vocabulário, usados pelos especialistas do domínio, primeiro questiona-se o usuário sobre seus termos de interesse. Depois é feito um levantamento dos documentos que são de seu uso frequente: manuais, Web, editais, revistas e jornais (**Erro! Fonte de referência não encontrada.**).

Para a construção dos domínios (os assuntos principais que servirão como referência para a classificação dos termos), parte-se do princípio de que os documentos de uma ‘instituição cliente’ já estejam organizados de acordo com a hierarquia desejada pelos clientes usuários deste *corpus*.

Após esse levantamento inicial, vem a fase de enriquecimento do *corpus*, sobre a qual se pretende levantar documentos adicionais àqueles já utilizados pela ‘instituição cliente’. Alguns termos do *corpus* inicial são pré-elegidos e procurados em ferramentas de busca. Os documentos retornados por essas ferramentas têm sua credibilidade avaliada pelos engenheiros do conhecimento. Quando o número de documentos selecionados pelos engenheiros do conhecimento atingir uma quantidade suficiente, parte-se para etapa seguinte de construção/seleção de termos.

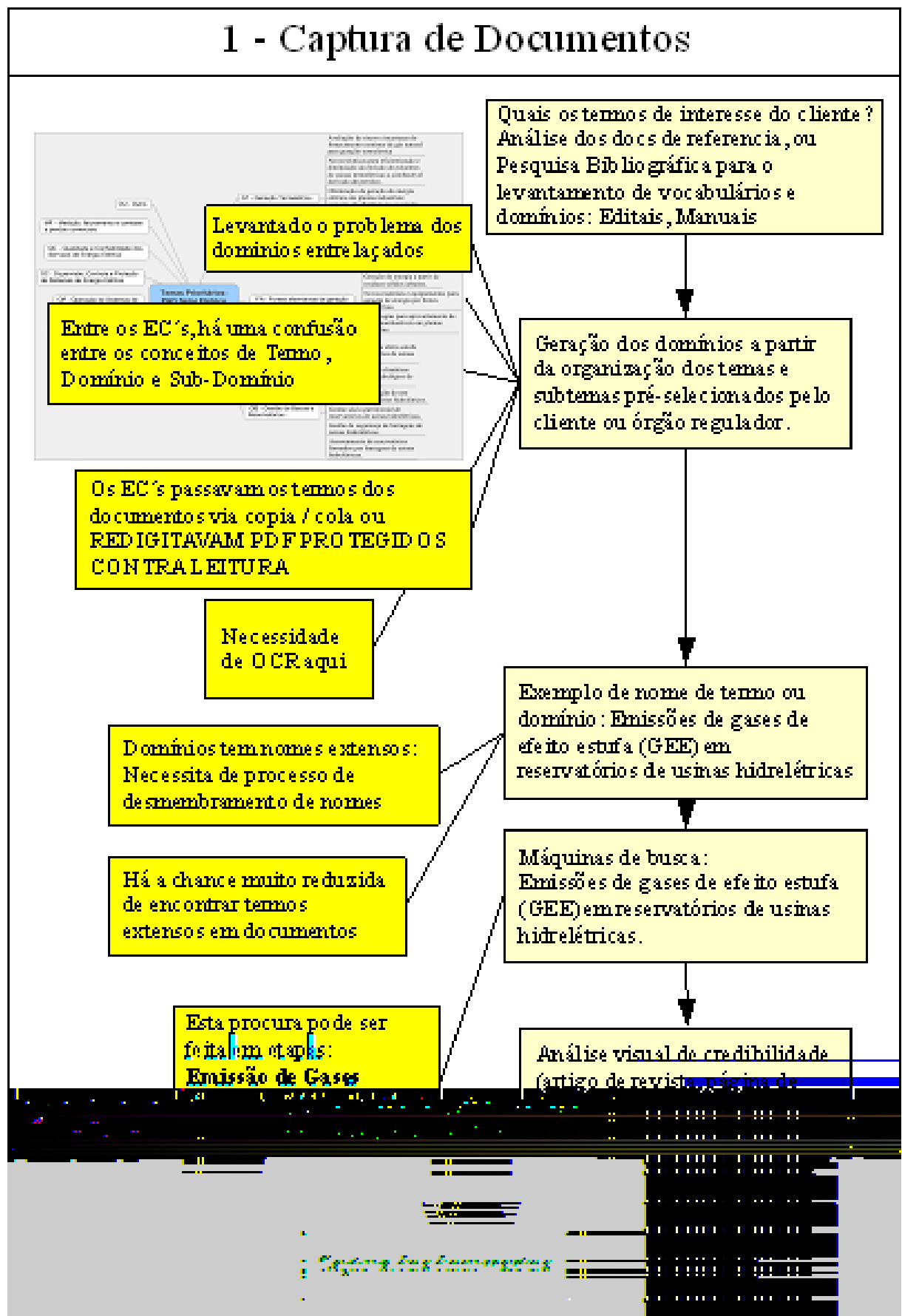


Figura 6: Etapa de captura de documentos.

### 2.3.2 Construção/seleção de termos

Uma vez que os documentos do *corpus* foram eleitos, estes são guardados em uma base de conhecimento; sobre eles é feito um levantamento de termos relevantes de duas formas:

1. Levantamento de termos com a ajuda de analisadores semânticos: situação em que é contada a frequência de termos nos documentos ou é contada a frequência dos radicais desses termos, procurando por alguma forma de generalização.

2. Levantamento de termos com a leitura dos documentos por engenheiros do conhecimento e por especialistas do domínio (**Erro! Fonte de referência não encontrada.**): se um determinado termo ou seus radicais aparecer em mais de um documento em uma quantidade pré-determinada, tal observação é um indicativo para incluir esse termo na ontologia. Essa leitura deve ser feita por profissionais que sejam especialistas do assunto desejado pelo cliente, pelos próprios clientes e por engenheiros de ontologias orientados por estes especialistas.

O processo termina quando se esgota a ocorrência de termos repetidos nos documentos, que sejam julgados como pertencentes ao domínio estudado.

## 2 – Construção / Seleção de Termos

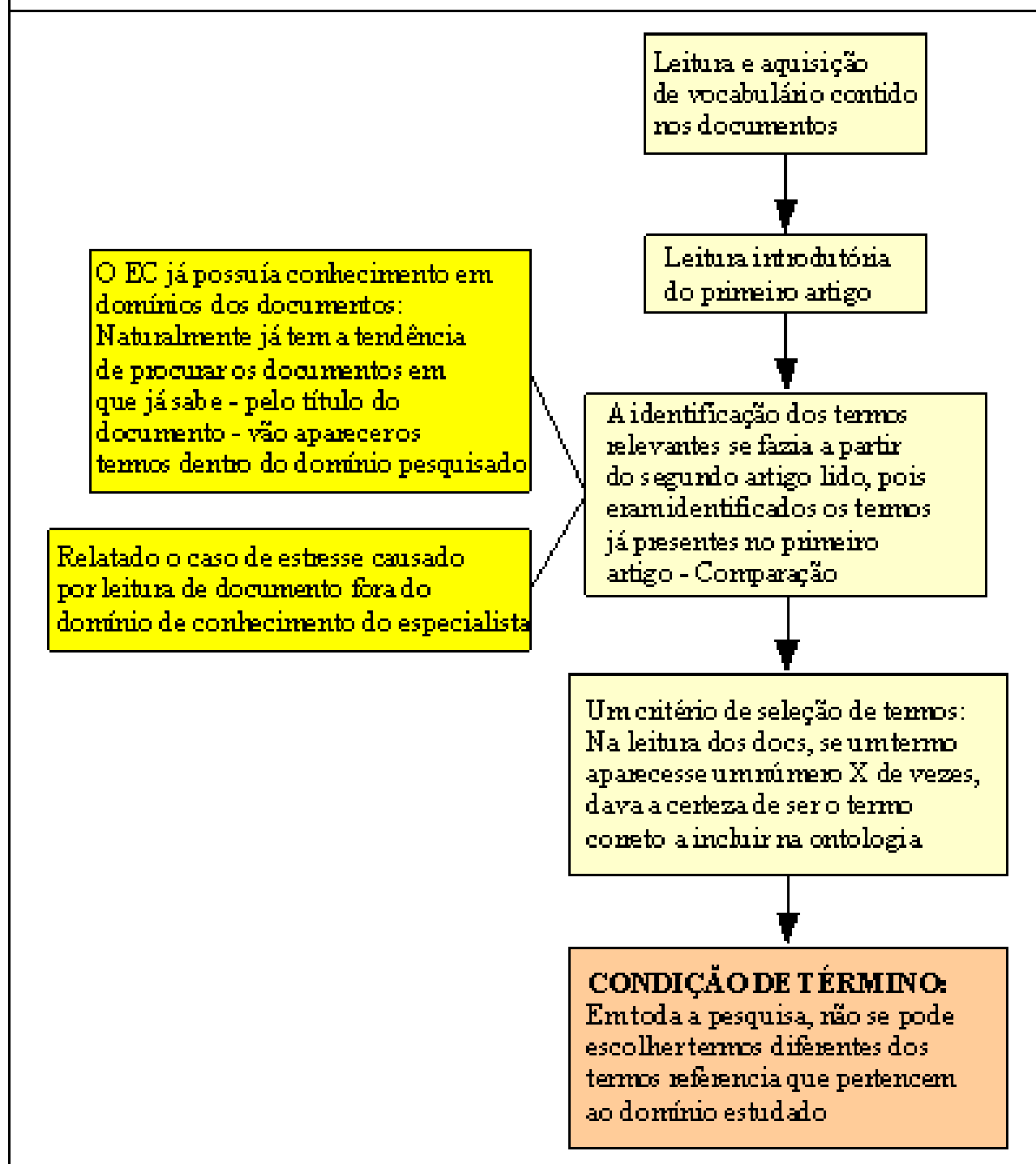


Figura 7: Diagrama representando a etapa de seleção / construção de termos.

### **2.3.3 Construção de relações semânticas**

A construção de relações semânticas (Figura 8) começa pelo caso mais fácil: a construção de relações de sinonímia entre termos. A procura de sinônimos é facilitada pelo uso de dicionários impressos ou *on-line*. Devido a uma limitação inicial da representatividade das relações semânticas, as relações de causa/consequência entre termos são também representadas por uma relação de sinonímia.

As relações de ‘tipo de’ e ‘parte de’ não são consultadas em dicionários, mas em documentos científicos ou jornalísticos sobre os domínios que estão sendo pesquisados.

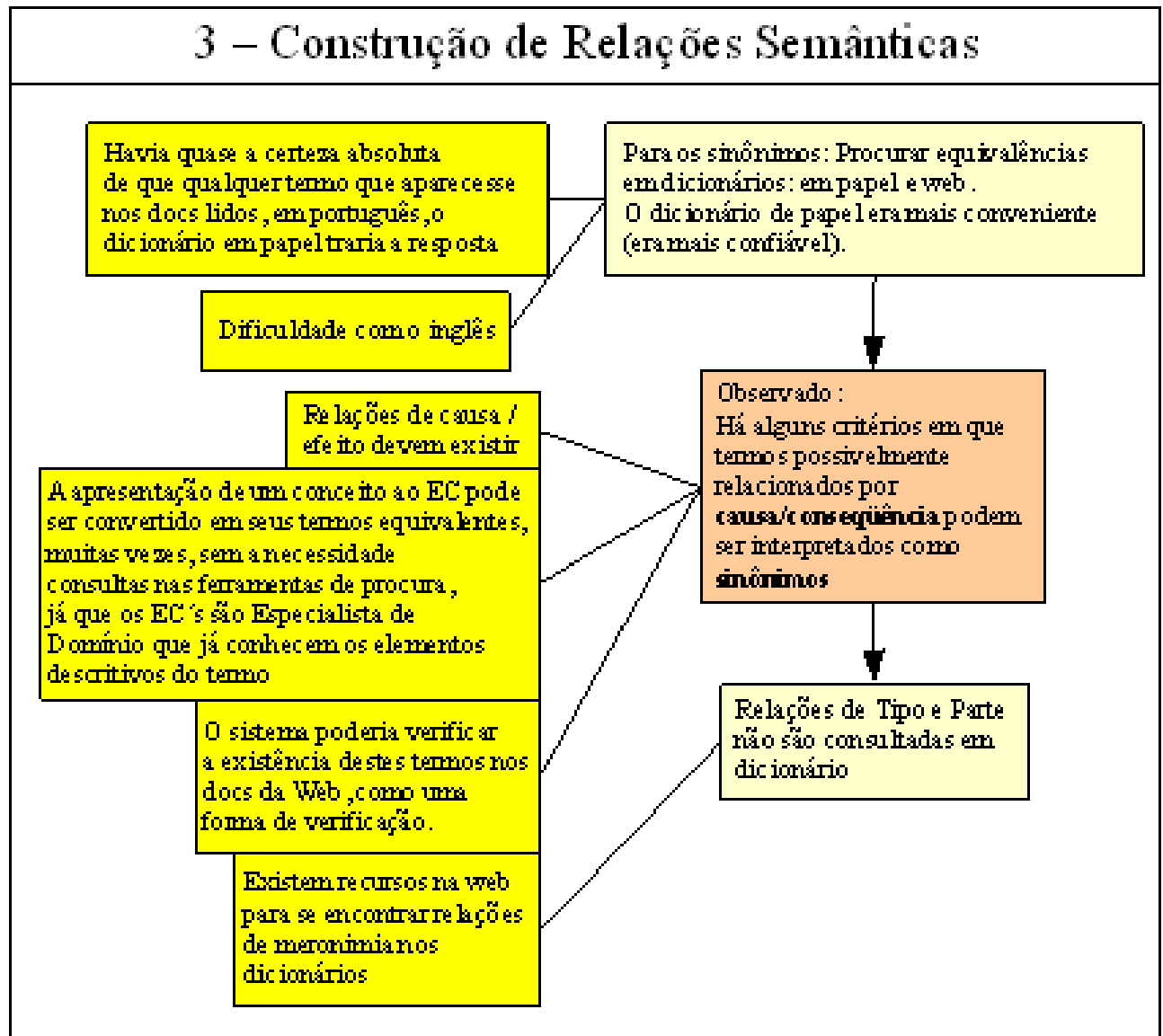


Figura 8: Diagrama representando a etapa de construção de relações semânticas.

### **2.3.4 Construção de palavras universais**

Feita a associação semântica dos termos, os engenheiros do conhecimento partem para a construção das palavras universais (Figura 9). A tradução das palavras dos termos para o inglês ou são traduzidas como expressões idiomáticas ou termos técnicos dentro do domínio estudado.

Os termos construídos pelos engenheiros de ontologias são traduzidos para o inglês. A palavra universal equivalente é procurada no UWDic (UNIVERSAL WORDS DICTIONARY, 2010). Caso a palavra universal equivalente não seja encontrada, a redação de uma nova palavra é proposta, passando por revisões, até ser armazenada na base de conhecimento. Caso o termo seja composto por de duas palavras e não exista uma palavra universal equivalente no Dicionário Web de Palavras Universais, sugere-se a construção de uma palavra universal composta, incluindo relações semânticas da UNL.

Considerando que os termos são compostos de no máximo três palavras – e mais os conectores sintáticos (de, para, por são os mais comuns), que são abstraídos das palavras universais devido às regras sintáticas no processo de conversão – o número de palavras universais simples que fazem parte da palavra universal composta resultante é relativamente pequeno e simples para a gestão pelo engenheiro de ontologias e para um sistema computacional.



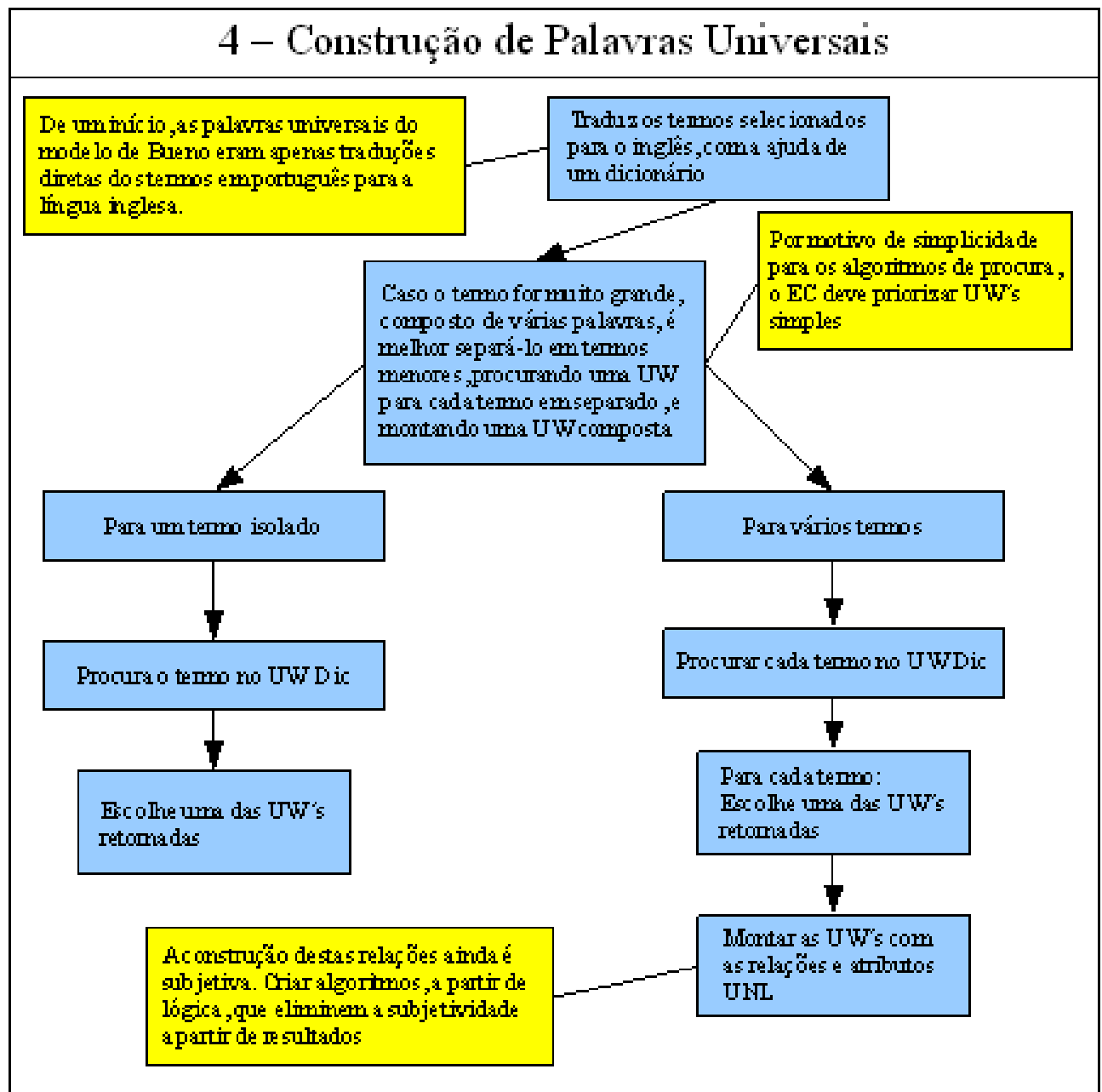


Figura 9: Diagrama representando a etapa de construção de palavras universais.

### 2.3.5 Verificação de equivalência entre a ontologia em um idioma e a ontologia de palavras universais

A etapa de verificação exige que o cliente ou o engenheiro de ontologias compreenda os idiomas sob os quais a busca será feita. Esta etapa de verificação de equivalência parte do princípio de que cada termo da ontologia em um dado idioma, e cada domínio desta mesma ontologia, tem a sua palavra universal equivalente na ontologia normalizada de palavras universais, como se fosse um espelho entre as redes (ver item 2.2).

Após a construção da equivalência entre as ontologias em **N** idiomas, uma busca de teste é feita (**Erro! Fonte de referência não encontrada.**). Um documento escrito em um idioma de entrada é fornecido ao sistema; este responde com apenas dois documentos em cada um dos idiomas alvo. O teste retorna somente dois documentos para que seja possível aos dois profissionais comparar a similaridade desses documentos com o documento de entrada, sem exigir demasiada atenção com vários documentos a serem comparados.

Os documentos são retornados com os termos contidos nas ontologias em formato destacado (artifício visual chamado de *syntax-highlight*). O especialista do domínio e o engenheiro de ontologias verificam a ordem dos documentos baseada na similaridade destes com o documento de entrada. Se os documentos – no critério dos dois profissionais – estiverem com a ordem invertida ou ainda que um dos documentos retornados possuam termos em *syntax-highlight* não relacionados com os termos do documento de entrada, significa que há uma incoerência no relacionamento de equivalência entre a ontologia de palavras universais e as ontologias dos dois ou mais idiomas envolvidos. Caso haja discrepância, as conexões de equivalência entre as ontologias nos idiomas e a ontologia normalizada pelas palavras universais é revista e corrigida.

## 5 – Verificação de Equivalência entre a Ontologia em um Idioma e a Ontologia de Palavras Universais

Dadas as UW's  $UW_i$  e  $UW_j$ , equivalentes aos termos  $T_i$  e  $T_j$ , respectivamente, e dada a relação  $R$  entre  $T_i$  e  $T_j$ ,  $UW_i$  e  $UW_j$  terão a mesma relação entre si

O nome de um domínio também é substituído por uma UW simples ou composta

Construída a ontologia com as UW's o usuário deve fazer os testes de procura multilíngüe.

Figura 10: Etapa de verificação de equivalência entre a ontologia de UW's e a ontologia em um dado idioma.

### 3 REVISÃO DA LITERATURA

Neste capítulo são descritos os modelos, métodos, metodologias, ferramentas e bibliotecas empregados para o desenvolvimento deste estudo. Também serão rapidamente descritos dois conceitos que não se encaixam nos cinco anteriores: A Web Semântica, e os Critérios de Usabilidade de Software.

Por **MODELO**, entende-se como uma abstração de um elemento real, capaz de ser replicável, comunicável, representativo da realidade que se deseja mostrar. Nesta tese, usam-se algumas representações de modelo de domínio, para representar e comunicar a estrutura em que são guardadas as ontologias trabalhadas, modelos matemáticos para a representação dos elementos gráficos nos protótipos, e modelos estruturais para representar a comunicação destes elementos. Aqui, entende-se também, por modelo, toda a forma de representar uma informação de equivalência (tal como os dicionários), ou de inferência de conhecimento (para o caso das ontologias, RDF, OWL e UNL).

Por **MÉTODO**, se entende o meio do caminho entre uma situação inicial, e uma situação final, desejada. Para o caso desta tese, a situação inicial é a não existência de uma instancia de ontologia. A situação final é a existência desta instancia. Para que esta instancia exista, é necessário um material inicial, e um conjunto de procedimentos, para que se possa construir – gradativamente – esta instancia. Como um primeiro exemplo de método para a construção desta instancia de ontologia, poderia ser citado o uso de pequenos blocos de papel, onde estariam os conceitos escritos, ligados por linhas – representando os relacionamentos semânticos entre estes, ou colados dentro de um determinado espaço físico, que represente seu domínio de conhecimento. Ao final da conexão dos vários blocos, com as linhas, estaria consolidada uma rede física de conceitos, relacionamentos, e domínios, que representassem a instancia desejada. Nesta revisão, são descritos somente dois métodos: O método 101, de construção de ontologias sobre a ferramenta Protege, e o método de validação de BEDIN (2007), das ontologias construídas com a metodologia de BUENO (2005) .

Por **METODOLOGIA**, se entende o levantamento, a descrição, a separação em pedaços menores, a comparação, a aplicação e observação de resultados, e finalmente, a síntese de métodos. Nesta revisão, são levantadas duas metodologias: A de construção de ontologias de

BUENO (2005), e a metodologia de representação de conhecimento descrita por HOESCHL (2002).

Por FERRAMENTA, se entende a concretização e disponibilização de artifícios que permitem e facilitam a aplicação de um método. Aqui foram revisadas somente ferramentas de informática. Tais ferramentas se caracterizam por apresentarem seus dados de forma bem definida, e de possuírem um conjunto de funcionalidades aplicáveis a estes dados. As ferramentas consideradas nesta revisão foram duas: a) As ferramentas para construção de ontologias, que trabalham com a apresentação e edição de dados de ontologias e sua aplicação a processos de inferência, e b) As máquinas de busca, que se caracterizam pelo retorno de documentos, dado um documento de entrada, e/ou parâmetros de pesquisa. Aqui, as máquinas de busca se destacam como elementos motivadores para processos de construção de ontologias, dado que, em uma busca requisitada por um usuário, alguns documentos podem ser retornados baseando-se em processos de inferência. Outro elemento motivador para a inclusão de máquinas de busca é o fato de que algumas delas já apresentam a tendência de apresentar os documentos e informações retornadas em vários formatos visuais, caracterizando a sua intersecção com ferramentas de visão de informações.

Por BIBLIOTECA, se entende um conjunto de funções e classes programáveis, dentro de uma determinada linguagem, que permitem, desta forma, a construção de ferramentas. As bibliotecas não são ferramentas prontas, mas permitem a criação destas. De forma específica, uma BIBLIOTECA DE VISÃO DE INFORMAÇÕES se caracteriza por possuir funções matemáticas e de manipulação de pixels tais, que permitem a construção de vários tipos de aplicações especializadas na apresentação de dados.

### 3.1 WEB SEMÂNTICA

Berners-Lee (1998) descreve a Web Semântica como uma rede com informações expressadas de forma a serem compreensíveis por robôs. Essa teia é composta por um modelo descritivo de grande simplicidade e generalidade: a *Resource Description Framework* (RDF) que é composta por dois elementos básicos: a) Assertivas (*Assertions*: Afirmções sobre uma verdade em forma de predicado); b) Quotações (*Quotations*: Assertivas sobre Assertivas, para descrever metadados, na forma lógica mais simples). Devido às aplicações específicas da RDF, as

assertivas são partes da descrição de algum recurso, e este recurso é freqüentemente um parâmetro implícito. Isto significa que a assertiva é uma propriedade do recurso.

Sobre isso há grande possibilidade de os problemas serem tratáveis, pelo fato de o modelo RDF – devido à sua extrema simplicidade – poder ajudar a determinar a existência ou não de uma prova, dada uma série de fatos e uma dada questão. Sobre o uso do modelo, por enquanto, serve somente para representar dados, sem regras de inferência ou *queries*. Alguns exemplos de aplicações que usam o modelo RDF para compreender os dados contidos nas páginas são: informação para indexação de cartas (pelas ontologias Dublin Core), informações de privacidade (P3P), associações de folhas de estilo com documentos, rotulação de direito de propriedade intelectual e rótulos PICS.

Berners-Lee (1998) comenta sobre a intenção da criação de aplicativos, que estejam dentro da Web, que possam combinar dados entre si. Esses aplicativos seriam capazes de usar uma *framework*<sup>1</sup> comum para combinar a informação de várias fontes. Exemplo: o aceite ou negação de permissão de um usuário por um sistema composto desses aplicativos, baseando-se em informações de lógica de controle de acesso, privacidade, pertinência a grupos de usuários e tipo de informação de dados. Por fim, sobre o aumento do nível de complexidade de *queries* sobre conjuntos de domínios cujos dados isolados, o autor diz que seriam de complexidade inexpressiva.

Berners-Lee *et. al.* (2001) valorizam a utilidade da Web Semântica com a múltipla oferta de serviços, com planejamentos em tempo real, executados por agentes inteligentes. Os autores citam o problema dos eixos de informação na web, diferindo a informação para humanos da informação para máquinas, onde, atualmente, possui mais informação para humanos do que para máquinas. E levantam o fato de que, assim como a web atual, a web semântica sofrerá uma descentralização, com o custo associado da contínua existência de links quebrados, tal como é hoje.

Os autores dizem que, sobre a representação do conhecimento na Web Semântica, ela está repleta de agentes inteligentes que são munidos de máquinas de inferência para simular um raciocínio automático.

---

<sup>1</sup> Na programação orientada a objetos, é um conjunto de classes projetadas para criação de aplicações com uma característica comum.

Fazendo um comparativo da utilidade dos paradigmas, os autores afirmam que a representação do conhecimento será para a Web Semântica o que o Hipertexto foi para a Web, e levantam mais dois problemas: a) O uso das tecnologias de representação do conhecimento na web ainda é incipiente, de forma que, apesar de existente, a Web Semântica ainda é uma utopia e deve ser globalizada; b) A padronização da representação do conhecimento ainda é centralizada, gerando sistemas gigantes, com gestão impossível.

O surgimento da Web Semântica traz consigo novos problemas. Para resolvê-los, Berners-Lee *et. al.* (2001) dizem que os sistemas de representação do conhecimento geralmente têm suas próprias regras estritas e idiossincráticas para fazer inferências sobre seus próprios dados, o que ocasiona, então, um novo problema, gerando a incompatibilidade entre os conjuntos de regras dos sistemas, apesar da possível compatibilidade entre seus conjuntos de dados. Os autores apontam o fato de que o preço da versatilidade da Web Semântica é o surgimento de paradoxos e questões sem solução. Para resolver estes problemas, os pesquisadores procuram criar linguagens expressivas o suficiente, de acordo com as necessidades, para permitir que agentes que atuam na web “raciocinem” artificialmente, da forma mais abrangente possível.

É uma filosofia similar àquela da web convencional: no início de seu desenvolvimento, pessoas céticas propunham o fato de que esta jamais seria organizada, não permitindo a procura por informações específicas, por possuir vários tipos e estruturas de bases de dados, não compatíveis entre si. Entretanto, Berners-Lee *et. al.* (2001) concordam com o argumento da heterogeneidade dos dados, e contra-argumentam que, com o tempo, os sistemas da web evoluíram, e neles as máquinas de procura, permitindo a indexação de grande parte do conteúdo desta rede.

Assim, o desafio da Web Semântica é providenciar uma linguagem que expresse tanto dados quanto regras, raciocinando sobre os dados, e que permita a exportação de regras de qualquer sistema de representação do conhecimento para dentro da web. Nesse sentido, os autores levantam as necessidades de regras de inferência para escolher cursos de ação, ou responder questões, citando o complicante de decisões envolvendo parâmetros matemáticos e de engenharia. Eles recomendam uma situação intermediária na programação da lógica destas regras, entre a capacidade de descrição de propriedades ou

objetos complexos, e a capacidade destes algoritmos de identificar que uma questão fornecida é um paradoxo.

## 3.2 MODELOS DE REPRESENTAÇÃO DE CONHECIMENTO

### 3.2.1 Dicionários

#### 3.2.1.1 IATE - Inter-Active Terminology for Europe

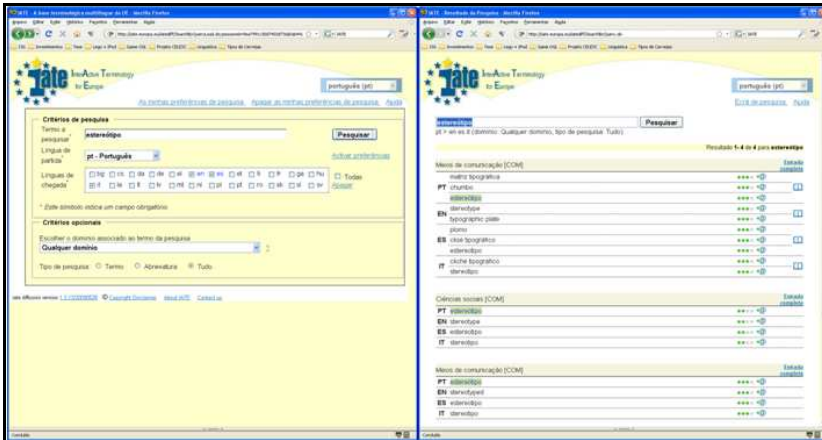


Figura 11: Interface do Sistema IATE de pesquisa de termos em vários idiomas. IATE(2008).

A IATE (2008) é uma iniciativa europeia de um sistema que permite a um usuário entrar com um termo procurado, em um idioma e domínio também escolhidos, retornando termos similares ao termo procurado, nesses idiomas, dentro do domínio determinado. O sistema gerencia uma base de dados interinstitucional de terminologias. A base é alimentada com termos novos pelos terminologistas e tradutores com base nas informações de tradutores, administradores, juristas/lingüistas, peritos e outras fontes fiáveis.

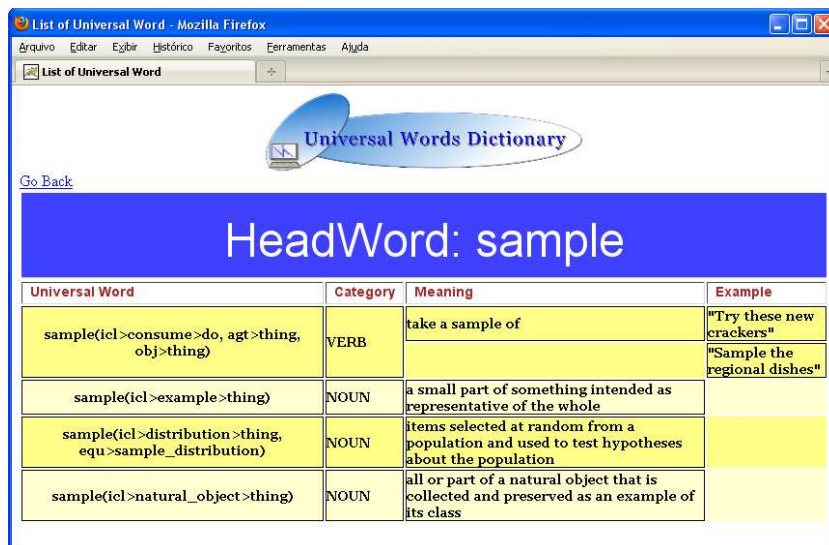
Na IATE, pode-se encontrar terminologia e gíria específicas da União Europeia, bem como termos de todas as áreas imagináveis, como Direito, agricultura, Tecnologia da Informação e muitas outras. Possui 8,4 milhões de termos, incluindo aproximadamente 540 000 abreviaturas e 130000 expressões, e cobre todas as 23 línguas oficiais da UE. O sistema consiste da união de bases terminológicas de instituições europeias. Ela reúne toda a informação anteriormente existente nas antigas bases terminológicas das instituições europeias, como:



EURODICAUTOM (Comissão); EUTERPE (Parlamento); TIS (Conselho).

### 3.2.1.2 Universal Words Dictionary

O Universal Words Dictionary (UWDIC, 2010) é uma página web disponível para consultas a palavras universais da UNL. Seu uso é útil para consultas a palavras chave, e na associação destas aos termos da ontologia do modelo de Bueno (2008).



Universal Word	Category	Meaning	Example
sample(icl>consume>do, agt>thing, obj>thing)	VERB	take a sample of	"Try these new crackers" "Sample the regional dishes"
sample(icl>example>thing)	NOUN	a small part of something intended as representative of the whole	
sample(icl>distribution>thing, equ>sample_distribution)	NOUN	items selected at random from a population and used to test hypotheses about the population	
sample(icl>natural_object>thing)	NOUN	all or part of a natural object that is collected and preserved as an example of its class	

Figura 12: Página web do Universal Words Dictionary.

As palavras universais, seus termos equivalentes e variantes e explicações sobre essas palavras, termos e equivalências, estão em estrutura tabular, em formato HTML, e as explicações a respeito da palavra universal desejada pode ser requisitada via sintaxe HTTP. Dessa maneira, é possível automatizar a consulta a palavras universais e aos seus significados, para auxiliar na pesquisa de palavras disponíveis pelo EC responsável por construir parte da ontologia multilíngüe.

### 3.2.1.3 Dicionários on-line e Wordnet

Outra alternativa que pode ser automatizada para auxiliar o trabalho de tradução e verificação de termos e palavras universais na construção de ontologias multilíngües são os dicionários internacionais on-line. Uma procura na internet retorna uma série de títulos de páginas

que oferecem serviços de tradução e classificação semântica de termos a partir de palavras entradas em campos texto, tal como um dicionário. Vários desses dicionários podem ser acessados por uma sintaxe simples com o protocolo HTTP, retornando os resultados das definições da palavra de entrada em página HTML.

Em virtude de existir um número crescente de dicionários internacionais, e que é possível acessá-los e receber seu resultado via internet, e que cada página possui seu formato de entrada e saída, nota-se a necessidade de criar scripts para envio de requisições HTTP e de *parsing* das páginas retornadas no formato HTML.

Quanto à Wordnet (2009), esta é uma base de dados léxica: substantivos, verbos, adjetivos e advérbios são agrupados em conjuntos de sinônimos cognitivos (*synsets*), cada um expressando um conceito distinto. Os *synsets* são interligados por meio de relações semânticas e léxicas. A rede resultante de palavras e conceitos com significado pode ser navegada com o browser. A WordNet está disponível publicamente para acesso e para download. A estrutura da WordNet a torna útil para as ciências da Linguística Computacional e Processamento de Linguagem Natural.

A WordNet pode ajudar no auxílio automatizado para consulta de termos, na construção de ontologias.

### 3.2.2 Ontologias

Segundo Gruber apud (Liu & Özsu, 2009), no contexto da Ciência da Computação e Informação, uma ontologia define um conjunto de primitivas de representação com as quais é modelado um domínio de conhecimento ou de discurso. As primitivas de representação são tipicamente classes (ou conjuntos), atributos (ou propriedades), e relacionamentos (ou relações entre os membros da classe). As definições das primitivas de representação incluem informações sobre o significado e as suas limitações em sua aplicação logicamente consistente. No contexto de sistemas de banco de dados, ontologias podem ser vistas como um nível de abstração de modelos de dados, análogo ao dos modelos hierárquicos e relacionais, mas que se destina à modelagem de conhecimento sobre os indivíduos, seus atributos e suas relações com outros indivíduos. Tipicamente, ontologias são especificadas em linguagens que permitem abstração além de Estruturas de Dados e Estratégias de Implementação. Na prática, as linguagens de ontologias estão mais próximas do poder expressivo de

linguagens de lógica de primeira ordem do que linguagens utilizadas para modelagem de bancos de dados. Por esta razão, se diz que as ontologias estão no nível "semântico", onde os esquemas de banco de dados são modelos de dados no nível "lógico" ou "físico". Devido à sua independência de modelos de nível inferior, ontologias são usadas para integrar bancos de dados heterogêneos, permitindo a interoperabilidade entre diferentes sistemas e especificando interfaces entre serviços independentes, baseados no conhecimento. Na pilha de tecnologia dos padrões da Web Semântica (Berners-Lee *et. al.* (2001)), ontologias são consideradas uma camada explícita. Atualmente há linguagens padrão e uma variedade de ferramentas de código aberto e comerciais para criar e trabalhar com ontologias.

Berners-Lee *et. al.* (2001) levantam a possibilidade de um mesmo conceito estar presente em duas ou mais bases diferentes, e por isso ser definido por identificadores diferentes; citam como exemplo o código postal de endereços (*zip code*). Para resolver o problema de múltiplos identificadores para um mesmo conceito, os autores apontam o uso de ontologias, definidas como documentos que contêm relacionamentos entre termos. São exemplos as ontologias de taxonomias e suas respectivas regras de inferência. Nesse sentido, as ontologias podem ser usadas para refinar uma procura, evitando ambigüidades de palavras, ou ainda relacionar as informações de uma página a estruturas de conhecimento ou regras de inferência. Os autores ainda prevêm a possibilidade da existência de páginas feitas exclusivamente com redes de ontologias, nas quais agentes poderão refinar termos ambíguos usados em outras páginas, para descobrir informações sobre um determinado tema.

Desse modo, 'ontologia' pode ser definida como um conjunto de termos para descrever e representar uma área do conhecimento, para que sejam usadas por pessoas, aplicações e bases de dados que precisam compartilhar informações sobre um dado domínio. Incluem definições de conceitos básicos e as relações entre eles, dentro de um domínio de conhecimento, para serem usadas por computadores. As ontologias têm em si codificadas o conhecimento que pertence a um domínio ou até mesmo vários domínios. Desta forma, o conhecimento torna-se reusável (OWL, 2004).

As ontologias têm sido utilizadas para descrever artefatos com diferentes graus de estrutura: taxonomias, esquemas de metadados, ou teorias lógicas. Há a necessidade de ontologias com diferentes níveis de

estrutura, e estas precisam de descrições para os seguintes tipos de conceito: classes de coisas, relações entre estas coisas e as propriedades que estas coisas devem ter. As ontologias são usualmente expressadas em uma linguagem de base lógica, de tal forma que distinções detalhadas, precisas, consistentes e relevantes podem ser feitas entre estas classes, propriedades e relações. Desta maneira, algumas ferramentas podem executar um raciocínio automático usando ontologias, oferecendo aplicações e serviços inteligentes tais como procura/recuperação conceitual ou semântica, agentes de software, suporte à decisão, compreensão de linguagem natural escrita e falada, gestão do conhecimento, bases de dados inteligentes e comércio eletrônico (OWL, 2004).

Por último, a previsão de que as ontologias serão usadas dentro da Web Semântica para representação da semântica de documentos, permitindo que esta semântica seja usada por aplicações web e agentes inteligentes. As ontologias podem provar que são úteis para uma comunidade como um caminho para estruturar e definir o significado dos termos metadados que são frequentemente coletados e padronizados. Desta forma, as aplicações podem trabalhar mais precisamente no nível conceitual humano (OWL, 2004).

Para Noy e McGuinness (2010), há a tendência dos processos de construção de ontologias passarem dos laboratórios de especialistas em inteligência artificial para as máquinas desktop de especialistas de domínio de conhecimento. Nesse sentido, os autores narram sobre algumas iniciativas de consórcios para a construção de linguagens para codificação de conhecimento em páginas web, que permitam que tal conhecimento seja processável por agentes inteligentes. Também apontam algumas aplicações de representação do conhecimento com o uso de ontologias, em que os especialistas podem compartilhar e anotar as informações em suas áreas. Por exemplo, no campo da Medicina, já foram produzidos grandes vocabulários padronizados e estruturados, tais como: a SNOMED (Price; Spackman, 2000) e a rede semântica da *Unified Medical Language System* (Humphreys; Lindberg, 1993). As ontologias de propósito gerais também estão emergindo, por exemplo: o Programa de Desenvolvimento das Nações Unidas e a empresa Dun & Bradstreet combinaram seus esforços para desenvolver a ontologia UNSPSC que oferece terminologias para produtos e serviços (Ver [www.unspsc.org](http://www.unspsc.org)). Segundo os autores, uma ontologia define um vocabulário comum para pesquisadores que precisam compartilhar informação em um domínio. Este vocabulário comum inclui definições

de conceitos básicos em um domínio de conhecimento e das relações entre estes conceitos, interpretáveis por máquinas.

Noy e McGuinness (2010) apresentam cinco razões para a construção de ontologias: 1) Compartilhar uma compreensão comum da estrutura de informação entre pessoas e agentes de software; 2) Permitir o reuso do conhecimento do domínio; 3) Explicitar suposições/hipóteses dentro de um domínio; 4) Separar o conhecimento do domínio do conhecimento operacional; 5) Analisar o conhecimento de um domínio. Para os autores, uma ontologia é composta por quatro elementos: Classes; Papéis; Restrições de Papéis e Instâncias.

### 3.2.3 RDF - Resource Description Framework

A RDF é uma linguagem para representação de recursos na *World Wide Web* (www). Ela serve para representar metadados sobre recursos Web, tais como: título, autor, data de modificação de uma página, *copyright*, informações de licença de um documento Web, ou o período de disponibilidade de um recurso compartilhado. Apesar disso, pela generalização do conceito de recurso Web, a RDF pode ser usada também para representar informação sobre coisas que podem ser identificadas na Web, mesmo que não possam ser recuperadas diretamente da Web. Alguns exemplos são as informações sobre especificações, preços e disponibilidade de itens de compras em lojas *on-line*, ou sobre a descrição de preferências de usuários a respeito de informações que devem ser disponibilizadas (RDF PRIMER, 2010).

A RDF foi construída para situações em que, além de serem apresentadas para os usuários, as informações disponíveis na 'www' possam ser processadas por aplicações, disponibilizando uma *framework* comum para expressar/comunicar informação entre máquinas, sem perda de significado. A RDF se baseia na idéia de que as coisas descritas na 'www' têm propriedades, que, por sua vez, possuem valores, e que os recursos podem ser descritos através de declarações, que determinam as propriedades e valores. Os componentes das declarações são denominados sujeito (elemento que possui a propriedade, e que se deseja descrever), predicado (a propriedade deste elemento) e objeto (o valor desta propriedade) (RDF PRIMER, 2010).

Explicando: Dado um assunto qualquer, descrito em uma página da web, este assunto pode ser descrito com conjuntos de afirmativas compostas de três partes: <sujeito, predicado, objeto>. Um exemplo é dado, na própria página, com informações sobre o livro *Jurassic Park*, e

alguns detalhes sobre a sintaxe são descritos: O livro tem um autor: Michael Crichton; foi lançado em 09/07/2006; e tem um comentário feito pelo usuário. Desse modo, verifica-se a facilidade e garantia da recuperação das informações contidas nas páginas web, sobre o assunto descrito, comparando o processo de descoberta de conhecimento em conteúdo *RDF* com os processos anteriores de desmonte de páginas *HTML* (*screen-scraping*). E também o processo maior de construção de um grafo semântico: dado que se duas duplas *RDF* descrevem o mesmo objeto, elas são conectadas (TWINE, 2010).

### 3.2.4 OWL - Web Ontology Language

A OWL foi criada com intenção de representar explicitamente o significado de termos em vocabulários, e as relações entre esses termos. Na OWL (2004), o grupo da W3C propõe uma visão de futuro, em que o conteúdo significa a informação disponível na Web, permitindo às máquinas fazerem o processamento do significado da informação. Também propõe uma linguagem de ontologias que pode descrever a semântica de classes e propriedades usadas em documentos Web, partindo do princípio de que a Web Semântica será construída sobre tags<sup>2</sup> XML e sobre representação de conhecimento com uso da RDF.

São descritas três sublinguagens, em ordem crescente de representatividade: a) A OWL-Lite, que suporta hierarquias de classificação, restrições simples, tais como: cardinalidade entre 0 e 1, e facilita a migração para formas de representação de conhecimento tais como taxonomias e thesauri; b) A OWL-DL, que correspondente à ciência da Lógica Descritiva, que permite a máxima expressividade para as conclusões, mas garantindo a completeza (garantia de que todas as conclusões são computáveis) e decidibilidade (todos os cálculos são feitos em tempo finito) destas conclusões. A OWL DL inclui todos os elementos de construção da linguagem, mas que somente podem ser usados sob certas restrições; c) A OWL-Full, que permite uma expressividade máxima e a liberdade sintática da RDF, mas sem garantias computacionais. Como exemplo citado, uma classe pode ser tratada simultaneamente como uma coleção de indivíduos e também como um indivíduo. A OWL-Full permite que uma ontologia aumente o significado de um vocabulário predefinido nos protocolos RDF e OWL. Os autores acreditam que seja improvável que qualquer aplicativo de

---

<sup>2</sup> Um marcador, em formato texto, para páginas em formato SGML \ XML \ HTML

inferência possa tratar o processo de raciocínio para todos os recursos da OWL Full.

### **3.2.5 UNL como linguagem de suporte a ontologias**

Cardeñosa *et al.* (2008) propõem a UNL como uma linguagem de suporte para tarefas orientadas ao conhecimento, devido às seguintes propriedades: flexibilidade, generalidade, neutralidade, número finito de relações para descrever qualquer conteúdo em linguagem natural e definição de procedimentos para criar novos conceitos.

Os autores afirmam que, apesar de as ontologias conterem muitos elementos, elas são compostas basicamente de conceitos e relações entre estes conceitos. A UNL oferece vários dos itens que precisam ser aprendidos quando o conhecimento ontológico é adquirido a partir de textos. E reconhecem que, apesar da existência destas propriedades nas ontologias, a UNL não é um formalismo para representar conhecimento, e tais formalismos ou não estão explícitos na linguagem ou até mesmo são completamente ausentes. Apesar disso, os componentes principais de uma ontologia podem ser prontamente extraídos das representações UNL.

Partindo do princípio de que da linguagem UNL podem ser extraídos elementos componentes de ontologias, Cardeñosa *et al.* (2008) fazem um comparativo entre os elementos pertencentes a uma ontologia e os elementos descritivos da UNL.

**Conceitos Ontológicos:** São conjuntos de classes dentro de um domínio. São descritos pelos seus atributos, pelas relações que possuem com outros conceitos, por regras e restrições. Conceitos são o vocabulário de uma ontologia e são definidos de maneira não ambígua. Comparando com a UNL, cada UW pode ser vista como o nome de um conceito, e a UNL pode disponibilizar ao Engenheiro de Ontologias um vocabulário de conceitos.

**Relações Ontológicas:** Descreve a interação entre dois conceitos ou mais. São classificadas em dois tipos: Relações verticais, que organizam conceitos em uma árvore hierárquica (são as relações que dizem que um conceito 'é' uma subclasse, uma particularização de algo), e Relações horizontais, que ligam conceitos ao longo de uma árvore de hierarquias. Alguns exemplos de relações horizontais são as de meronímia, localização no espaço e tempo, funcionais, procedurais, atributivas e causais.

A UNL suporta relações conceituais, funcionais e procedurais, contando com o fato de que possui poucos argumentos obrigatórios para dar uma representatividade básica ao conceito, e muitos argumentos opcionais, oferecendo grande flexibilidade no refinamento de conceitos (CARDEÑOSA *et. al.*, 2008).

**Atributos Ontológicos:** Servem para definição de conceitos e caracterização de instâncias. Exemplo de uma classe: “carro”; com seu atributo de “cor”, que pode representar qualquer instância, e justificam que é possível utilizar palavras universais do projeto UNL para a descrição de classes ontológicas e atributos de classes ontológicas. Entretanto, os atributos devem ser deduzidos a partir do conteúdo de uma ontologia. Este fenômeno ocorre porque a representação UNL contém um conhecimento não estruturado, enquanto a ontologia já possui este conhecimento estruturado.

Cardenosa *et. al.* (2008) sugerem que a conversão de uma ontologia para a UNL precisa de capacidades dedutivas que podem ser extraídas do conjunto de relações UNL ao redor da palavra universal envolvida. A identificação dessas características é feita usando o dicionário UNL construído a partir da *WordNet*. No exemplo do “carro”, citado anteriormente, este conceito estaria dentro do conceito de “coisa física” contendo um estado chamado “cor”. Seguindo esse exemplo, a partir da representação UNL de um pedaço de informação, tal como “O carro de John é vermelho”, é possível deduzir que “vermelho” é uma característica do “carro de John”. A expressão UNL não representa explicitamente a “cor” do carro de John, ela somente fala que ele é vermelho.

O exemplo também apresenta a forma como a UNL incorpora a distinção entre atributos de conceito e atributos de instância e entre os escopos local e global. Finalmente, muitas linguagens de ontologias permitem expressar o escopo de um atributo, que diz se o intervalo de conceitos atribuídos a este atributo é restrito a um simples conceito, a um subconjunto de conceitos (usualmente descendentes de um super conceito) ou a todos os conceitos definidos na ontologia.

**Instâncias de conceitos nominais:** A UNL possui uma relação conceitual específica para conectar indivíduos concretos (representados em textos por nomes próprios) com seus respectivos conceitos (classes): É a relação de nomeação (“nam”). Desta forma, indivíduos referidos por nomes próprios são ligados com suas classes em documentos UNL. Apesar de estas relações serem úteis na identificação de indivíduos



presentes em um pedaço de informação, não é tarefa do processo de codificação revelar indivíduos mencionados no texto por meios de artifícios linguísticos, tais como descrições definidas, demonstrativos, frases possessivas e assim por diante. Novamente seria tarefa dos procedimentos de aquisição do conhecimento extrair as referências às instâncias individuais da representação UNL.

Classificações taxonômicas mais complexas: A UNL oferece a relação taxonômica “is a” (é um, referindo-se que um elemento é uma classe do tipo), mas não pode expressar decomposições disjuntas ou exaustivas. Por outro lado, informação negativa ou incerta pode ser representada na UNL já que todos os links são rotulados com uma medida de confiabilidade.

Restrições: Os autores descrevem as restrições em vários níveis. As restrições sobre o número e tipo dos argumentos de uma relação são expressas na UNL: A definição de 41 relações conceituais impõe certa restrição sobre o tipo de palavra universal que pode ser conectada por uma delas. Palavras Universais Verbais têm uma determinada posição na base de conhecimento, e por isso os tipos e papéis dos argumentos dessas palavras universais também são restritos.

Os possíveis atributos ou recursos de muitos conceitos nominais podem ser restringidos pela sua participação em conceitos bem gerais (por exemplo: coisa viva, coisa física, etc.). E também os valores de alguns atributos são restringidos por um número limitado de opções (por exemplo: o conceito de “cor” pode somente ter como valor um dos conceitos (ou instâncias) específicos classificados como “cor”). Apesar disso, deve ser notado que existem restrições que refletem o conhecimento de senso comum que foi linguisticamente materializado. Pelo mesmo motivo, não se espera que restrições específicas de tarefa ou domínio (por exemplo, as placas de carros consistem de um arranjo de números e letras, ou o comprador de um item caro é elegível para um desconto) sejam encontradas na UNL. O mesmo pode-se dizer de valores *default*, que são completamente dependentes da aplicação e podem ser modelados nos formalismos da representação do conhecimento. Obviamente, se essas restrições específicas de domínio e valores *default* forem adquiridas a partir de fontes textuais, a aquisição dessas restrições a partir da representação UNL dos mesmos textos deve ser muito mais simples.

Funções e axiomas: esses tipos de construções não são suportados pela UNL. A UNL implementa axiomas que intencionalmente

expressam a transitividade da relação “icl” e a herança de restrições de conceitos mais gerais para conceitos mais específicos. Mas, mais do que isso, a UNL não foi projetada para expressar funções ou axiomas arbitrários.

Possibilidades de inferência: A UNL não possui capacidades de inferência, já que o seu objetivo é representar significado linguístico, embora a UNL possua recursos limitados de raciocínio sobre as relações determinadas entre as palavras universais. As capacidades de raciocínio são requeridas para verificar a correta representação do conhecimento na sintaxe UNL e reforçar as restrições usadas na representação do conhecimento, e para providenciar suporte para o processo de codificação UNL (desambiguação, análise sintática e interpretação semântica) e geração de texto (por exemplo: substituir as palavras universais não conectadas com o dicionário de uma dada linguagem por outras semanticamente próximas ou conectadas).

Devido à possibilidade de representação de textos em vários idiomas, e à possibilidade de conter informação semântica embutida em sua representação, a UNL é usada para ter um protocolo de conversão do significado dos termos da linguagem humana – em qualquer idioma – para os elementos UNL contidos em bases. A tecnologia serve como um contêiner de equivalentes de cada um dos termos de cada idioma. É possível, portanto, representar o significado de termos e expressões, escritos também em qualquer idioma para, a partir desses termos e expressões, fazer uma estruturação do conhecimento contido em textos multilíngues. Nesse contexto, a representação UNL de termos tem a utilidade de permitir a conversão de um termo, dentro de um domínio, em um idioma, em um termo dentro de um determinado domínio, em outro idioma.

### 3.2.5.1 Elementos da UNL

A Palavra Universal Simples é composta de três elementos: 1) A *headword*, ou a palavra que define o termo, independente do domínio; 2) As restrições de função, que indicam se a palavra é um substantivo, adjetivo, verbo ou advérbio; 3) As restrições de domínio, que indicam que a palavra define um conceito dentro de um domínio específico do conhecimento. Dessa forma, a palavra universal tem o seguinte formato: UW = HW([RF,] [RD]).

A UW pode estar composta somente pela *headword*, ou se possuir alguma restrição, esta pode ser somente de função, somente de

domínio, ou as duas restrições simultaneamente. Alguns exemplos de palavras universais:<sup>3</sup> Casa: house(icl>legislature> thing); Civil: civil (icl>adj); Presidência: presidency(icl>abstract\_thing); República: republic (icl> political\_system>thing); Subchefia: UW Nominal: head(icl>leader> thing); Assuntos: UW Nominal: subject(icl>message>thing); Jurídicos: UW Adjetival: juridical (icl>adj, com>law).

A relação UNL e a Palavra Universal Composta: os exemplos de palavras vistos acima são de palavras simples. Ainda há um segundo tipo de palavra universal, a palavra composta, que é construída com as palavras universais simples e mais as relações semânticas que indicam a função de cada uma das palavras dentro de uma sentença.

Palavras universais compostas são construídas por relacionamentos entre palavras universais e tem o seguinte formato:  $UW = R1(UW_i, UW_j); R2(UW_i, UW_k); R3(UW_j, UW_n); R_i [...]$ , onde  $R_1, R_2, \dots, R_i$  são as relações semânticas UNL entre cada uma das palavras universais  $UW_1, UW_2, \dots, UW_i, UW_j$ . Exemplos de palavras universais compostas, construídas a partir de palavras universais simples, vistos anteriormente, com sua justificativa:

---

<sup>3</sup> O documento parcialmente convertido para a sequência de palavras universais está na página LEI N° 9.279, DE 14 DE MAIO DE 1996.  
[http://www.planalto.gov.br/ccivil\\_03/Leis/L9279.htm](http://www.planalto.gov.br/ccivil_03/Leis/L9279.htm)

Presidência da República (Considerar também que é o título principal, ou de primeiro nível):	A expressão inicial	
mod(presidence(icl>abstract_thing) @entry, republic(icl>political_system>thing))	Uma relação modificadora de valor ( <b>mod</b> ) que diz que o conceito de <b>presidência é da república</b>	A expressão Presidência da República
shd(presidence(icl>abstract_thing), title(pof>book))	Uma relação de cabeçalho de sentença ( <b>shd</b> ), que indica que a sentença <b>Presidência da República</b> é um <b>título</b> , que por sua vez é parte de um conceito chamado <b>livro</b>	Um conjunto de palavras universais indicando que a expressão Presidência da República é um Título de nível 1, ou o título principal do documento descrito.
aoj(title(pof>book), level(icl>property>thing) @ordinal)	Uma relação de atributo de objeto ( <b>aoj</b> ) que diz que o conceito nível ( <b>level(...)</b> ) é um atributo do conceito título ( <b>title</b> ), e que é um valor que possui uma ordem ( <b>@ordinal</b> )	
mod(level(icl>property>thing) @ordinal, 1)	Uma relação modificadora de valor ( <b>mod</b> ) que diz que o conceito de nível ( <b>level(...)</b> ) tem o valor de <b>1</b> (um).	

Quadro 1: Exemplo de método para construção de UW's, a partir do conteúdo de um documento da Web.

### 3.3 METODOLOGIAS DE REPRESENTAÇÃO DO CONHECIMENTO

#### 3.3.1 RC2D

O RC2D é uma metodologia de representação de conhecimento cujo enfoque está centrado no modelo teórico do Raciocínio Baseado em Casos (RBC). É um processo dinâmico de aquisição do conhecimento de textos, definido através da elaboração de um vocabulário controlado e um dicionário de termos, associado a uma análise de frequência das palavras e expressões indicativas do contexto. Outros desenvolvimentos semelhantes, como a UNL, a *Semantic Web* e a *WordNet* trabalham com a marcação do texto integral, definindo a sua classificação sintática. No RC2D, isso não é necessário; apenas as partes realmente significativas

são representadas. Essa técnica realiza comparações entre o contexto dos documentos, possibilitando a realização de uma busca mais precisa e com maior qualidade; foi desenvolvida especialmente para apoiar implantações que utilizem algoritmos com Inteligência Artificial (HOESCHL, 2001).

O RC2D consiste num processo de análise do contexto geral que envolve o problema focado. Ele realiza comparações entre o contexto dos documentos, possibilitando a realização de uma busca mais precisa e com mais qualidade. Além disso, os documentos são recuperados através de índices pré-determinados, que podem ser valorados pelo usuário quando da consulta.

Hoeschl (2001) classifica o processo RC2D em uma etapa teórica e outra empírica: 1) Etapa teórica: diz respeito à análise de documentos e textos referentes ao assunto focado, e posterior processo dialético, envolvendo especialistas na área de recuperação documental e/ou no assunto abordado pelo sistema, a fim de identificar quais expressões são relevantes e caracterizadoras dos assuntos tratados nos documentos. No caso específico, foram analisados os principais documentos orgânicos da ONU (Carta, Regimento do Conselho de Segurança, relatórios, listas oficiais de Países, resoluções, etc). 2) Etapa empírica: consiste na experimentação feita com as expressões extraídas do processo teórico sobre os documentos que farão parte do sistema, bem como análise numérica sobre a ocorrência das expressões. *In casu*, tanto a base de casos da ONU quanto os documentos selecionados para a construção do sistema foram objeto do processo empírico. Também foram levadas em consideração as estatísticas sobre incidência das expressões nos documentos, dado utilizado para inclusão/ampliação de índices, ou supressão de alguns deles, ou decomposição ou, ainda, agrupamento.

### **3.3.2 Engenharia da Mente**

Segundo Bueno (2008), as entidades usadas no modelo da Engenharia da Mente são: termo, relação e domínio. A seguir, os seus significados:

**Termo:** São elementos de representação de um conceito isolado. Um termo tem significados diferentes, dependendo do assunto (domínio como explicado mais abaixo) que o contém. Além da classificação de termos de acordo com o assunto, o termo é construído com um artifício, para reduzir sua ambiguidade: Termos são descritos como um conjunto de várias palavras.

**Relação:** Inicialmente, é uma conexão entre dois termos, que determina se o conceito representado por um termo é uma classe do conceito representado pelo outro termo, ou uma parte deste outro termo, ou ainda, se estes dois termos são semelhantes ou co-ocorrem em um mesmo texto. Uma relação é determinada por uma regra, que define como os termos conectados por ela interferem na interpretação do conhecimento contido em um documento. Pensando em usos futuros do modelo de representação do conhecimento, dado que uma relação é definida pelos seus termos conectados, e por regras programáveis que determinam a interpretação do conhecimento representado pela máquina, esta relação pode ser criada pelo próprio engenheiro do conhecimento. Dois conceitos podem conter várias relações entre si, dependendo do domínio a que estas relações pertencem.

Para o sistema de construção de ontologias, por conveniência, a relação de sinonímia, por ter um número muito grande de ocorrências, é graficamente representada como um conjunto. Essa forma de representação é menos custosa em performance, do que representar cada relação de sinônimo entre os vários termos.

Como consideração inicial, caso as relações de sinônimos fossem representadas por arestas, que fossem desenhadas como linhas simples, para  $N$  termos sinônimos entre si, existiriam  $N * N$  linhas simples desenhadas na tela. Para o caso de representação de grupos de sinônimos rodeados por um conjunto, o número de linhas desenhadas na tela é de somente quatro, para uma envoltória retangular, ou de oito linhas, para uma envoltória octaédrica.

**Domínio:** Visualmente representa um conjunto de conceitos. Semanticamente, representa um assunto, uma área do conhecimento sobre o Termo. É uma forma de classificação de termos e de relações entre termos, por um assunto específico. Os domínios podem ser encadeados em vários níveis, inclusive com continência múltipla (um domínio pai pode conter vários domínios filhos) e pertinência múltipla (um domínio filho pode pertencer a vários domínios pais).

### 3.3.2.1 Modelos de domínio da metodologia da Engenharia da Mente

Bueno (2008) apresenta três modelos: no primeiro modelo (Figura 13), a UW é integrada através de outros sistemas ou do próprio desenvolvimento UNL (BUENO, 2008).

No segundo modelo (Figura 14), a UW é parte integrante da classe Termo, tendo uma relação de equivalência com esta classe, ou, explicando de outra forma, tendo a relação equivalente à relação de sinônimo entre Termos. Observa-se, neste modelo, a questão do idioma é central na definição de uma estrutura de construção de ontologias multilíngüe e que esta classe não é contemplada nestes dois modelos (Figura 2).

Assim, Bueno (2008) apresenta uma nova proposta de modelo de domínio (Figura 15). O modelo apresenta seis Classes Conceituais: 1) Idioma: Define a coleção de idiomas que será tratada pelo sistema; 2) UW de Domínio: Define a expressão em UNL que fornece o conceito-chave que nomina o domínio e subdomínios de forma Universal, isto é, compreensível por qualquer idioma; 3) Domínio: Correspondente, em um determinado idioma, da UW de Domínio ao qual está relacionado; 4) UW de Termo: Define a expressão em UNL que fornece o conceito-chave que nomina os termos de forma Universal, isto é, compreensível por qualquer idioma; 5) Termo: Correspondente em um determinado idioma da UW de Termo ao qual está relacionado; 6) Relacionamentos: Entidade que liga dois termos distintos no contexto de um domínio, estabelecendo uma associação de hierarquia (pai e filho).

Esta associação pode ser de duas categorias distintas: a) Relação Tipo de (hiperonímia e hiponímia): É a conexão existente entre expressões da qual se depreende uma relação de categoria e classe, ou gênero e espécie. É a relação existente entre um lexema mais específico ou subordinado e um lexema mais geral ou superordenado. Ex.: cocaína é um tipo de substância entorpecente. b) Relação Parte de (meronímia e holonímia): É aquela relação que determina uma idéia de fração e o todo, sendo mais comumente encontrada na estrutura de organizações (ex.: Secretaria do Ministério do Planejamento é parte de Ministério do Planejamento) ou nos casos em que a descrição das partes é tão relevante quanto à previsão do todo.

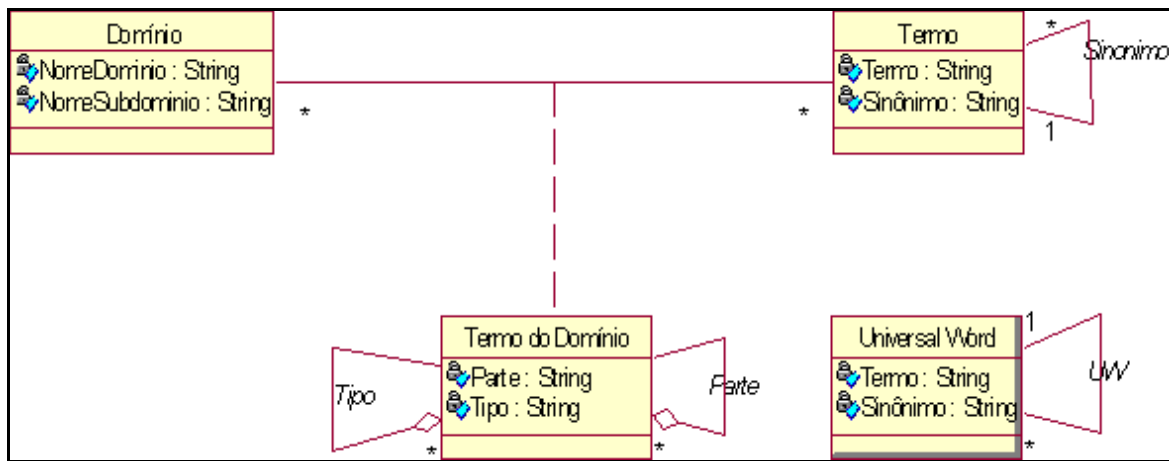


Figura 13:Primeira versão do modelo de domínio. BUENO (2008).

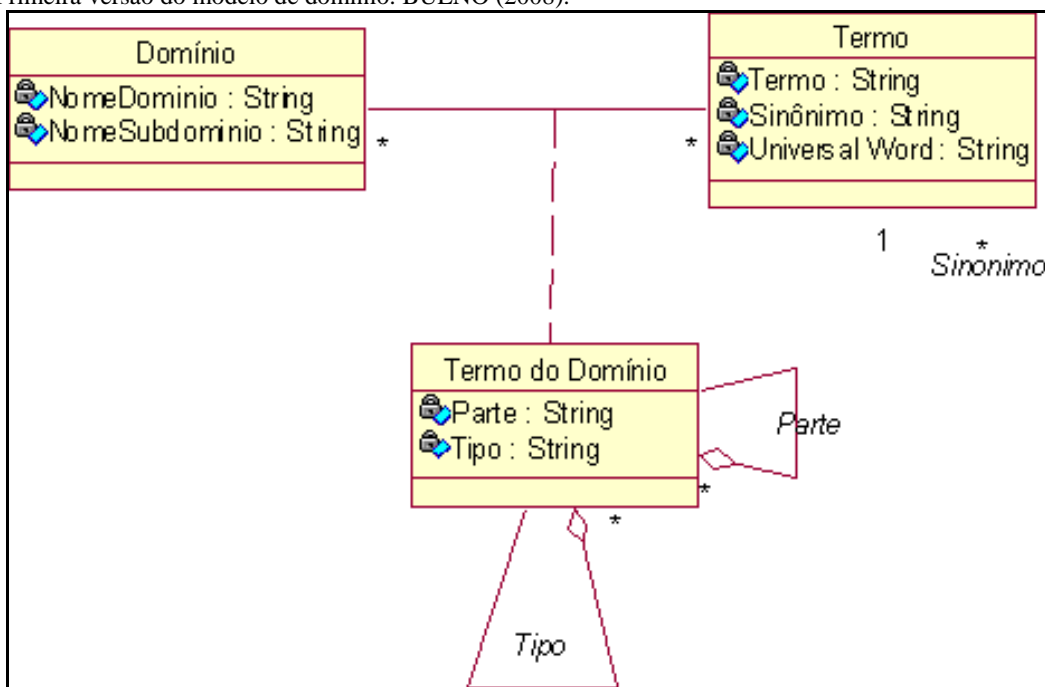


Figura 14:Segunda versão do modelo de domínio. BUENO (2008).

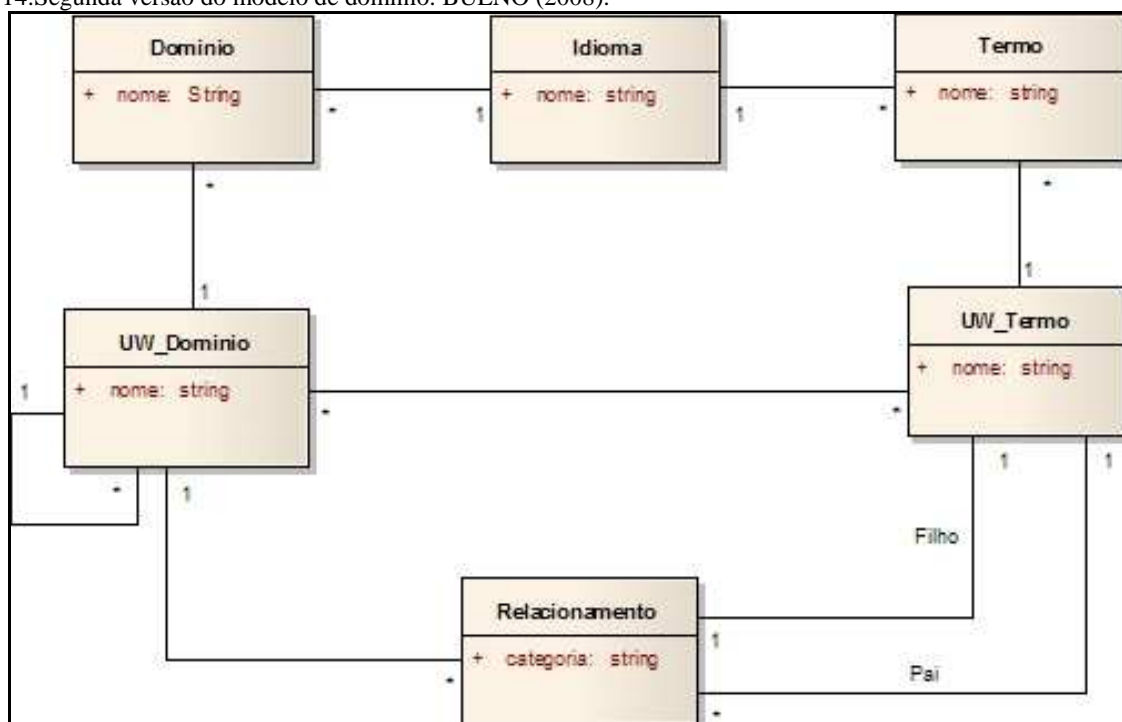


Figura 15:Modelo de domínio aplicado no Editor Multilíngüe. BUENO (2008).



Um termo só pode estar associado a um relacionamento ‘Tipo de/Parte de’ somente uma vez enquanto contido. Um termo pode estar associado a um relacionamento ‘Tipo de/Parte de’ várias vezes enquanto contenedor. Isto é, o relacionamento ‘Tipo de/Parte de’ de um termo só pode estar contido uma única vez e um termo pode conter muitos outros termos no relacionamento ‘Tipo de/Parte de’. Além disso, como o relacionamento é feito baseado nos conceitos universais (UW de Termo e UW de Domínio), este modelo permite que um amplo número de pessoas consiga construir suas ontologias independentemente do idioma. Portanto, este modelo cujo um dos objetivos é tornar o ambiente multilíngue.

Fixando o trabalho nos conceito de UW, o modelo permite criar um repositório de palavras universais e também um dicionário universal, ao associar as relações de Tipo e Parte em palavras universais. A associação entre as classes termo e UW de termo permite definir a tradução para o idioma do usuário como também criar a relação de sinônimo, pois todos os termos que forem associados a uma mesma UW possuem o mesmo significado.

A UML inclui um forte modelo de agregação (composição simbolizada por um sólido losango negro), que implica a apropriação das partes pelo agregado que não foi utilizado no presente modelo.

Uma das características mais importantes deste modelo é permitir que as ontologias iniciadas por usuários de uma determinada língua possam ser concluídas por usuários que falam outro idioma, porque a construção não está associada ao idioma e sim ao conceito universal (UW).

### *3.3.2.2 Premissas*

Para Bueno (2008), a Engenharia da Mente engloba o estudo de pessoas, processos e tecnologias representativas de um domínio através de três premissas: 1) Compartilhamento do Conhecimento; 2) Visualização; 3) Definição de Relevâncias.

A premissa de ‘Compartilhamento do Conhecimento’ é composta de três fases: a) A identificação do vocabulário subjetivo, que alinha o conhecimento subjetivo da equipe, com o conhecimento formal do domínio, tendo a conseqüências da integração da equipe, da ajuda ao engenheiro de ontologias a entender o conhecimento do especialista, e da identificação das ambigüidades do vocabulário usado pelos

especialistas. b) A normalização deste vocabulário, que ajuda a estabelecer o vocabulário usual entre o Engenheiro de Ontologias e o Especialista do Domínio. Esta comunicação de vocabulário acelera a tomada de decisão, e prepara a equipe de Engenheiros de Ontologias para a realização do inventário para a construção dos dicionários de termos e suas relações semânticas entre estes termos, no domínio trabalhado. c) O inventário de fontes de informação, que consiste no levantamento das principais fontes de informação referentes ao contexto. Tais fontes servirão como referencia para a avaliação da qualidade das ontologias produzidas.

A premissa da ‘Visualização’ consiste da organização e contextualização do conhecimento adquirido através de mapas conceituais. Ela permite a visão das informações de uma forma ampla. Desta forma, é possível selecionar expressões que não estão adequadas ao domínio estudado, permite construir um vocabulário controlado através de mapas conceituais, e permite compreender as relações entre os termos selecionados de um domínio.

A premissa de ‘Definição de Relevâncias’ consiste na divisão da equipe para o trabalho de construção de ontologias, determinando a competência entre os Engenheiros do Conhecimento e os Especialistas do Domínio. Esta premissa ajuda a definir a dimensão social (até onde cada profissional envolvido deve interferir, qual o impacto do resultado final deste processo de representação de conhecimento para o cliente final, e quais as comunidades que serão beneficiadas por este processo de representação de conhecimento) do contexto trabalhado. A definição de relevâncias também auxilia na compartimentação do vocabulário dos domínios: são conhecidos quais subdomínios pertencem a um determinado domínio e quando dividir um domínio de extensão considerável (com muitos termos) em subdomínios a ponto de facilitar a compreensão de seu papel nos processos de representação do conhecimento contido nos documentos.

Fases		Definição	Importância	Aplicação na Engenharia de Ontologias
Compartilhamento do Conhecimento	Identificação do vocabulário subjetivo	Identifica e organiza os condicionamentos do conhecimento subjetivo do domínio na equipe em relação ao conhecimento formal do domínio.	Facilita a integração da equipe e auxilia o Engenheiro de Ontologias a entender o conhecimento especialista.	Identifica as ambigüidades do vocabulário utilizado pelos especialistas e o conhecimento formal do domínio.
	Normalização do Vocabulário	Estabelece o vocabulário usual entre o Engenheiro de Ontologias e o Especialista do domínio.	Melhora a velocidade na tomada de decisão e prepara a equipe para realização do inventário. Na etapa de construção dos dicionários, melhora a coerência das relações entre termos.	Define os principais significados dos termos utilizados, denominados expressões indicativas.
	Inventário de Fontes de Informações	Levantamento das principais fontes digitais de informações referentes ao contexto.	As fontes servirão com referência para avaliação da qualidade das ontologias.	Auxilia a manutenção das ontologias na base de conhecimento.
Visualização	Organizar e contextualizar o conhecimento através de mapas conceituais.	Permite a visão das informações de uma forma ampla. Bem como selecionar as expressões que não estão adequadas ao contexto.	Permite a Construção do Vocabulário Controlado através de mapas conceituais e compreender as relações entre eles.	
Definição de Relevância	Nesta fase a equipe divide o trabalho de construção de ontologias, determinando as competências entre Engenheiros do Conhecimento e os Especialistas do Domínio.	Define a dimensão social do contexto trabalhado e os assuntos relacionados (por exemplo: redução da criminalidade e/ou reforma política).	Define a subdivisão do vocabulário controlado em Domínio e Subdomínios.	

Quadro 2: As três premissas da metodologia Engenharia da Mente. Fonte: BUENO (2008).

O modelo social do processo de construção de ontologias da autora é composto por três tipos de profissionais:

- a) Os especialistas do domínio, que possuem as informações sobre os assuntos, termos utilizados e suas relações de significado, que serão inseridos nos sistemas de representação do conhecimento.
- b) Os Engenheiros de Ontologias, que selecionam o vocabulário representativo do conhecimento daqueles especialistas e constroem as ontologias que servirão de base para os processos de procura daqueles sistemas de representação de conhecimento.
- c) Os analistas de sistema, que se encarregam de implementar os processos automáticos ou semi-automáticos de transformação de conhecimento tácito em conhecimento explícito, dentro dos sistemas de representação do conhecimento.

### *3.3.2.3 Validação de Ontologias construídas com a metodologia da Engenharia da Mente*

Bedin (2007) descreve a construção de um sistema de gestão de informações que subsidia políticas públicas de desenvolvimento na área de Saúde Ambiental, Saúde Humana e Violência Urbana:

O sistema-alvo de estudo busca estruturar informações que vão subsidiar políticas públicas de desenvolvimento para a região metropolitana por meio de monitoramento, recuperação e composição da base do conhecimento sobre Saúde Ambiental, Saúde Humana e Violência Urbana na Região Metropolitana de Curitiba. O sistema conta com um dicionário de termos construído por especialistas nas áreas definidas como objeto de representação (BEDIN, 2007).

E, para tal, a autora sugere uma metodologia de validação de ontologias de BUENO(2005), sistematizando recomendações sobre estabelecimento de relações semânticas entre termos dessas ontologias, para definir critérios para futuras validações. Apresenta também resultados sobre a consistência dessas relações, dentro de uma rede experimental de ontologias:

A presente pesquisa propõe uma metodologia de validação das ontologias que integram este dicionário, considerando os critérios lingüísticos que reflipam o uso da língua no

contexto comunicativo, visto que se trata de uma linguagem utilizada para recuperar conteúdos cuja função é comunicar. Apoiando-se no estudo de publicações já reconhecidas nas áreas de Ciência da Informação e Linguística, foram sistematizadas recomendações sobre o estabelecimento das relações semânticas entre os termos, a fim de definir critérios de validação de ontologias sob o ponto de vista lingüístico. Esses critérios foram aplicados em um “corpus” de 28 termos e 163 relações semânticas. O resultado da análise apresenta 82,21% de relações semânticas consideradas consistentes e 17,79% inadequadas. Para este último conjunto, foram tecidas recomendações. A aplicação de uma metodologia de fundamentação lingüística sob a perspectiva do uso foi a obtenção de uma ontologia atualizada, tanto no seu aspecto lexical como semântico, aumentando a eficiência de recuperação da informação (BEDIN, 2007).

O método proposto é um ciclo que contém as seguintes etapas:

1. É feito um levantamento de um grupo de domínios e subdomínios, com seus respectivos dicionários de termos de trabalho.
2. Como referencia para a avaliação das relações semânticas, foram levantados - a partir de bibliografias - alguns critérios para as relações de equivalência, relações hierárquicas e associativas. Abaixo são vistos alguns critérios e suas justificativas:
  - a. **Relações de equivalência** permitem a substituição completa de termos.
  - b. **Relações hierárquicas** compartilham algumas características dos termos gerais e específicos (classe), ou descrevem dois termos onde um faz parte integrante do outro (parte).
  - c. **Relações de associação** descrevem o mesmo tema, mas não são de equivalência, nem hierárquicas.
3. É feita uma análise quantitativa das relações, para determinar a prioridade de análise sobre a coerência de relações construídas nas ontologias observadas, através da frequência de ocorrência destas

relações (na pesquisa da autora, as relações mais frequentes foram as de equivalência).

4. As relações observadas nas ontologias construídas foram classificadas em três tipos (Sinônimo, Hierárquica e Associativa) e dois grupos (Adequadas e Inconsistentes).

Abaixo é mostrada uma tabela com os resultados de amostragem dos tipos de relações encontrados durante a aplicação do método de (BEDIN, 2007) em um domínio de ontologia:

**Tabela 1: Tipos e grupos de relações semânticas para a análise do impacto de redes de ontologias sobre ferramentas de busca (copiado de BEDIN, 2007)**

RELAÇÃO	ADEQUAÇÃO	INCONSISTÊNCIA
SINÔNIMO	50	15
HIERÁRQUICA	24	11
ASSOCIATIVA	60	03
<b>TOTAL</b>	<b>134</b>	<b>29</b>

5. É feita a análise qualitativa das relações – com base nos critérios levantados na bibliografia, para determinação de relações semânticas. A análise qualitativa é separada em dois grupos: Análise das relações semânticas julgadas Adequadas, e Análise das relações semânticas julgadas Inconsistentes:

a. Para o grupo de relações Adequadas foi retirada uma amostra de tríades (termo, relação semântica, termo), e para cada uma destas amostras a sua adequação foi justificada com base nos critérios para as relações de equivalência, relações hierárquicas e associativas

b. Para o grupo de relações Inconsistentes foi retirada uma amostra de tríades (termo, relação semântica, termo), e para cada uma destas amostras, a sua inconsistência foi justificada também com base nos critérios vistos na etapa 3, e foram propostas novas relações entre os termos de cada tríade para torna-las coerentes com os critérios para as relações de equivalência, relações hierárquicas e associativas

6. Por último, para testar o impacto das novas relações semânticas adaptadas dentro do domínio de ontologia escolhido, na precisão do

resultado de busca, são feitas novas buscas de documentos a partir de palavras-chave de teste, e observados alguns parâmetros da lista de documentos resultantes da busca, tais como genericidade ou especificidade do assunto dos documentos, quantidade de documentos retornados, ordem dos documentos recuperados, e pertinência destes documentos recuperados a um contexto esperado.

7. Por ultimo, o ciclo é repetido, desde a etapa 3) até a 6), até que se encontre uma lista satisfatória de documentos retornados como resultados de busca.

### 3.4 FERRAMENTAS DE CONSTRUÇÃO COLABORATIVA DE ONTOLOGIAS

#### 3.4.1 OntoKEM

Rautenberg *et. al.* (2008) apresentam a ontoKEM (*Ontology Knowledge Engineering and Management*), uma ferramenta em plataforma Web construída pelo grupo do curso de Pós-Graduação em Engenharia e Gestão do Conhecimento, da Universidade Federal de Santa Catarina, para apoiar o processo de construção e documentação de ontologias. O processo de construção para o qual o ontoKEM foi concebido está baseado nas metodologias de construção de ontologias On-to-Knowledge, Methontology e no Guia 101.

Os autores enumeram os pontos das três metodologias que serviram como base para a implementação do sistema:

1. On-to-Knowledge: metodologia que utiliza questões de competência como modo simples e direto para determinar o escopo de uma ontologia e permite identificar conceitos, propriedades, relações e instâncias. O OntoKEM faz uso de questões de competência na compreensão da aplicabilidade da ontologia, e disponibiliza um artefato para documentá-las.

2. Methontology: metodologia que formaliza a construção de ontologias através de uma rica gama de artefatos de documentação (documentos-texto e quadros). Estes artefatos são usados como modelos de documentos no OntoKEM.

3. Guia 101: metodologia que prega a construção de ontologias num processo iterativo de sete passos (determinar o escopo da ontologia, considerar o reuso, listar termos, definir classes, definir propriedades,

definir restrições e criar instâncias). Este processo interativo é adotado no OntoKEM.

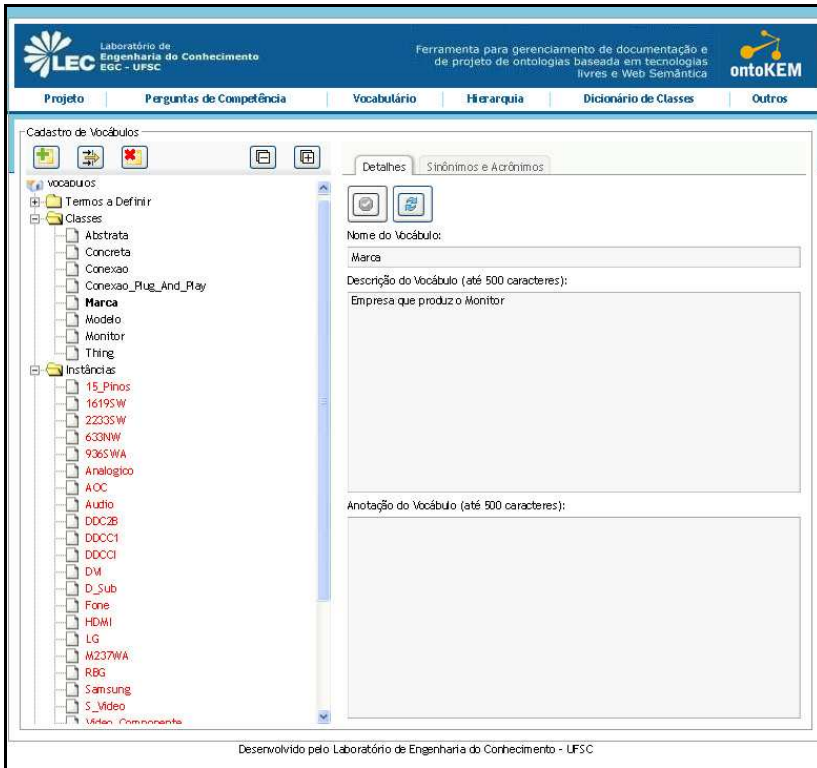


Figura 16: Interface web da ferramenta OntoKEM para construção de ontologias. RAUTEMBERG et. al. (2008).

### 3.4.2 Twine

Na página da Twine (2008), é apresentado o conjunto das tecnologias usadas na respectiva página para permitir aos usuários organizar, compartilhar e descobrir informações de seu interesse. A tecnologia base do sistema Twine foi projetada para representar propriedades e classes e determinar seus usos, através de uma ontologia própria, e também através da OWL, para que possa se comunicar com outros sistemas através deste protocolo.



### 3.4.3 Suíte de Engenharia do Conhecimento

Bueno (2008) apresenta uma ferramenta denominada Suíte de Gestão de Conhecimento, que permite a construção de aplicativos *web*. Tem as seguintes funções: a) Para o engenheiro do conhecimento (EC): programar observadores de fontes *web* de conhecimento; determinar domínios de conhecimento; definir termos representativos de conceitos. b) Para o cliente: efetivar procura por assuntos específicos, utilizando textos completos e linguagem natural, dentro de determinados intervalos de tempo; pesquisar a frequência de aparição de documentos sobre determinado assunto, e o grau de semelhança dos documentos encontrados com o texto procurado.

A Suíte de Engenharia do Conhecimento permitiu a construção de alguns sistemas de conhecimento: MONGEE, Ontojuris, Ontoweb e SERPI, descritos a seguir.

#### 3.4.3.1 MONGEE-Monitoramento On-Line de Gases de Efeito Estufa

Segundo Bedin *et al.* (2008), o sistema MONGEE (Figura 17) foi criado com a intenção de organizar o conhecimento sobre redução na emissão de gases de efeito estufa no setor madeireiro do Planalto Norte Catarinense, para acelerar, transparecer e compartilhar os processos de tomada de decisão:

[...] um Sistema Inteligente de Monitoramento de Informações que traz em seu escopo o objetivo de tornar os processos de tomada de decisão e organização do conhecimento muito mais céleres, transparentes, compartilhadas e eficientes através do monitoramento de informações para redução dos Gases de Efeito Estufa do Setor Madeireiro do Planalto Norte Catarinense. É um Sistema de Gestão de Informação com o uso de Inteligência Artificial e Ontologias que permite atingir resultados tecnológicos inovadores capazes de avaliar as práticas adotadas pelas empresas para reduzir os gases de efeito estufa, monitorar e integrar informações relativas à gestão ambiental das atividades de silvicultura e transformação madeireira na região, bem como facilitar iniciativas e processos de melhoria da gestão e certificação ambiental.

### 3.4.3.2 *Ontojuris*

Segundo Bueno (2009), o projeto Ontojuris (Figura 18) foi criado para facilitar o acesso a informações legais multilíngues na área de Propriedade Intelectual, Direito do Consumidor e Direito Eletrônico, em um consórcio formado por países ibero-americanos. Sua tecnologia se baseia na tecnologia de representação do conhecimento através de ontologias e de palavras universais:

O projeto ONTOJURIS tem por finalidade facilitar o acesso a Informações sobre legislação na área de Propriedade Intelectual, Direito do Consumidor e Direito Eletrônico no consórcio formado pelo Brasil, Chile, Espanha e Argentina. Para isso, conta com uma rede social estabelecida entre estes países e com uma estrutura computacional, representada pelas tecnologias de busca inteligentes do instituto i3G e de universal words da Universidade Politécnica de Madrid. O Sistema Inteligente de busca baseia-se em ontologias, que permitem o mapeamento dos assuntos e termos relacionados às matérias tratadas pelo sistema, possibilitando pesquisas contextualizadas. [...]

A singularidade desta tecnologia permite a realização de consultas com grandes volumes de texto e destaca-se na nova era da Internet, na qual semântica e ontologias trabalham juntas para incrementar o processo de busca de informações relevantes em documentos digitais. Além disso, a utilização de universal words aliada a ontologias em português, espanhol e inglês permite ao ONTOJURIS ativar uma sistemática completamente inovadora na localização de documentos, por considerar o contexto do assunto que está sendo pesquisado em vários idiomas.

### 3.4.3.3 *Ontoweb*

Hoeschl (2009), em um rápido levantamento sobre as características de ferramentas de busca, e separando-as em gerações, descreve a ferramenta de gestão de conhecimento especializada em Governo Eletrônico, chamada Ontoweb (Figura 19):

As ferramentas de busca nasceram praticamente junto com a interface www da internet. Na primeira geração, tivemos os diretórios (Yahoo! e

similares). Logo em seguida, vieram os robôs (spiders) e as tecnologias automatizadas (Altavista). A terceira geração veio com os metabuscadores (Miner's). Logo em seguida, veio mais refinamento na organização dos resultados (All The Web). O Google, reunindo sofisticação e muita abrangência, trouxe o Page Rank para a web, e marca a quinta geração. Juntar vários tipos de arquivos diferentes em uma mesma busca (textos e imagens, por exemplo) é o foco da sexta geração (A9), que está se desenvolvendo (HOESCHL, 2009).

O autor caracteriza a ferramenta como tendo recursos de seleção de informações através de análise inteligente de conteúdo, com uso de ontologias e inteligência artificial, e sendo o primeiro buscador do mundo a trabalhar com ontologias e estruturas de avaliação de relações entre as palavras existentes nos textos procurados:

A sétima geração é marcada pela qualidade na seleção das informações, por meio das análises inteligentes de conteúdo, e ela já está na internet com o Ontoweb, um incrível e inteligente motor de busca, baseado em ontologias e técnicas de inteligência artificial, que é capaz de “pensar” durante a seleção das informações.

O Ontoweb é o primeiro buscador do mundo a trabalhar com engenharia de ontologias, e é fruto de um projeto de desenvolvimento científico, utilizando semânticas e estruturas valorativas para contextualizar as buscas e refinar os resultados. O uso das ontologias é um dos principais segredos da nova ferramenta (daí o nome Ontoweb), e é também um dos principais fatores responsáveis pela alta qualidade dos seus resultados (HOESCHL, 2009).

#### 3.4.3.4 Sistema SERPI

Segundo Hoeschl *et. al.* (2005) e Oenning *et. al.* (2006), o Sistema SERPI (Figura 20) foi criado pelo Núcleo de Tecnologia da informação de Santa Catarina (NECATI), para facilitar a conexão entre pesquisa, proteção e gestão industrial orientada a negócios. O instrumento apresenta, em um ambiente integrado, dados e informação consolidada com a intenção de identificar problemas institucionais, permitindo o planejamento de investimentos em Pesquisa e

Desenvolvimento. A ferramenta é usada por uma rede comercial envolvendo os setores produtivos e acadêmicos do estado, com a intenção de atrair mais recursos e redirecionar investimentos para além do mercado local.



Figura 17: Telas de pesquisa e de resultados do Sistema Mongee.

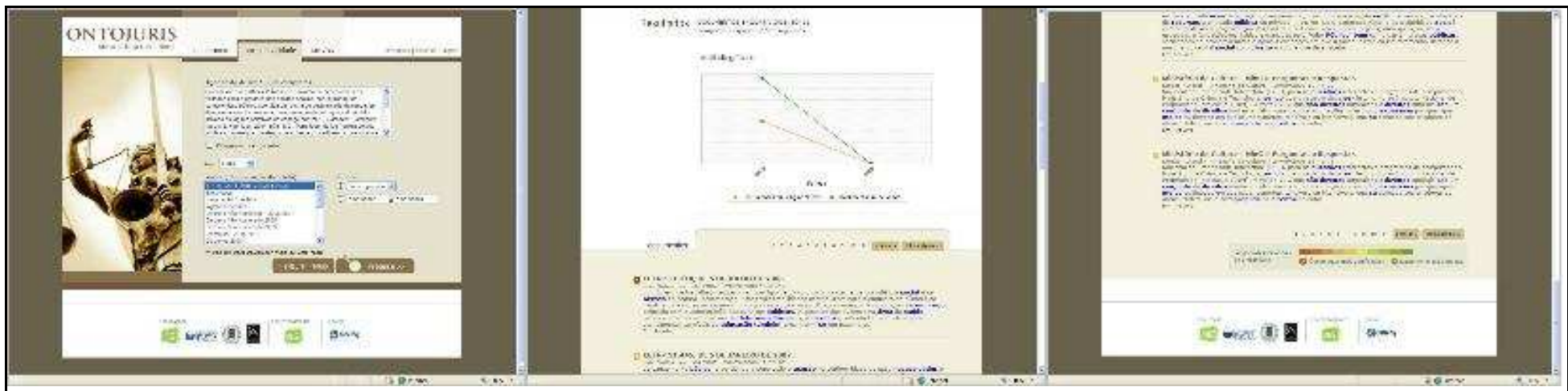


Figura 18: Telas de pesquisa, frequência de resultados, resultados e legenda do Sistema Ontojuris.

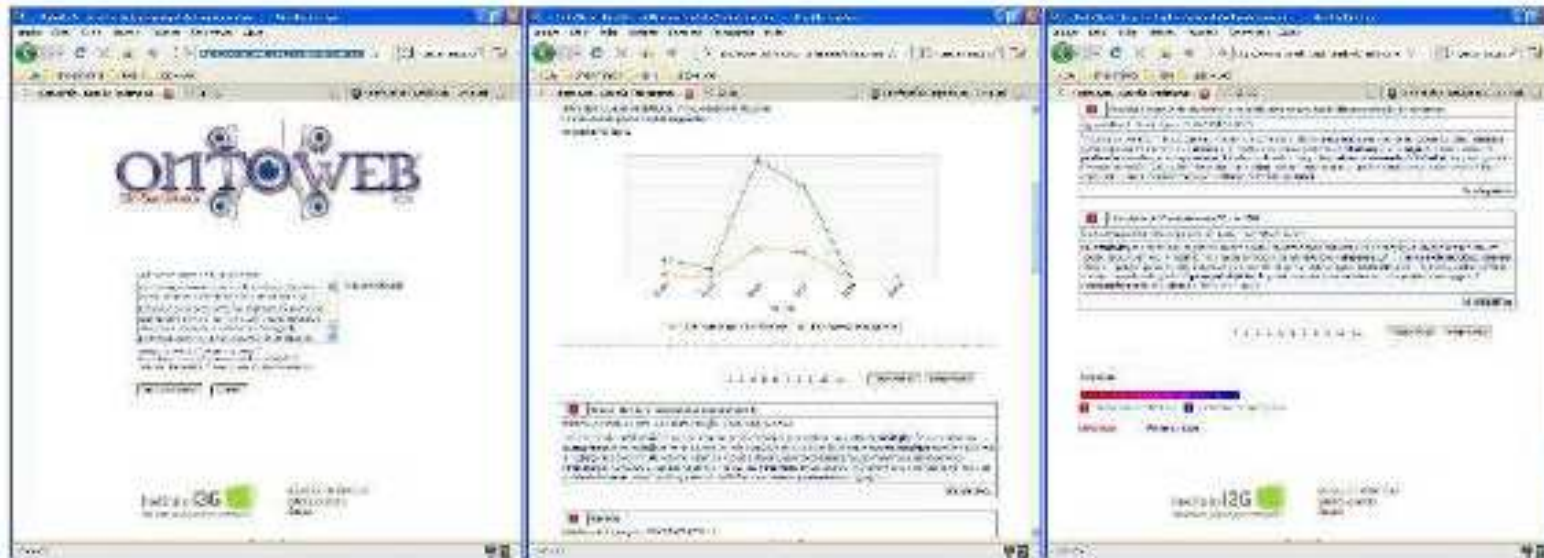


Figura 19: Telas de pesquisa, frequência de resultados, resultados e legenda do Sistema Ontoweb.

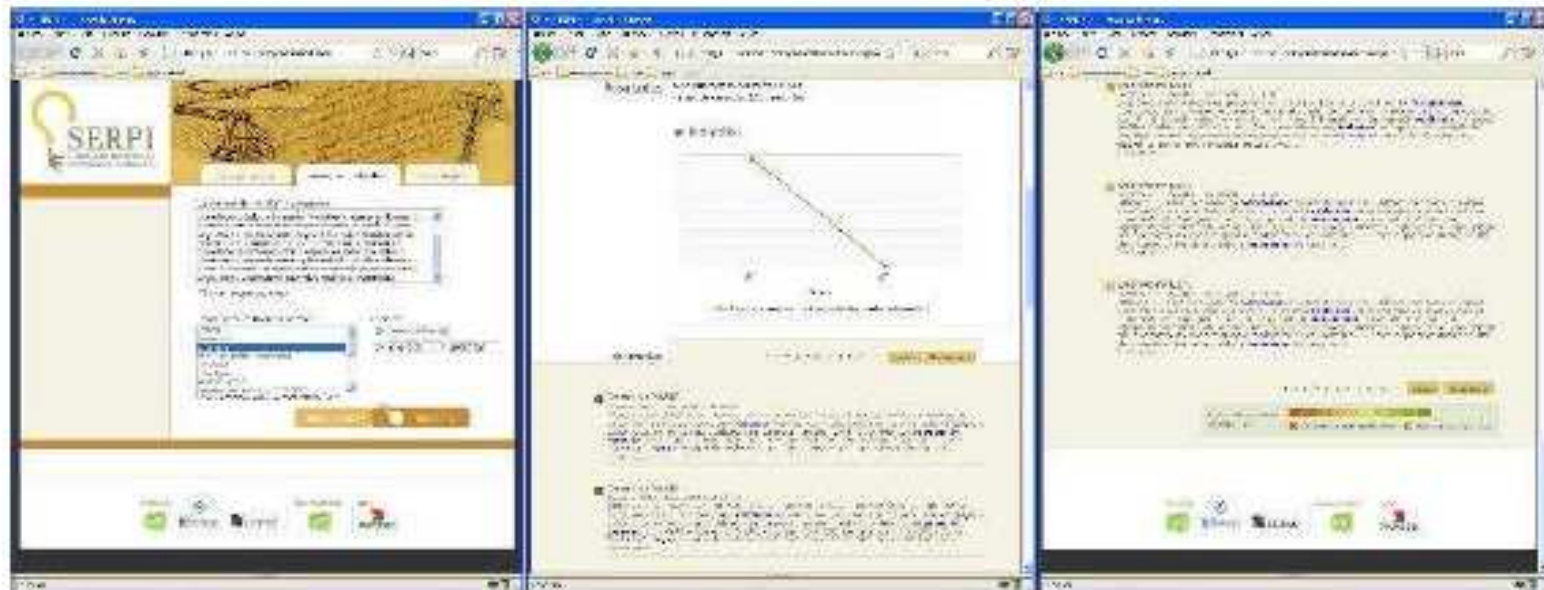


Figura 20: Telas de pesquisa, frequência de resultados, resultados e legenda do Sistema SERPI.

### 3.4.4 Protégè

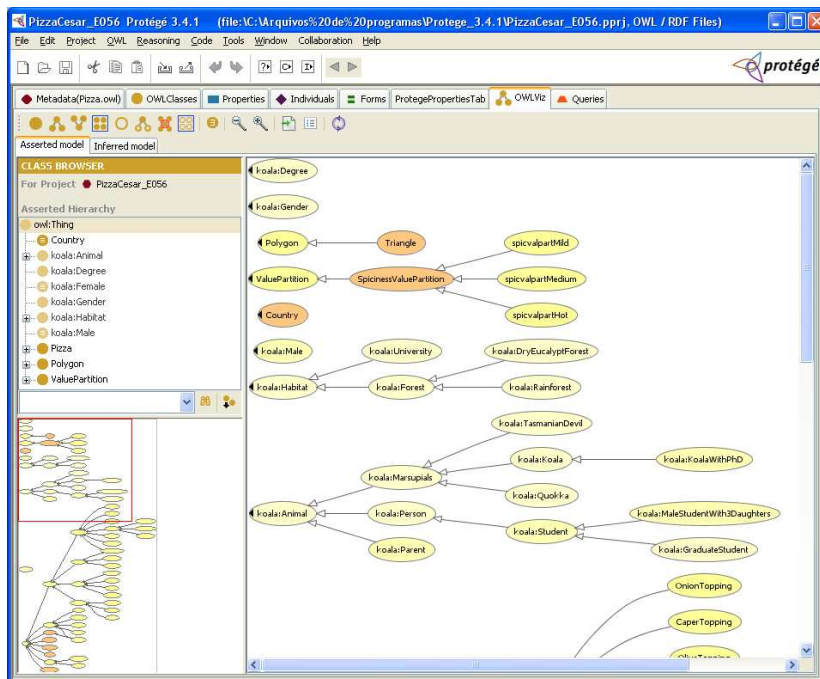


Figura 21: Interface do plug-in OWLViz no Protégè.

O aplicativo Protégè (2010) (Figura 21, Figura 22) é uma ferramenta de construção e consulta de conhecimento em ontologias, através de *queries*. O sistema trabalha com as três variantes de ontologias OWL: Lite, DL e Full, possuindo algumas variações em relação ao formato OWL original, o Protégè-OWL.

A partir do exemplo didático de construção de uma ontologia sobre pizzas, mostrado em Horridge *et al.*, (2004), é sugerido como construir um modelo de classes compatível com o modelo OWL-DL. O aplicativo também permite a exportação e importação de ontologias a partir de um endereço HTTP. Então, o EC pode definir classes, suas propriedades e objetos das classes criadas.

É possível também definir as condições das classes, a partir da lógica quantificacional Buchsbaum (2006) (Figura 22). Desta forma, o aplicativo pode determinar, com o auxílio de uma máquina de inferência (Racer Pro, 2009), e dentro de um conjunto gigantesco de classes, quais



classes são filhas diretas de outras classes, sem a necessidade da análise completa de um time de EC's.

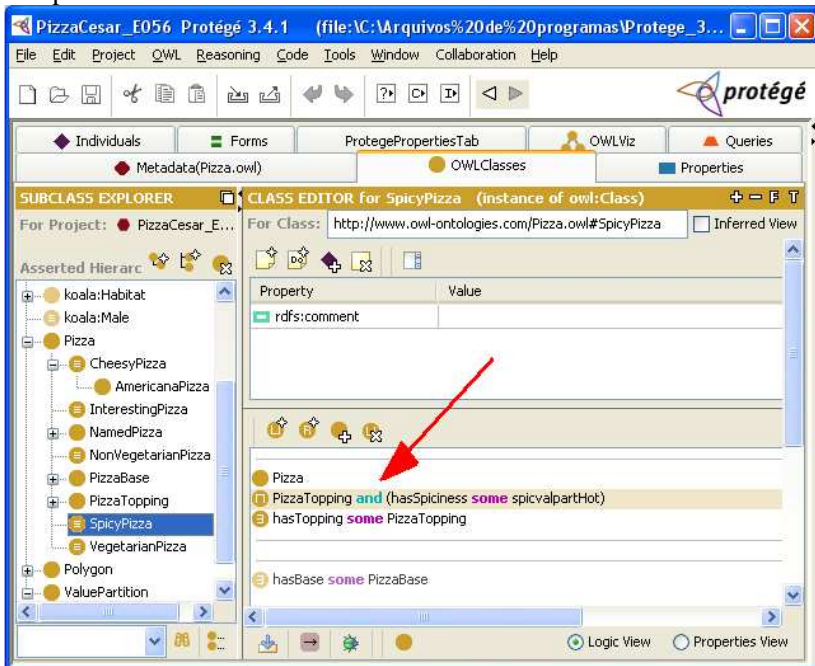


Figura 22: Determinação de condições para as classes, utilizando lógica quantificacional.

Horridge *et al.* (2004) mostra outro exemplo contendo o conjunto de condições para a classe *SpicyPizza* (pizza apimentada). Os autores estabelecem três condições necessárias e suficientes (apontadas pela seta vermelha na Figura 22): 1) A classe *SpicyPizza* é herdada da classe base *Pizza*; 2) A classe *SpicyPizza* possui uma instância da classe *PizzaTopping* (o molho da pizza) e também (intersecção  $\cap$ ) tem, pelo menos ( $\exists$ ), a propriedade **hasSpiciness** (acidez da pimenta), com um valor Forte (**spicvalpartHot**); 3) A classe *SpicyPizza* possui pelo menos ( $\exists$ ), a propriedade **hasTopping** com alguma instância de *PizzaTopping*.

Quanto à variedade de funções, o Protégé permite o acoplamento a vários plug-ins, sendo que alguns já vêm disponíveis na instalação do aplicativo, outros são desenvolvidos à parte e tem seu código ou instaladores específicos disponíveis na internet.

Finalmente, após ou durante a construção da ontologia, o time de analistas pode consultar o conhecimento contido nas ontologias criadas



a partir de *queries*, que são preenchidas a partir das propriedades das classes que representam os objetos da ontologia.

#### 3.4.4.1 Método 101

Noy e McGuinness (2010) apresentam um método para a construção de ontologias, utilizando a ferramenta Protégé. Inicialmente, enfatizam os seguintes princípios: a) Não há um meio correto de modelar um domínio; b) O desenvolvimento de uma ontologia é um processo iterativo; c) Os conceitos dentro da ontologia devem ser próximo aos objetos físicos ou lógicos, representados. Recomendam os seguintes passos na construção de uma ontologia:

1. Determinar o domínio (qual o conjunto de conhecimento que uma ontologia vai representar) e escopo (qual finalidade) da ontologia, utilizando questões básicas, e também questões de competência, que serão as questões as quais se deseja que a ontologia responda depois de construída.

2. Reutilizar ontologias existentes, para poupar trabalho, considerando que o problema do formato das ontologias pode ser abstraído devido á existência de várias ferramentas que executam a importação / exportação de ontologias<sup>4</sup>.

3. Enumerar os termos importantes de uma ontologia: Quais e o que gostaríamos de falar sobre estes termos? Que propriedades têm estes termos?

4. Definir as classes e sua hierarquia: Podendo se dar de três formas: Das classes mais genéricas para as classes mais específicas; Das classes mais específicas para as classes mais genéricas; ou uma combinação entre estas duas formas, iniciando com os conceitos mais salientes. Rosch *apud* Noy e McGuinness (2010) sugerem que a abordagem da construção combinada é a mais adequada, já que as classes intermediárias tendem a ser as mais descritivas do domínio.

5. Definir as propriedades das classes: Dada a lista de termos gerada na primeira etapa, os projetistas de ontologias selecionam quais

---

<sup>4</sup> Noy e McGuinness (2010), nesta etapa, sugerem algumas referências de métodos de ontologias reusáveis na Web, tais como: a Ontolingua (<http://www.ksl.stanford.edu/software/ontolingua/>); a biblioteca de ontologias DAML (<http://www.daml.org/ontologies/>); e as bibliotecas comerciais: UNSPSC ([www.unspsc.org](http://www.unspsc.org)), RosettaNet ([www.rosettanet.org](http://www.rosettanet.org)) e DMOZ ([www.dmoz.org](http://www.dmoz.org)).

termos são classes, e quais termos são propriedades. Após esta diferenciação inicial, estes projetistas determinam quais classes são descritas pelas propriedades.

Os autores diferenciam dois tipos de propriedades: a) Propriedades intrínsecas das classes, que não podem ser alteradas; b) Propriedades extrínsecas, que podem ser determinadas de acordo com a conveniência de uso destas classes em aplicações; Partes, que compõem as classes e as instâncias destas classes; e Relações com outros indivíduos.

6. Definir as restrições das propriedades: São a cardinalidade e os tipos das restrições (se são números inteiros, *strings*, valores booleanos, enumerações ou instâncias).

Os autores acrescentam mais dois conceitos referentes a uma restrição: Domínio e Abrangência. O Domínio é a classe que possui a propriedade. A Abrangência aborda as instâncias de uma determinada classe que servem como propriedade da classe Domínio.

Noy e McGuinness (2010) ainda apresentam um conjunto de regras para simplificar ou reprojeter a abrangência de classes: a) Se uma lista de classes definindo um intervalo ou um domínio de uma propriedade inclui uma classe e a sua subclasse, remova a subclasse; b) Se uma lista de classes definindo uma abrangência ou um domínio de uma propriedade contém todas as subclasses da classe A, mas a própria classe A, a abrangência deveria conter somente a classe A não as suas subclasses; c) Se uma lista de classes definindo uma abrangência ou um domínio de uma propriedade contém quase todas as subclasses de uma classe A, considere o caso em que a classe A poderia ter uma melhor definição de abrangência.

Nesta etapa da determinação de propriedades das classes, os autores comparam ontologias em que se pode determinar a pertinência de uma propriedade a uma classe através de duas formas: Ou a ontologia permite criar uma lista de propriedades, e incluir classes dentro desta lista, informando desta maneira que a classe dentro desta lista terá estas propriedades, ou a ontologia permite a inclusão da propriedade dentro de uma classe. Para estes dois casos, há uma função de equilíbrio (*trade-off*) entre a genericidade da propriedade, e a garantia de que a classe realmente necessita daquela propriedade.

7. Criar Instâncias: A última etapa na construção da ontologia é composta de três subetapas: escolher a classe; criar uma instância

individual para aquela classe; e preencher os valores de suas propriedades.

### 3.4.5 Growl

A ferramenta Growl, baseada na biblioteca *Prefuse*, permite a visualização de ontologias OWL e DL na forma de redes de relacionamento (KRIVOV *et. al.*, 2007). A ferramenta é baseada na semântica DL das ontologias OWL e suporta edição visual. Está implementada como uma Applet Java, um *plug-in* para o aplicativo Protégé e uma aplicação Java standalone. A construção atual do aplicativo é de código aberto. O aplicativo está sendo usado em um portal para ajudar na previsão de valores econômicos de serviços de ecossistema. (Figura 23).

Para casos de ontologias extensas, os autores contam com as capacidades de filtragem que o aplicativo oferece, durante a navegação da rede formada pela instância das ontologias carregadas.

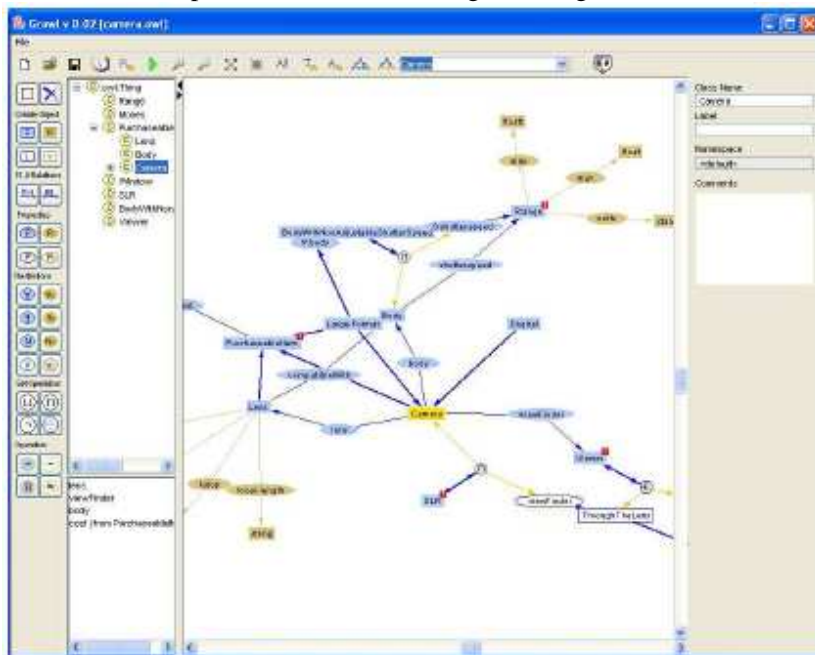


Figura 23:Ferramenta GROWL para visão de ontologias OWL e DL, na forma de redes de relacionamento.

### 3.4.6 Comparação de Outras Ferramentas

Noy et. al. (2008) descrevem o evento CKC Challenge (2007). Este evento colaborou para o fornecimento de elementos para a construção de ferramentas de suporte para a construção colaborativa de ontologias. Os autores identificaram que muitas dessas ferramentas estão ainda em desenvolvimento. Por isso, houve a necessidade de organizar o evento Collaborative Knowledge Construction Challenge (CKC Challenge, 2007) dentro da 16th International World Wide Web Conference que ajudou a descobrir o estado da arte dessas ferramentas.

Entretanto, os autores fazem uma distinção entre a Web 2.0 e a Web Semântica, em que a primeira oferece recursos para a criação de conhecimento não estruturado, somente com recursos de criação, compartilhamento, anotação e conexão de conhecimento, enquanto a segunda oferece uma semântica bem definida, com uso de ontologias, esquemas de bases de dados e taxonomias com definições explícitas.

Noy et. al. (2008) também apontam as características das ferramentas que integram as abordagens da Web 2.0 e Web Semântica, como Semantic MediaWiki<sup>5</sup>, BOWiki<sup>6</sup> e Platypus Wiki<sup>7</sup>. Elas oferecem extensões Wiki para criar links semânticos entre páginas. Outras ferramentas permitem organizar rótulos em alguma estrutura semântica. Outras ainda mais avançadas, como a OWL, suportam a distribuição e desenvolvimento colaborativo de ontologias. Ferramentas comerciais, como a Freebase<sup>8</sup>, também estão entrando no ramo. Eles avaliaram seis ferramentas de construção colaborativa no evento (Quadros Quadro 3 e Quadro 4).

---

<sup>5</sup> <http://meta.wikimedia.org/wiki/SemanticMediaWiki>

<sup>6</sup> <http://onto.eva.mpg.de/bowiki>

<sup>7</sup> <http://platypuswiki.sourceforge.net>

<sup>8</sup> <http://freebase.com>

Ferramenta	Descrição	Instituição Desenvolvedora
BibSonomy (www.bibsonomy.org)	Um sistema social de compartilhamento de recursos que permite que usuários organizem, rotulem e compartilhem marcações e referências bibliográficas. Os usuários podem criar relações entre os rótulos. Por exemplo, programação <= Java indica que o rótulo <b>Programação</b> é mais genérico do que <b>Java</b> . Desta forma o sistema pode incluir recursos rotulados com <b>Java</b> o usuário procura pelo assunto <b>Programação</b> .	Universidade de Kassel
Collaborative Protégé (http://protege.stanford.edu)	Um editor de ontologias e instâncias que suporta histórico de conceitos, discussões e comentários tanto sobre os componentes de ontologias quanto sobre as alterações nas ontologias. Por exemplo, os usuários podem sugerir ou explicar mudanças, criar propostas de mudanças, e votar sobre estas propostas.	Universidade de Stanford
DBin2 (http://dbin.org)	Uma aplicação ponto a ponto (P2P) que permite aos usuários editarem bases de conhecimento de maneira colaborativa. A interface <i>brainlet</i> (uma interface específica de domínio gerada pelo sistema DBin a partir de uma coleção de componentes) inclui componentes para edição de ontologias e aquisição de instâncias. O sistema permite descobrir a proveniência da informação para classes e propriedades.	Università Politecnica delle Marche
Hozo3 (www.hozo.jp)	Um editor de ontologias que permite aos usuários – assincronicamente – desenvolver ontologias que são sub-divididas em múltiplos módulos interconectados. Um usuário pode agir de maneira privilegiada sobre um módulo (operação de <i>check-in</i> ), e depois devolver o módulo atualizado para a base (operação de <i>check-out</i> ). A ferramenta Hozo usa dependências declaradas entre módulos para identificar conflitos. Os usuários podem ver a lista de mudanças e aceitá-las ou rejeitá-las.	Universidade de Osaka
OntoWiki4 (http://3ba.se)	Uma ontologia baseada na Web um editor de instâncias que providencia o histórico e permite a avaliação destes elementos. A ferramenta OntoWiki permite visões diferentes sobre as instâncias de dados (por exemplo, uma visão de mapa para dados geográficos ou uma visão de calendário para datas).	Universidade de Leipzig
Soboleo	Um sistema baseado na Web que permite aos usuários associar rótulos a recursos Web e organizar estes rótulos em hierarquias. Os usuários podem procurar através dos recursos Web anotados usando conceitos de uma taxonomia.	Centro de Pesquisa FZI para Tecnologia da Informação

Quadro 3: Comparativo de ferramentas de construção colaborativa. Fonte: Noy et. al. 2008.

	BibSonomy	Collaborative Protégé	DBin	Hozo	OntoWiki	Soboleo
Hierarquia de Conceitos	X	X	X	X	X	X
Propriedades		X	X	X	X	
Instâncias de conceitos na hierarquia (inclui tags)		X	X	X*	X	X
Comentários nos componentes da ontologia		X	X		X*	
Avaliações		X			X	
Edição assíncrona de módulos da ontologia				X		
Espaço pessoal	X			X		
Histórico de mudanças		X	X*	X (module level)	X	
Discussão		X	X*			X
Chat Instantâneo		X				X
Criação de conteúdo usando botões integrados no navegador	X					X
Interface Web	X				X	X

Quadro 4: Recursos das ferramentas. Fonte: NOY et. al. (2008). \* Recursos disponíveis em algumas versões das ferramentas, mas não na versão usada no evento.

Outras características anotadas por Noy *et. al.* (2008): todas as ferramentas pesquisadas permitiam a criação de informação hierárquica. Três delas focavam na construção de ontologias. Uma delas focava na integração de módulos de ontologias e identificação de conflitos, e duas delas focavam em rotular recursos web e criar taxonomias especificando relações entre estes rótulos. Além dessas, os autores incluem a possibilidade de criar *chats* e listas de discussão, avaliação e inclusão de comentários sobre componentes da ontologia, criação de um espaço de trabalho particular do usuário, separado do ambiente compartilhado, geração e publicação de *logs* de mudança da ontologia, suporte para construção assíncrona, e bloqueio de edição (operações de *check-in* ou *lock*).

Quanto aos cenários para a construção colaborativa de conhecimento, Noy *et. al.* (2008) apontam as ontologias Collaborative Protégé e Hozo como os produtos finais esperados, e também como anotação de recursos web com rótulos e como hierarquia de rótulos como resultado de anotações colaborativas – sendo que esta hierarquia de rótulos auxilia na navegação e procura desses recursos (BibSonomy e Soboleo) – e ainda, como produto final, espera-se a aquisição de conhecimento, permitindo aos usuários criar e compartilhar bases de conhecimento e entrar novas instâncias de dados usando interfaces personalizadas (DBin ed OntoWiki).

O método utilizado pelos autores permitiu a participação dos usuários. Cada participante deveria construir conhecimento estruturado para um portal hipotético para pesquisa de informação. Os pesquisadores sabiam que as ferramentas possuíam estruturas diferentes. Portanto, esperava-se que a informação exata que cada usuário inserisse no sistema seria diferente. Noy *et. al.* (2008) adotaram um processo de participação do usuário como se fosse um jogo. Os usuários foram estimulados a ter a maior atividade, e o registro do nível dessa atividade era publicado, de forma que os próprios usuários pudessem medir seu desempenho e a dos demais concorrentes. Os pesquisadores procuraram incentivar o uso de mais de uma ferramenta pelos usuários. Ex: fazer trinta mudanças em uma única ferramenta contava menos do que fazer dez mudanças em cada uma de três ferramentas diferentes. Edições na segunda ferramenta tinham mais valor do que edições feitas na ferramenta preferida do usuário. A concorrência entre usuários era notória: os dois usuários mais ativos fizeram a entrada de mais de duas mil afirmações cada, ao longo das duas semanas. Ao final, os

participantes preencheram formulários descrevendo a experiência. Noy *et. al.* (2008) apresentam os resultados contendo os recursos das ferramentas comparadas no Quadro 4.

Durante o experimento, além dos resultados, os autores observaram algumas tendências, como: a escolha de somente uma ferramenta ou de ferramentas semelhantes por um mesmo usuário: dos 49 usuários registrados, 33 participaram ativamente. Desses, metade tentou somente uma ferramenta. Os organizadores receberam 36 formulários de feedback: 31 falaram especificamente sobre uma ferramenta e cinco formulários mais genéricos.

Os autores explicaram o motivo da escolha dos usuários por ferramentas semelhantes. A ferramenta Soboleo teve a maior quantidade de usuários. Observaram que a Soboleo e BibSonomy focavam em anotação de recursos, como páginas Web e entradas de bibliografias, enquanto as outras ferramentas focavam em construção de dados e ontologias estruturadas. Então, fazia sentido o fato de que aqueles usuários que tentavam a Soboleo também o fizessem com a Bibsonomy; por sua vez, aqueles usuários que tentavam a Collaborative Protege também fizessem com a Ontowiki. Aqueles usuários que usaram mais de duas ferramentas, as duas ferramentas mais usadas eram do mesmo tipo.

Noy *et. al.* (2008) argumentam sobre o fenômeno da ‘vantagem cumulativa’, quando os usuários, em grande escala, seguiram os padrões de representação que os organizadores do evento estabeleceram, quando inseriram os dados iniciais: tal condição, apesar de provocar tendenciosidades no experimento, era necessária. Caso os usuários tivessem de inserir dados a partir do zero, haveria o risco de não conseguirem criar qualquer conteúdo.

Os autores também observaram que ferramentas que devem ser baixadas e instaladas de servidores oferecem um trabalho adicional para o usuário, reduzindo sua preferência. Ferramentas desktop teriam a preferência sobre ferramentas Web somente em casos de extrema necessidade.

Quanto ao uso de ferramentas de discussão, Noy *et. al.* (2008) verificaram pouco uso, mesmo em ferramentas que ofereciam essa possibilidade. Uma das hipóteses para a falta de discussão entre os usuários era de que o tema sobre o qual se deveria gerar conteúdo (um portal para pesquisa científica) era extremamente restritivo. Outro

possível motivo foi a falta de relevância do tema trabalhado para os usuários, já que os resultados foram gerados somente para cumprir os objetivos do evento.

Noy *et. al.* (2008) concluíram que não há uma definição clara sobre o que é o conceito de ‘desenvolvimento colaborativo’. As definições variaram desde a distribuição de acessos de usuários para a mesma ontologia ou base de conhecimento, suporte de anotações de diferentes usuários, meios para descobrir a origem e o responsável por uma operação executada (*provenance*), meios de conseguir consenso sobre um assunto, entre outros.

### **3.4.7 Identificação das necessidades**

Após a experiência comparativa descrita no item anterior, Noy *et. al.* (2008) identificaram as necessidades para o desenvolvimento de ferramentas colaborativas para construção de conhecimento. As necessidades anotadas são:

1. Fluxos de trabalho e cenários colaborativos: os autores exemplificaram uma diferença entre a construção de uma ontologia específica para um projeto por um grupo fechado de usuários e a construção de uma taxonomia por uma comunidade aberta. Os usuários que participaram da experiência demonstraram preferências diferentes quanto às ferramentas de construção de ontologias ou quanto às ferramentas que adicionavam estruturas aos elementos de uma rede social. Todos chegaram a um acordo de que tais ferramentas necessitavam de funcionalidades que permitissem a conversão de uma estrutura em outra (a interoperabilidade).

De acordo com os usuários, foram levantadas algumas necessidades de políticas de mudança, desde as políticas mais organizadas, em que alguns usuários sugerem mudanças e outros praticam a discussão e o voto para a seleção dessas mudanças; um terceiro grupo realmente efetiva tais mudanças, até as políticas mais liberais, onde qualquer participante do processo colaborativo de construção pode efetivar suas mudanças imediatamente.

Apesar de que nenhuma das ferramentas avaliadas suportasse especificamente uma política de construção de consenso, muitas delas disponibilizavam meios técnicos que suportassem tal processo, tais como avaliação, votação e discussão.



2. Poder de Expressão: os autores compararam as necessidades dos usuários de uma ferramenta de construção do conhecimento com as funcionalidades que as ferramentas oferecem e buscaram um equilíbrio entre os dois. Uma comunidade que desenvolve taxonomias a partir de conexões entre recursos da internet será menos exigente do que um time treinado na construção de uma ontologia específica. Nenhuma das ferramentas oferecia recursos para o uso de uma ontologia como a OWL. Apesar disso, os usuários participantes do evento julgaram que a edição colaborativa de ontologias com uma capacidade de expressão mais complexa seria muito difícil.

3. Credibilidade de usuários e possibilidade de rastrear responsáveis e razões de uma mudança: Noy *et. al.* (2008) sugerem uma abordagem para medir a credibilidade dos participantes em um ambiente colaborativo de construção de conhecimento. Outros usuários podem avaliar as mudanças feitas por um dado participante. Foi levantada também a necessidade crítica de rastreamento das mudanças feitas na rede de conhecimento, para fins de diagnóstico: saber quem e quando se fez determinada mudança, ler comentários do autor da mudança, entender o estado da base de conhecimento quando a mudança foi feita, acessar conceitos e históricos de modificações.

4. Compreensão da Ontologia: dado que uma ontologia é construída por vários participantes, haverá vários casos em que um participante deve alterar elementos dessa ontologia que foram feitos por outros participantes. Devido a essa necessidade, é primordial que os usuários possam entender como os componentes de uma ontologia são construídos. Para entender esses elementos, há a necessidade de artifício de ‘visualização da ontologia’ construída colaborativamente. Os autores apontam que ainda não há uma solução bem sucedida de visualização das ontologias produzidas. Apesar disso, ferramentas que apresentam recursos visuais mais sofisticados foram preferidas pelos usuários.

5. Design de interface para o usuário: a interface dos aplicativos de construção colaborativa de redes de conhecimento é crítica. Vários recursos oferecidos pelo sistema, que são difíceis de achar, são inúteis. Outro ponto é que as ferramentas web vêm se tornando mais importantes, considerando tecnologias como *Asynchronous JavaScript Technology and XML (AJAX)*, que suportam uma ampla variedade de recursos para interface com o usuário. Os usuários participantes do experimento sugeriram a construção de *plug-ins* para inserir e acessar o conhecimento a partir de algumas ferramentas.

6. Espaços compartilhados e espaços pessoais: os autores apontam este tópico como controverso entre os participantes, com três variantes: a) Espaços pessoais para a construção da própria estrutura e dados; b) Compartilhamento imediato da estrutura e dados com os demais usuários, c) O caso em que o usuário pode desenvolver sua própria estrutura, de forma assíncrona, e compartilhá-la posteriormente com os demais usuários. Nenhuma das ferramentas implementa uma visão personalizada dos dados. Os autores presumem que os usuários freqüentemente necessitem restringir sua visão dos dados para confirmar suas decisões pessoais ou a sua rede de confiança.

7. Artifícios de construção de consenso: Noy *et. al.* (2008) descobriram a partir de depoimento dos usuários que os artifícios de avaliação de conceitos e participantes não funcionam se não há como justificar estas avaliações. Alguns usuários sugeriram o uso de ferramentas convencionais de conversação (*chat*), tais como MSN Messenger e Yahoo, ao invés de *chats* implementados nas próprias ferramentas. Os usuários concordavam que os recursos de discussão e voto para avaliação tornariam o processo de construção muito burocrático. Os autores também comentam que ainda tem dúvida sobre ser possível a construção colaborativa de ontologias, já que estes observaram que para alguns usuários, o espaço de trabalho é muito complexo, coletando alguns comentários, inclusive. Finalmente, os autores apontaram a necessidade de os ambientes colaborativos disponibilizarem registros de mudanças e comentários, gravando as discussões e conectando estes registros e discussões aos componentes das ontologias, desta forma ajudando na contextualização da discussão.

Finalmente, como conclusão, os autores comentam sobre as sugestões dos usuários para a inclusão de funcionalidades de uma ferramenta em outra. Vários usuários sugeriram adicionar os recursos de anotação de recursos web das ferramentas Bibsonomy e Soboleo às ferramentas de edição de ontologias, como Collaborative protege e Hozo. Os desenvolvedores da Bibsonomy e Soboleo atualmente estão integrando essas duas ferramentas para complementar a abordagem baseada em rótulos da Bibsonomy com a abordagem baseada em ontologias da Bibsonomy.

### **3.4.8 Requisitos levantados para ferramentas de construção colaborativa de ontologias**

Tudorache *et. al.* (2008) desenvolveram um levantamento de requisitos para o projeto de ferramentas que suportem a construção de ontologias de forma colaborativa. O levantamento foi baseado em entrevistas com usuários institucionais do Protege. Os autores estendem significativamente o conjunto de requisições descobertos no evento do CKC. Esse levantamento tem seu foco nas requisições de desenvolvedores de ontologias para domínios, como biomedicina. Os usuários são mais especialistas de domínio do que engenheiros de conhecimento. Eles explicam que os usuários do aplicativo Protégé já estavam acostumados a trabalhar no modelo cliente-servidor, vindo imediatamente as mudanças feitas por outros usuários. Por isso, focaram nos recursos que exclusivamente suportem a colaboração.

O motivo para o levantamento dos requisitos para colaboração era descobrir todo o raciocínio envolvido na construção das ontologias. Raciocínio este que pode ser capturado de dentro de ferramentas de comunicação entre os usuários do aplicativo, como mensagens de e-mail, postagens em fóruns de discussão, conversações por telefone e bases de dados, já que os autores identificaram a necessidade de existir um espaço onde fosse justificado o processo racional da construção de cada elemento da ontologia. Por outro lado, quando os desenvolvedores liam um e-mail ou comentário em fórum discutindo um módulo da ontologia, eles não conseguiam ver o contexto da discussão.

Os requisitos encontrados pelos autores para os módulos de colaboração para ferramentas de construção de ontologias foram os seguintes:

1. Integração de discussões e anotações no desenvolvimento de ontologias: ontologias exigem acordos, e é comum haver desacordos sobre modelagem dos conceitos entre os especialistas. Há a necessidade de ferramentas que permitam discussões e que registrem estas discussões, e estes registros devem estar associados aos artefatos produzidos.

2. Suporte para vários níveis de expressividade: os autores comentam a existência de restrições complexas, com intersecções, nas definições de classes para um de seus projetos de trabalho. As versões colaborativas de ontologias devem possuir o mesmo poder expressivo que o aplicativo referência (o Protégé) tem.

3. Gestão de usuário e garantia de informação: dado que existem vários usuários contribuindo para a construção da ontologia e com sua correspondente discussão, os usuários desejam saber de onde uma dada informação vem.

4. Escalabilidade, confiabilidade e robustez: são os critérios para o uso de ferramentas em sistemas de produção. Escalabilidade tanto para o número de usuários quanto para o tamanho da ontologia, confiabilidade para os especialistas do domínio, que não podem se arriscar a perder seus dados. Quanto à robustez, os autores comparam o conceito de robustez de aplicações de construção colaborativa de ontologias com outras aplicações que os especialistas de domínio já utilizam, mas não a definem. Tudorache *et. al.* (2008) falam sobre o problema da incredulidade inicial de um especialista de domínio em usar ferramentas novas, já que a construção de ontologias não é a sua tarefa prioritária.

5. Controle de acesso: os autores citam alguns recursos de controle de edição de uma ontologia: alguns sistemas permitem a alteração de uma permissão para a ontologia completa, mas não para apenas uma parte dela, o que foi apontado como uma necessidade dos usuários. Os autores declaram que há a necessidade de refinamento de modelos para controle de acesso.

6. Suporte ao fluxo de trabalho: os autores citam os casos do projeto NeOn e da metodologia Diligent, que possuem suporte a controle de tarefas para mudança na ontologia produzida. A metodologia Diligent para desenvolvimento colaborativo foca na formalização do processo de argumentação. Uma especificação de fluxo de trabalho deve conter o processo de propor uma mudança e encontrar um senso comum e deve possibilitar a inclusão de papéis para diferentes usuários. Os modelos de fluxo de trabalho ainda estão sendo pesquisados.

7. Acesso síncrono e assíncrono para ontologias compartilhadas: o acesso síncrono ou assíncrono depende da escolha do usuário. Este depende do tamanho do grupo de trabalho e da complexidade da ontologia. Os autores presenciaram casos em que tanto usuários desejavam ver as alterações feitas por colegas imediatamente quanto desejavam primeiro fazer seus testes lógicos, antes de atualizar a ontologia através de operações de *'check-in'*.

### 3.4.9 Editores 3D e Ferramentas de Visão 3D de Ontologias

Bosca *et. al.* (2006) apresentam o OntoSphere 3D, uma ferramenta de visão tridimensional para redes de ontologias, plugado no Sistema Protègè. Os autores fazem uma comparação com outras ferramentas de visão de ontologias, como a Jambalaya (JAMBALAYA, 2010), OWLViz (HORRIDGE, 2010), e TgViz (ALANI, 2003), com objetivo de obter processos mais efetivos de modelagem de ontologias.

Katifori *et. al.* (2007) mostram um levantamento de técnicas de visualização para ontologias. Primeiro define o que é uma ontologia, utilizando duplas compostas por Classes, Slots, Relações de Herança e Instâncias de Classes. Os autores levantam *frameworks* de categorização de visualização usando critérios como tipo de dado a ser apresentado na interface e topologia de tarefa; apresentam também critérios cognitivos tais como: atenção, abstração e compreensão imediata do uso (*affordance*) para o caso de visualizações 3D de ontologias; critérios de design como estilos de nós e links, planos de informação (*information landscapes*), desenho de grafos, incluindo pontos de vista a partir da Bioinformática.

Nagy *et. al.* (2010) apresentam uma *framework* sobre visões 3D de ontologias que permite aos usuários armazenar e ver o raciocínio por de trás dos mapeamentos entre ontologias, discutindo o nível de suporte cognitivo que pode ser alcançado, usando modelagem 3D de redes de ontologias. Os autores tentam modelar o processo de raciocínio humano para mapear conceitos entre ontologias, apresentando esse processo através de cinco etapas:

1. Seleção de termos candidatos;
2. Hipóteses para possíveis mapeamentos;
3. Procura por evidências para provar as hipóteses;
4. Eliminação de termos que não cumprem com a crença inicial;
5. Procura de consenso sobre os mapeamentos.

Sobre a visualização da ontologia, os autores estão conscientes sobre as variáveis visuais que podem ser controladas para facilitar a procura, a conexão e a compreensão dos mapeamentos entre termos, como: o número de termos, escala de toda a rede, amostragem parcial da rede, e descobriram que a visão 3D melhora a produtividade dos usuários.

<b>Objetos de Sistema de Arquivo</b>	<b>Documentos Categorizados</b>	<b>Ontologia</b>
Pasta	Categoria	Entidade (Classe ou Instância)
Relação pasta/subpasta	Relação categoria / subcategoria	Relação “É um”
Visão em árvore	Categorização	Taxonomia
Arquivo	Documento	Instância
Propriedades de arquivo	Propriedades do documento	<i>Slots</i>

Quadro 5: Equivalência entre documentos categorizados de arquivo e recursos de ontologia.

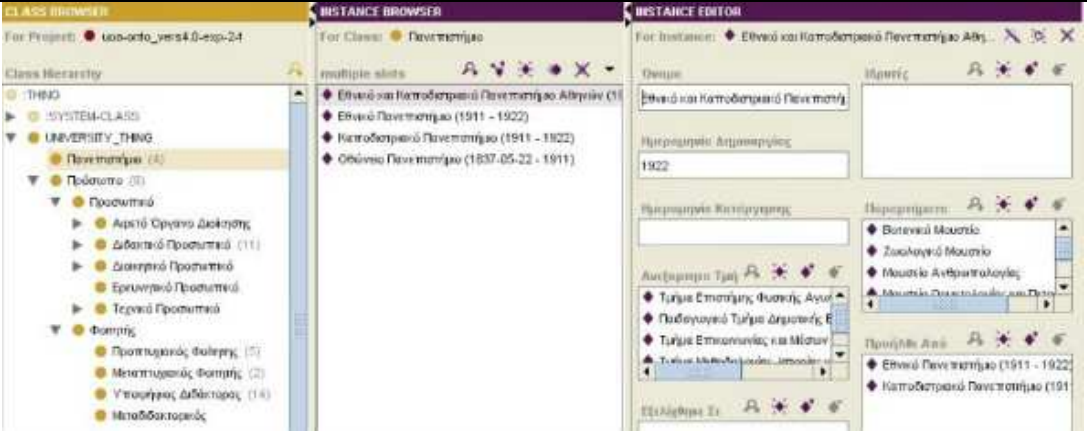
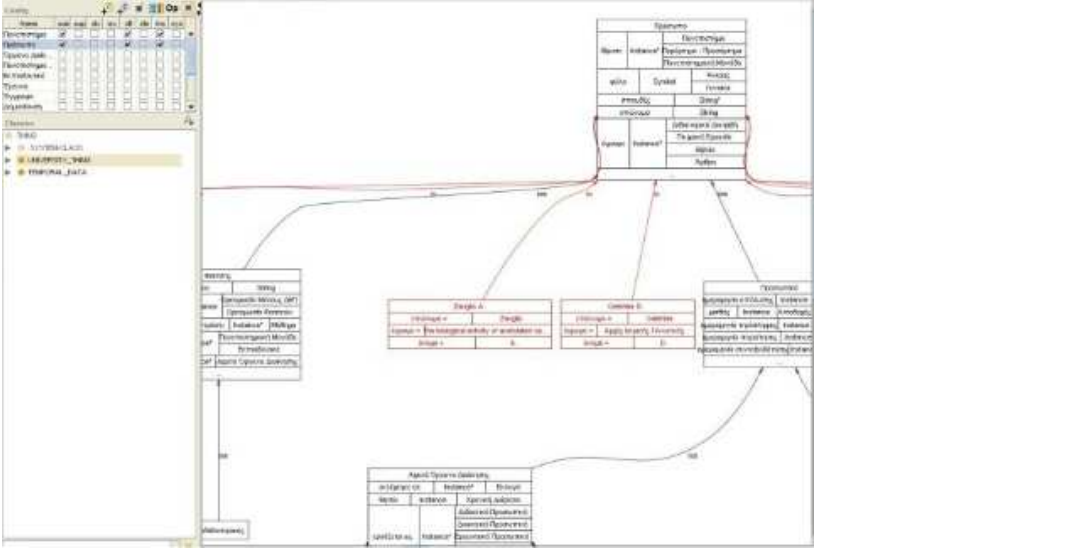
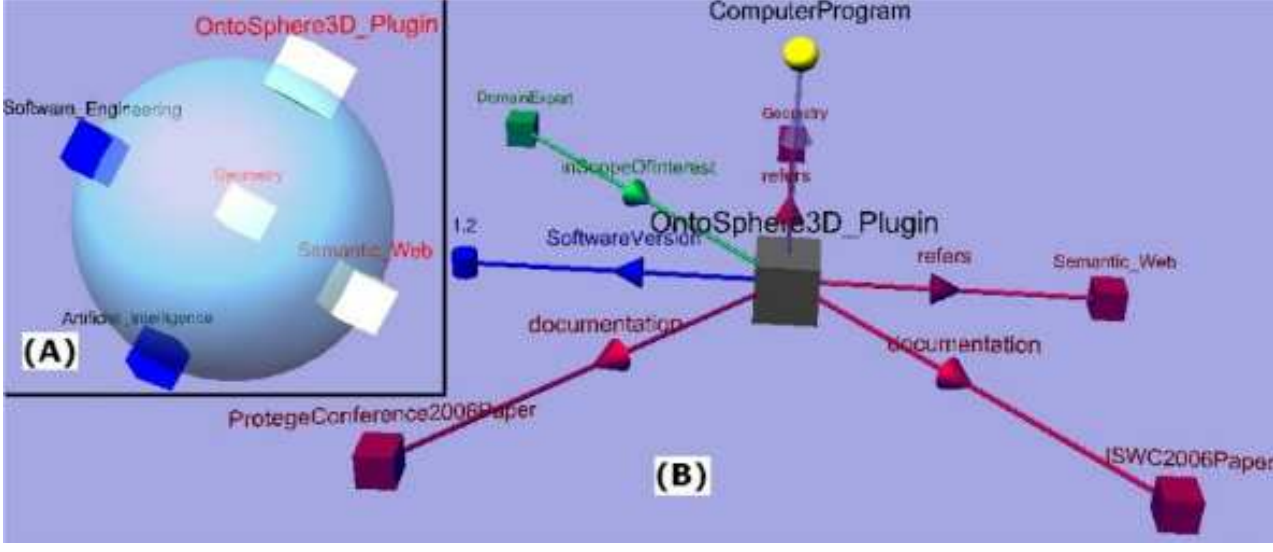
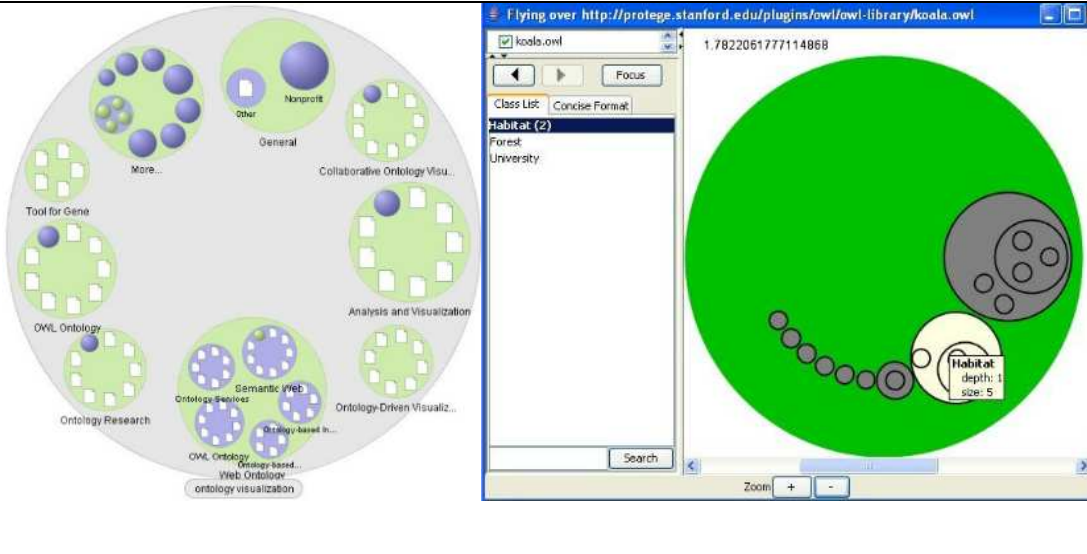
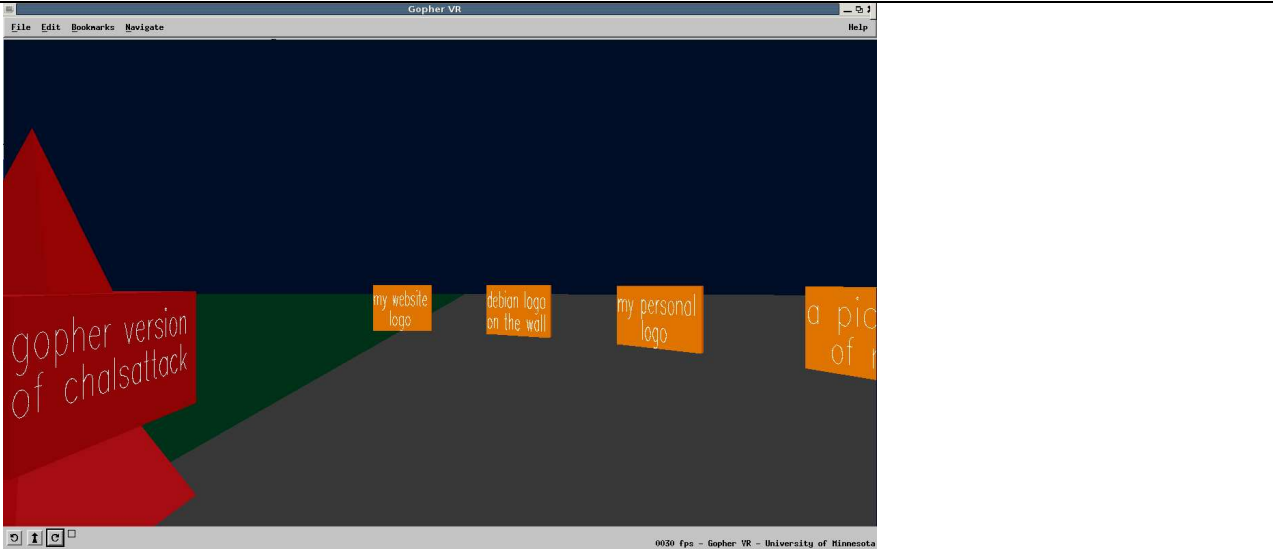
Os autores agruparam as técnicas de visualização (ver Quadro 6, página 78), sem esgotarem todas as opções, com a intenção de gerar novas idéias sobre a disposição e visualização dos elementos de instâncias de ontologias no espaço. Os métodos foram subdivididos em três categorias:

1. Visualização 2D: usa o plano e nenhuma noção de profundidade;
2. Visualização 3D: usa a profundidade e simula os objetos como na realidade, procurando melhorar a usabilidade;
3. Visualização 2 ½ D: é uma visão bidimensional que utiliza um eixo de profundidade que é uma combinação dos eixos de largura e altura, e que não permite a manipulação. Ex: a rotação ou movimentação - neste terceiro eixo.

Para esta pesquisa, foram consideradas ferramentas construídas até 2006.

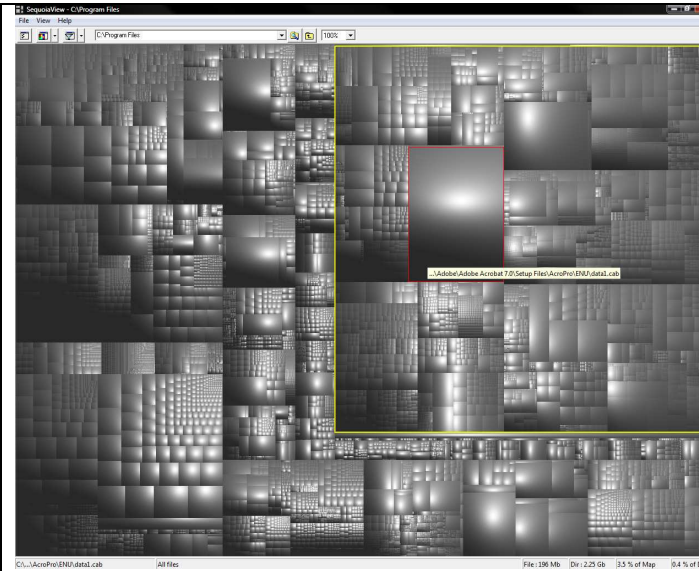
Considerando essas seis categorias, o pré-requisito para a formatação da apresentação de uma ontologia é que todas as ontologias, quando comparadas, devem apresentar elementos comuns: Classes, Relações, Instâncias e Propriedades (também chamadas de Slots).

Entretanto, Nagy et. al. (2010) dizem que os tipos de disposição dos dados apresentam vantagens e desvantagens. Para tal, é necessário observar que tipos de tarefas um usuário deseja desenvolver. Outro ponto considerado pelos autores é que, até 2007, as máquinas comuns não possuíam processamento suficiente para garantir uma visualização contínua e em tempo real para visualização de dados em um espaço 3D.

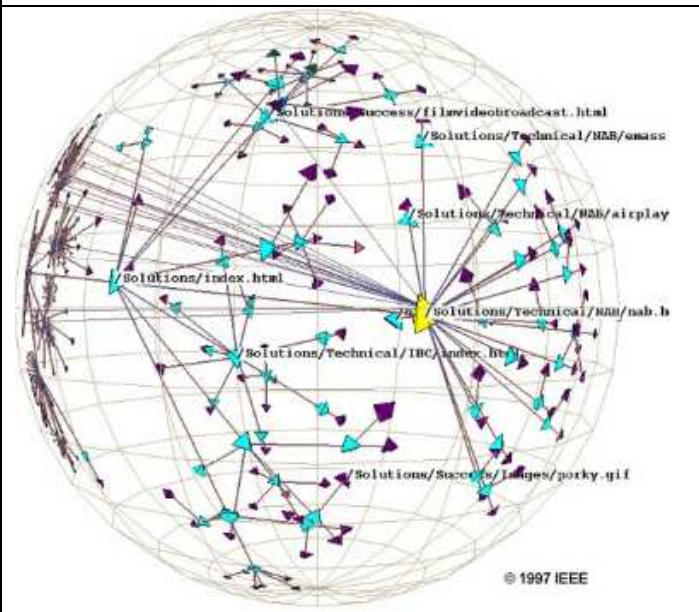
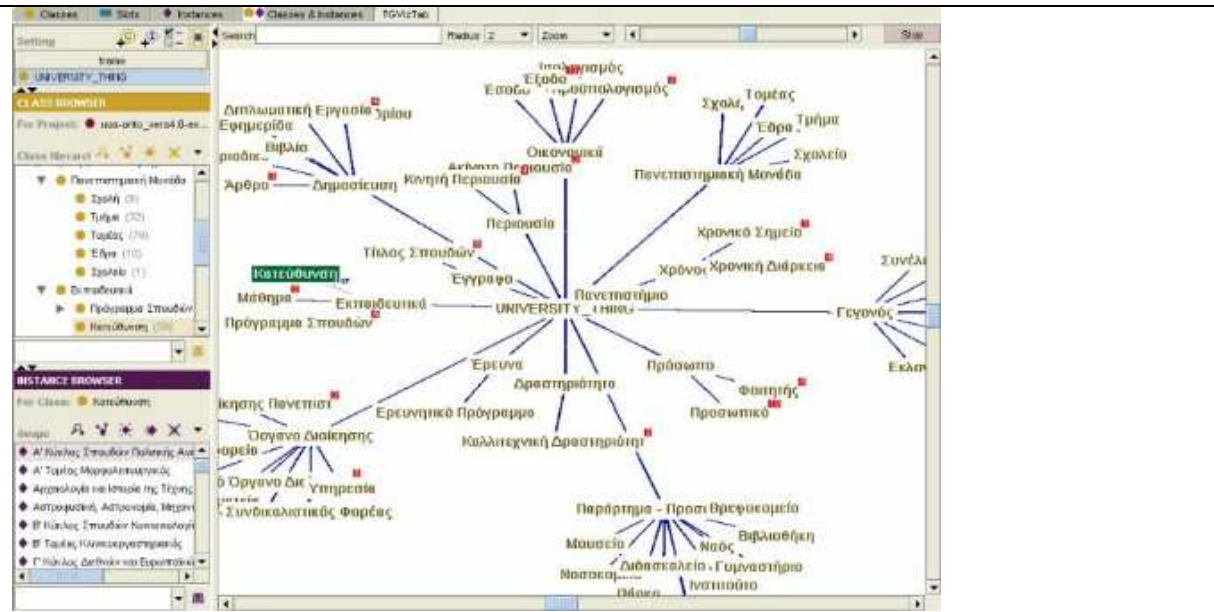
<p><b>Tipo de Visualização</b></p>	<p><b>2D</b></p>	<p><b>3D</b></p>
<p><b>Lista Indentada</b></p>	 <p>The screenshot shows the Protege 2D interface. On the left is the 'Class Hierarchy' tree, showing a hierarchy from 'THING' down to 'UniversityThing' and 'University'. The middle pane is the 'Instance Browser' for the class 'University', listing instances like 'Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών (1911 - 1922)'. The right pane is the 'Instance Editor' for a selected instance, showing fields for name, date, and other properties.</p>	
<p><b>Άρvore Link-Νό</b></p>	 <p>The screenshot shows a complex link-node tree diagram. The left pane shows a class hierarchy. The main area displays a network of nodes and edges, representing relationships between different classes and instances. Some nodes are highlighted in red, and the edges are colored lines connecting them.</p>	 <p>The screenshot shows a 3D visualization of an ontology graph. It features a central blue sphere labeled 'OntoSphere3D_Plugin' with various nodes and edges extending from it. Nodes include 'Software_Engineering', 'Security', 'Semantic Web', 'Artificial Intelligence', 'ProtegeConference2006Paper', 'OntoSphere3D_Plugin', 'ComputerProgram', 'DemandExpert', 'inScopeOfInterest', 'SoftwareVersion', 'documentation', 'Semantic Web', and 'ISWC2006Paper'. Edges are labeled with terms like 'refers', 'documentation', and '1.2'.</p>
<p><b>Passível de Zoom (Zoomable)</b></p>	 <p>The screenshot shows a zoomable ontology visualization. On the left is a circular menu with icons for 'General', 'Collaborative Ontology Visu...', 'Tool for Gene', 'OWL Ontology', 'Ontology Research', 'Semantic Web Ontology Explorer', 'Strategy-based...', 'Analysis and Visualization', and 'Ontology-Driven Visualiz...'. The main area shows a large green circle with a smaller grey circle inside it, representing a zoomed-in view of an ontology. A 'Class List' window is open, showing 'Habitat (2)', 'Forest', and 'University'.</p>	 <p>The screenshot shows a 3D visualization of a website in a virtual environment. The scene is dark with a red and green floor. There are several orange rectangular signs on the floor, each containing text: 'gopher version of chalsattack', 'my website logo', 'debian logo on the wall', 'my personal logo', and 'a pic of'. The interface includes a menu bar at the top with 'File', 'Edit', 'Bookmarks', 'Navigate', and 'Help'.</p>



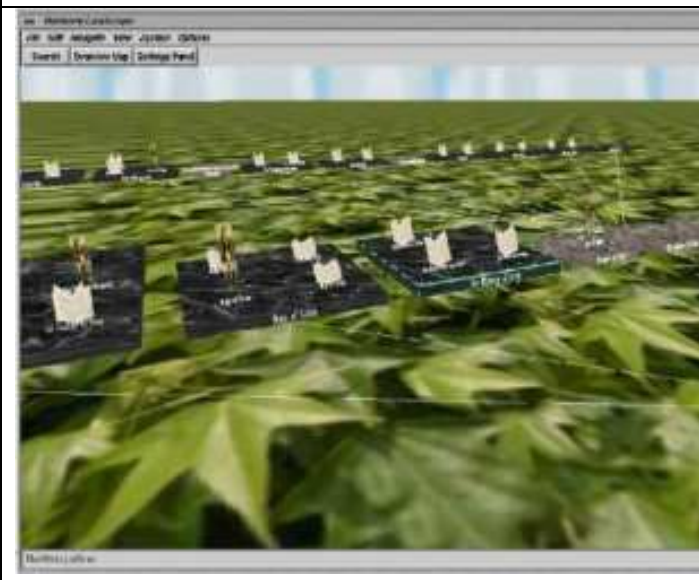
Preenchimento de Espaço



Foco Contexto mais ou menos



Pianos de Informação



Quadro 6: Tipos de visualização, separadas em 2D e 3D.



Katifori et. al. (2007) argumentam que o princípio da visão humana baseia-se em projeções 3D do mundo real, conduzindo o raciocínio de que as visões mais próximas da tridimensional seriam as mais efetivas. Porém, os autores reconhecem que as visões 3D de dados não oferecem resultados da forma como se espera. Contudo, Bosca et. al. (2005) afirmam que o mapeamento de vários recursos de uma ontologia (como hierarquias de classes, relações de papéis, propriedades e as instâncias em duas dimensões) pode ser restritivo, enquanto a apresentação 3D oferece a possibilidade de uma apresentação mais rica. Para Smallman et. al. (2001), a visualização 3D tem uma forte preferência pelos usuários.

Segundo Katifori et. al. (2007), apesar dessa discussão, não está claro se a visão 3D é mais preferida do que a visão 2D. Exceção do sistema OntoSphere, citado pelos autores, a visualização 3D ainda não foi aplicada de forma extensiva para o domínio das ontologias; como consequência, não há resultados conclusivos sobre a sua eficácia. Nesse sentido, de acordo com Smallman et. al. (2001), estão surgindo diversas literaturas sobre as vantagens e desvantagens da visualização 2D e 3D. Hicks et al. (2003) observaram que a visão 2D parece apresentar uma melhor performance. Plaisant et. al. (2002) diz que as representações 3D melhoram, de maneira marginal, o uso do espaço, enquanto aumentam a complexidade da interação. Cockburn e McKenzie (2002) mostraram que a navegação em um espaço 3D pode ser difícil para um usuário novato e que até mesmo uma tarefa simples, tal como a seleção de um objeto, pode ser problemática. Kobsa (2004) sugere que o resultado negativo de visualizações 3D é, em alguns casos, o resultado da falta de outros recursos, como: ferramentas de procura, de destaque visual de resultados de procura (*highlighting*), de filtragem e de navegação.

A comparação de visões de ontologias em espaço 2D e 3D, feita por Katifori et. al. (2007) (Figura 24, Figura 25), resultaram em uma conclusão de que não há uma disposição que seja a melhor. Wiss et al. (1998) e Golemati et al. (2006) propõem uma solução de visão dos dados na qual o usuário escolheria a visualização da instância de ontologia que mais lhe conviesse, como é o caso do sistema Protege, com seus vários plug-ins. Os autores também concluíram que as visões 3D de grandes instâncias de ontologias devem ser auxiliadas com ferramentas de procura ou artifícios de *query*. Essa conclusão considerou que os usuários preferem uma navegação ordenada e clara sobre os dados, a utilização das vantagens oferecidas pelo contexto semântico da informação e do perfil do usuário, de maneira a guiar e suportar a hierarquia ou exploração da ontologia.

Table VIII. Ontology-Related Tasks versus Visualization Methods. ++ Shows That the Method Supports this Task Very Effectively, + That it Supports It but Not Very Effectively,—That It Does Not and—That it Does Not Support it and in Some Cases it May Hinder it. Comments Are Provided When Appropriate									
		Indented List	Node-Link and Tree	Zoomable	Space-Filling	Focus + Context and Distortion	3D Information Landscapes	Time-Related Visualizations	
Overview	Hierarchy Overview (is a hierarchy along with multiple inheritance)	+	++	--	-	-	++		
	View depth of the hierarchy	+ The user has to expand all nodes to find out.	++	--	-	-	++		
	Identify areas with many/few classes/instances	-	+	--	++	+	++		
	Overview of instances related to some property	--	+	-	++	+	++		
	Overview of role relations	--	+	+(Jambalaya) -- (Others)	--	-	+(Harmony) --- (fsn)		
	Number of instances per class	This task is partly tool-dependent. A number should be available, otherwise the user has to count.							
	View total number of classes	This task is partly tool-dependent. A number should be available, otherwise the user has to count.							
	View total number of instances	This task is partly tool-dependent. A number should be available, otherwise the user has to count.							
	Quick Browsing	++	++	+	++	++	+		
Zoom	Find Class or Instance by name or other property.	This task is partly tool-dependent. A search tool is necessary in this case, especially for big ontologies. If search is not available, then methods that better support this task are those supporting quick browsing, in order to examine all ontology nodes quickly.							
	View sub-hierarchy (retaining context)	+	++	-	+	+	++		
	View path to instance or class	++	++	--	-	+	++		

(Continues)

Figura 24:Comparativo entre tarefas de ontologias e métodos de visão. Fonte: KATIFORI et al. (2007).



## 3.5 MÁQUINAS DE BUSCA

### 3.5.1 Clusty

Valdes-Perez *et al.* (2008a, 2008b) e Watzman *et al.* (2008) apresentaram a página Clusty. Valdes-Perez *et al.* (2008b) dizem que muitas empresas de aplicativos de gestão de informações já estão anunciando a clusterização de resultados como a tecnologia que faz parte de seus produtos. Definiram também o que é o processo de clusterização: o agrupamento de resultados por tópicos e sub-tópicos.

Valdes-Perez *et al.* (2008b) descrevem três vantagens de clusterização dos resultados de procura: 1) Visão ampla sobre os principais temas retornados; 2) Rápido alcance do documento através da navegação pelos temas e sub-temas; 3) Agrupamento de assuntos relacionados

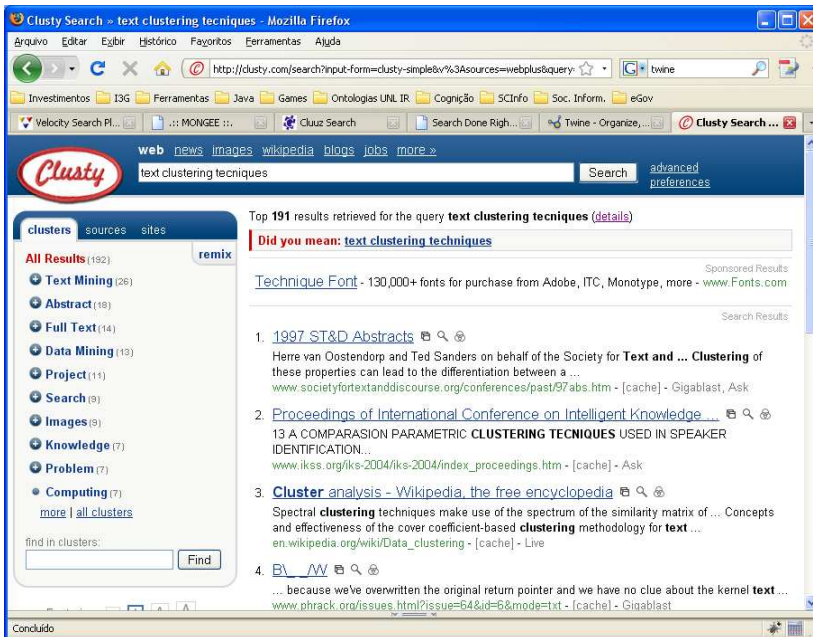


Figura 26: Página do Clusty.

### 3.5.2 Cluuz

A Cluuz é uma ferramenta de procura que mostra, além dos links para as páginas resultantes, as entidades e imagens que são extraídas dessas páginas resultantes. É como se o sistema rastreasse os documentos para ver dentro deles, o que é importante para o usuário (CLUUZ, 2008b).

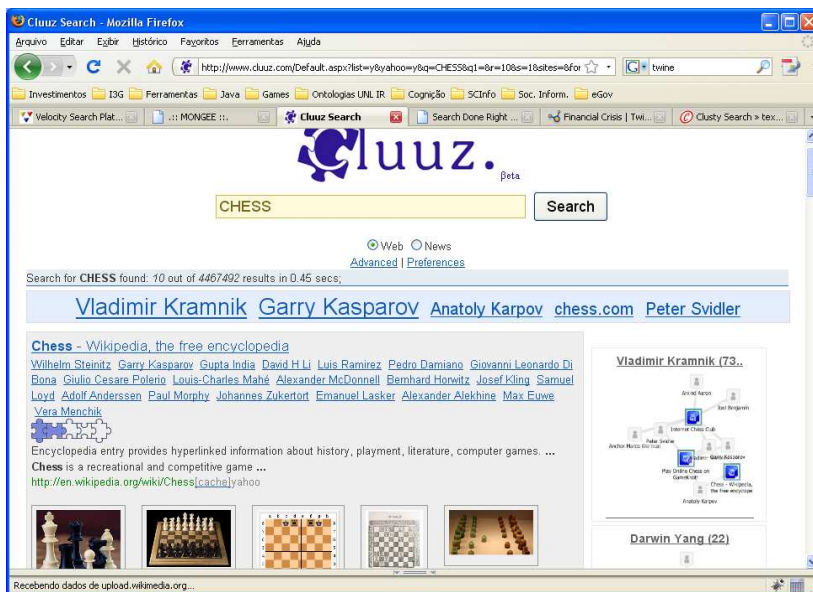


Figura 27: Página do Cluuz.

O sistema também pode mostrar redes de relacionamentos para os *clusters* de entidades que foram encontrados nos resultados do texto procurado do usuário. (Figura 28)

E também seções de links, classificados por temas principais, imagens e *tag clouds*. Para o caso das *tag clouds*, a procura será refinada com a inclusão da *tag* selecionada junto à procura inicial.



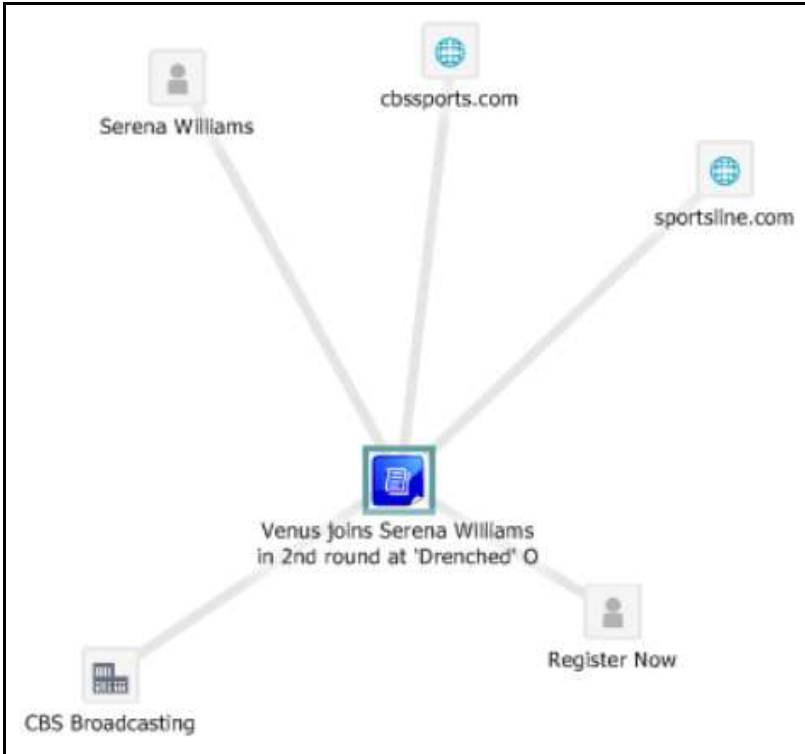


Figura 28: Redes de relacionamentos de resultados em clusters, na página da máquina Cluuz.

### 3.6 BIBLIOTECAS DE VISÃO DE INFORMAÇÃO

#### 3.6.1 Alphaworks

Wattenberg (2005) descreve três possíveis direções a serem tomadas pelos projetistas de ferramentas de visão de informações: 1) Visão de texto e de conversações em texto: e-mails, listas de discussão e wikis; 2) Fluxo histórico: A história de objetos complexos e estruturas tais como organizações corporativas e bibliotecas de software; 3) Exploração de dados sociais: Segundo o autor, uma visão destes dados se converte em um acelerador dos processos de mineração, análise e conversação destes objetos.

### 3.6.2 Framework RR

Para criar uma solução de visão de termos e relacionamentos contidos em bases de dados de ontologias e de visão de dados de telefonia e transações financeiras em redes criminais, Stradiotto et. al. (2004, 2006a, 2006b, 2006c) criaram uma framework para a implementação de soluções para problemas que necessitem de visão de dados na forma de redes de relacionamentos.

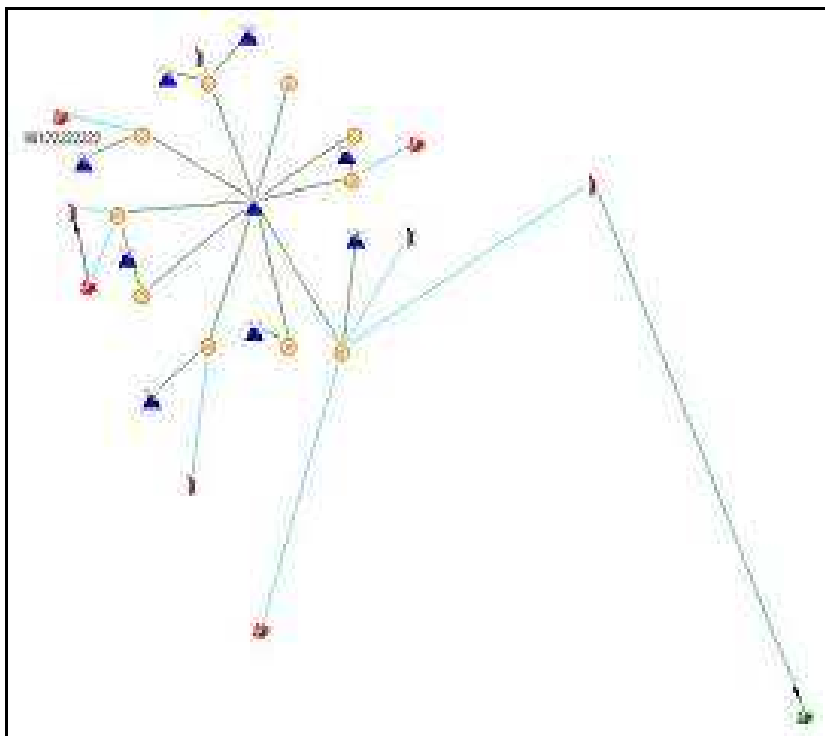


Figura 29: Interface de aplicativo de visão de dados de telefonia e transações financeiras em redes criminais.

### 3.6.3 Prefuse

A biblioteca *Prefuse*, segundo Heer *et. al.* (2005), é um conjunto de classes programadas em Java para permitir a construção de ferramentas de visão de informação.

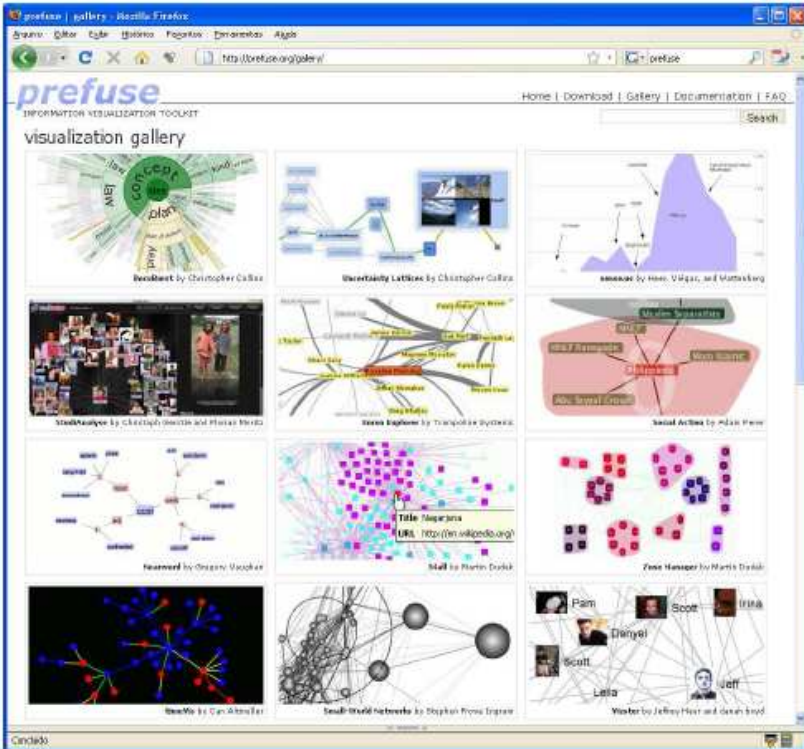


Figura 30: Página Web da Prefuse Toolkit, uma biblioteca para criação de soluções de visão de informação.

### 3.6.4 Java 3D

Java 3D (JAVA3D, 2010) (Figura 31) é uma interface de desenvolvimento de aplicativos (API) que funciona sobre a plataforma Java. Uma API utiliza as camadas gráficas da OpenGL (para aplicações científicas de alta precisão com Gráficos 3D) e DirectX (Biblioteca da Microsoft para jogos, contendo recursos para Gráficos 3D, Som e Entradas).

Dmitrieva e Verbeek (2008, 2009-1, 2009-2) apresentam um visor de ontologias baseado na API Java 3D (ONTVIS, 2010) (Figura



31), utilizando da tecnologia Java Web Start, de tal forma que ontologias no formato OWL, disponíveis na internet ou em arquivos, possam ser vistas em um espaço tridimensional. A ferramenta apresenta algumas formas de representação de ontologias com uso de grafos no espaço tridimensional: Esfera e Geometria Hiperbólica com duas variantes: Modelo de Klein (Figura 32) e o Modelo de Disco de Poincaré (Figura 33).

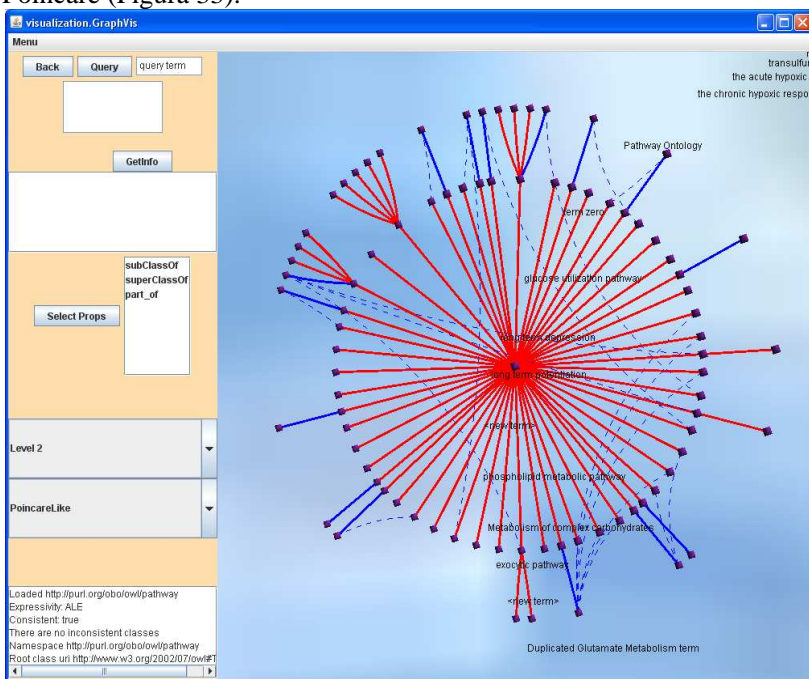


Figura 31: Visor de ontologias baseado na API Java 3D. Fonte: ONTVIS (2010).

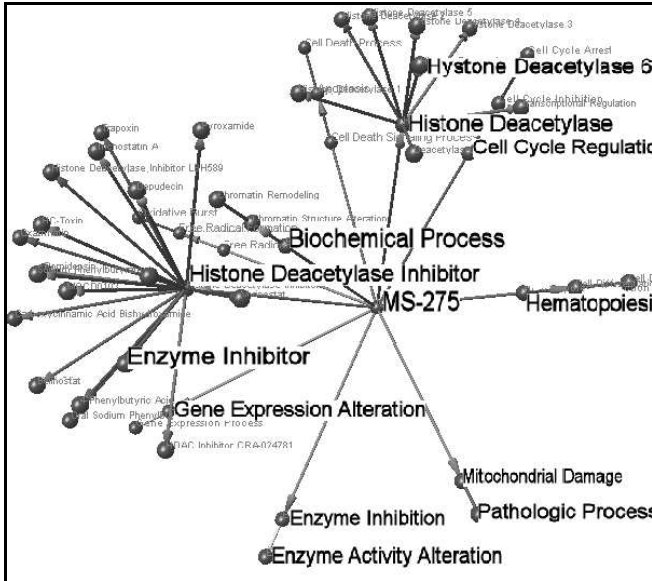


Figura 32:Modelo de Klein.

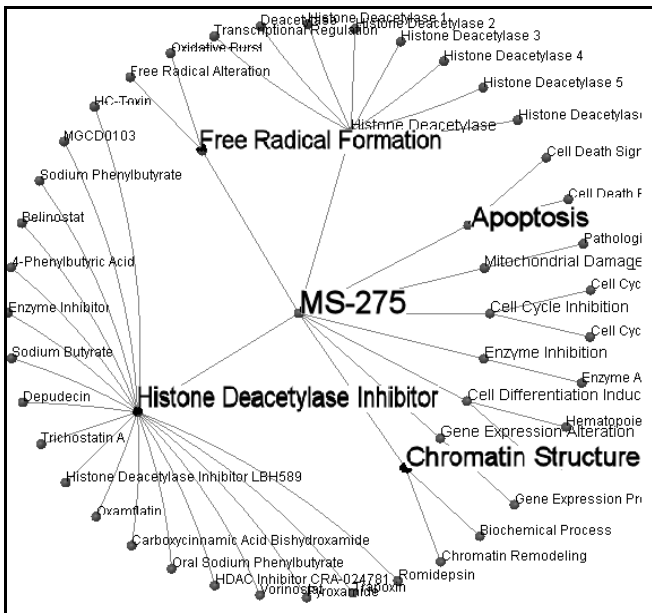


Figura 33:Modelo de Disco de Poincaré.

### 3.7 CRITÉRIOS DE USABILIDADE DE SOFTWARE

Para que seja possível comparar dois aplicativos de edição de ontologias (nesta tese, são os editores do *Ontojuris* e o proposto editor com visão de redes de relacionamento), é necessário que sejam enumerados critérios e suas correspondentes escalas.

Bresciani *et. al.* (2008) descreve os critérios de usabilidade de software (Fatores de Design), suas justificativas, as contrapartidas destes (caso em que um bom valor em um critério pode resultar em um baixo valor para outro critério), e comenta casos em que baixos valores nesses critérios possam trazer benefícios no planejamento e implementação de aplicativos. Os critérios ou ‘fatores de design’ sugeridos pelo autor são apresentados nos Quadros Quadro 7, Quadro 8 e Quadro 9.

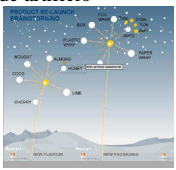
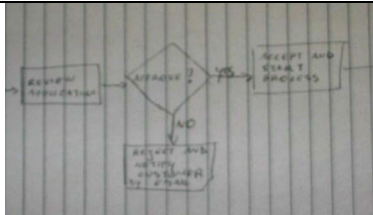
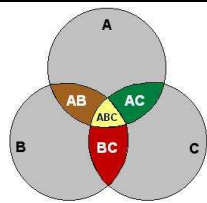
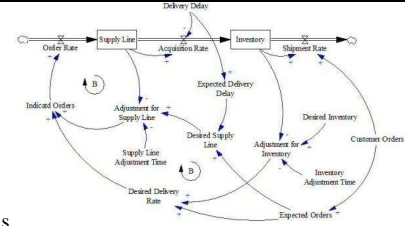



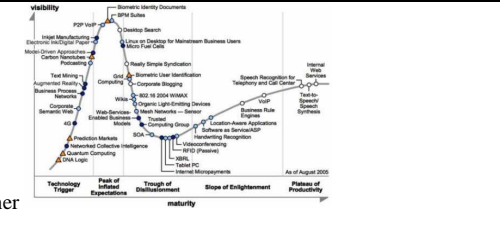

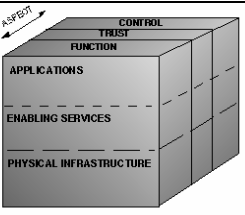

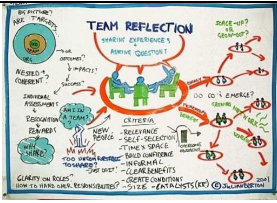


Bresciani *et. al.* (2009) isolaram os fatores que auxiliam a transformar os métodos de visualização em gestão de informações, e que também auxiliam a identificar as visualizações interativas como catalisadores para o compartilhamento do conhecimento e a criação de idéias. O Quadro 10 mostra uma listagem destes fatores.

Título do Critério	Questão Orientadora	Descrição (Explanation)	Dimensões Similares (Similar dimensions in literature)	Citações de praticantes e pesquisadores (Quotes from practitioners and researchers)	Implicações de design (Design implications)	Contrapartidas (Tradeoffs)	Escala (Scale)	Exemplo de baixo valor (Example Low)	Exemplo de alto valor (Example High)
1. Impacto Visual									
2. Clareza									
3. Trabalho Final Percebido									
4. Foco									
Direcionamento									
5. Facilitação de Idéias									
6.									
Modificabilidade									
7. Suporte à Interação de Grupo									

Quadro 7: Visão completa da tabela de Fatores de Design. Fonte: Bresciani et. al. (2008).

Título do Critério	Questão Orientadora	Descrição ( <i>Explanation</i> )	Dimensões Similares ( <i>Similar dimensions in literature</i> )	Citações de praticantes e pesquisadores ( <i>Quotes from practitioners and researchers</i> )	Implicações de design ( <i>Design implications</i> )
1. Impacto Visual	Quão atrativa é a visualização?	Esta dimensão está relacionada com as características gráficas da visualização. O Impacto Visual é alto se a visualização é capturada imediatamente, se é passível de ser recordada, emocionalmente passível de ser recordada, ou se relembra um trabalho artístico. O Impacto Visual é baixo se a visualização é simples ou de baixa qualidade. Um Impacto Visual alto pode ser contra produtivo porque pode dispersar atenção. Uma visualização prazerosa é mais passível de atrair a atenção do observador e criar um efeito de aura nos demais aspectos da visualização, inspira criatividade e dá emoções, conseqüentemente aumentando seu suporte mnemônico.	Ímpeto Visual: “Quão atrativo e convidativo à ação e posterior exploração é uma visualização”; Conteúdo de Estória; Expressividade de papéis; Memorabilidade;	“Eu uso desenhos, é o meu estilo: É divertido e as pessoas gostam”;  “Quão prazeroso e atrativo o diagrama é para o usuário final e quão encorajador isto é para ser usado.”	Uma visualização que tem um baixo valor no impacto visual pode ser melhorada usando uma paleta de cores convencional restrita (i.e. baseada em <i>hue</i> ), ícones e símbolos. Um alto nível é dado por itens visuais distintos, desenhos não convencionais ou sofisticados que distribuem surpresa ou emoções: metáforas visuais, desenhos artísticos, olhos, faces, formas com cantos. Contornos enfatizados, simetrias e razões de aspecto prazerosas aos olhos ( <i>golden ratios</i> ) também evocam impressões de beleza.
2. Clareza	A visualização é facilmente compreendida com um baixo esforço cognitivo?	Clareza é alta se o significado da visualização pode ser compreendido em uma primeira vista. Ela é baixa se a visualização requer tempo e concentração de forma a ser compreendida. Baixa clareza deve ser necessária quando o conceito representado é muito complexo, e se uma alta clareza pode conduzir a uma super simplificação. Este fenômeno é relacionado com a familiaridade de audiência com as convenções, com a complexidade e completeza, e com a consistência dos elementos. <b>Dificuldade em compreender não é sempre negativa, já que a super simplificação da realidade pode conduzir à compreensão imprecisa, com conseqüentes erros em julgamentos.</b>	Imediatismo visual: “A primeira impressão; características que permitem ao observador perceber e reconhecer <i>de um princípio</i> ” Visibilidade: “Habilidade de ver componentes de maneira fácil”	“Se o diagrama é muito complicado, as pessoas simplesmente desistem da apresentação” “O que funciona depende do grupo e do propósito: Se o diagrama tem uma vida de cinco minutos ou se este tem de ser compartilhado e distribuído; se o alvo é somente as pessoas o criaram, ou um público maior”	Se diferentes elementos são mostrados de forma similar, e vice-versa, se os elementos são mostrados de forma diferente, a clareza fica comprometida. A possibilidade de facilmente fazer comparações (através de justaposição) melhora a clareza. Uma visualização familiar deve ser mais clara, mas no fundo esta visualização nem sempre suporta a descoberta de novas idéias. A ambigüidade pode ajudar no suporte a pensamentos divergentes, porque os elementos podem ser interpretados de maneira diferente.
3. Trabalho Final Percebido	A visualização convida a contribuições e modificações, ou ela lembra um produto finalizado e polido?	Esta dimensão é alta se a visualização se parece com uma ilustração final ou uma figura de um livro. Quando a percepção final é baixa, a visualização lembra um croqui ou um trabalho provisório, por exemplo, feito com notas de <i>post-it</i> ou com lápis. Este tipo de apresentação influencia fortemente a boa disposição do grupo a interagir, questionar e modificar o diagrama. A percepção de trabalho terminado deveria ser consistente com a atual possibilidade de modificação.	Provisionalidade: “Grau de compromisso com ações e marcos” Fluidez-Congelamento: “Materiais são observados para ser tratados como congelados, e até indisponíveis para mudança; e também fluidos, abertos e dinâmicos”	“Desenhando no quadro branco é temporário e então as pessoas assumem mais riscos. É melhor para prototipagem e exploração, então, quando você está satisfeito, você tira uma fotografia ou faz um desenho final em um <i>flip chart</i> ” “é importante esclarecer se a visualização é congelada ou não, determinando-a, sem ambigüidades, se é um esboço ou uma arte final”	Em um contínuo de meios que suportam a aparição “finalizada” de uma visualização, temos, em uma extremidade, lápis e papel, e em outra, apresentações digitais tais como <i>Power Points</i> , ou gráficos impressos relembando pedaços de publicidades de desktops.
4. Foco Direcionado	A visualização dirige a atenção para os itens principais da discussão?	A direção do foco é alta quando a atenção dos participantes é dirigida somente para os itens principais (relevantes) da discussão. Quando não há foco, ou o foco é disperso em muitos itens, o Foco Direcionado é baixo. Baixo Foco Direcionado pode ser útil quando são requisitados pensamentos divergentes. Um gráfico inteligente que foca em um, ou em poucos itens, pode ajudar o grupo a manter o centro da discussão. A saliência é um recurso fundamental a ser considerado, já que pode alternar a atenção do conteúdo para a forma gráfica. O foco pode ser dispersado de forma útil quando há a procura por pensamentos divergentes ou alternativas diferentes de necessidades a serem consideradas. Como conseqüência de um foco não convencional, novas descobertas podem surgir (conduzindo ao suporte de alta inferência).	Foco: “Prestar atenção no assunto” Saliência: “Qualquer coisa que um usuário final foca durante o processo de construção de uma visualização tende a se tornar o foco de discussões subseqüentes mediadas pela visualização” Notação secundária “Informação extra com outro significado além da sintaxe formal”	“É um problema quando você fica mais dedicado ao modelo visual do que à idéia”	Usando uma cor distintiva, ou uma fonte negrito, posição no centro, circulando um item (com um quadrado ou um círculo), naturalmente direciona a atenção para aquele item como o ponto central da discussão; colocando vários itens do mesmo tamanho e cor à mesma distância dispersa o foco. Em uma visualização o foco pode ser imposto (por exemplo quando impresso em papel), ou pode ser mudado para se ajustar à dinâmica da discussão (quando alguns itens podem interativamente ser iluminados, circulados ou coloridos).
5. Facilitação de Idéias	As novas idéias são geradas como resultado da forma de visualização?	Quando uma nova compreensão é gerada “livremente” usando visualizações ou mudando o foco (dê um passo atrás e veja a grande imagem), ou mudando as restrições de representação, então a Facilitação de Idéias é alta. Se a visualização não ajuda no ganho de novas idéias ou compreensões, ela tem uma baixa Facilitação de Idéias. Esta dimensão está relacionada com o potencial de uma visualização de ajudar a descobrir novos padrões ou relacionamentos. Ela descreve a habilidade da visualização em ajudar a pensar e a descobrir processos. Facilitar idéias é um diferenciador vital e um valor agregado na visualização sobre textos: Ela permite ganhar novas compreensões “de maneira livre” somente mudando o tipo de visualização, o foco, ou as restrições de representação.	“Modo particular em que uma restrição estrutural que governa representações é equivalente a uma restrição que governa os objetivos da representação” “Nova informação é gerada como um resultado de seguir as regras notacionais”. Eficiência Computacional: Através do melhoramento da percepção	“É importante encontrar novas idéias, ir à procura do mapeamento que é a iluminação ou a explicação.” “Outro bom caso para o <i>brainstorming</i> , quando os padrões emergem.”	Para aumentar a habilidade de gerar novas idéias, várias implicações são possíveis, tais como: fazer os relacionamentos graficamente explícitos e então permitir múltiplas comparações (i.e. Entre posições, escalas, distâncias, tamanhos, etc.); ou agrupar toda a informação necessária que é usada junto de uma maneira facilmente acessível. Ou também providenciar funções interativas para aproximar ou afastar (mudar escalas) ou ver detalhes em contexto, relacionar elementos diretamente (filtrar ou combinar) ou para mudar perspectivas. Finalmente uma medida muito eficiente é enriquecer itens integrando múltiplas dimensões para cada elemento (através de cores, tamanhos, posições, símbolos, animações, comentários disparados por eventos de interface humana (mouse, luvas digitais, <i>scanners, touch-screens, ...</i> ))
6. Modificabilidade	Os itens da visualização podem ser modificados em resposta à dinâmica da discussão?	A Modificabilidade é alta se cada item da imagem puder ser mudado facilmente (mover, eliminar, substituir). Se os itens são difíceis de mudar, ou se as mudanças não são possíveis, então a Modificabilidade é baixa. Baixa modificabilidade pode ser útil quando encoraja outros modos de pensar antes que as modificações sejam feitas, enquanto a alta modificabilidade melhora a possibilidade de interação.	Enraizamento: “Resistência ao movimento de objetos ou seu arrançamento” Viscosity: “Resistência à mudança” Inconveniência útil: “Interfaces inconvenientes podem forçar o usuário a refletir sobre a tarefa, com um ganho geral de eficiência”	“Em um cenário típico, uma pessoa desenha algo no quadro branco, então, se alguma outra pessoa não concorda, ou tem algo a destacar, ou adicionar, ela é convidada a ir ao quadro branco e modificar o diagrama. Isto encoraja outras pessoas a construir juntas um cenário compartilhado.”	Caneta e papel é um meio muito mais modificável do que um aplicativo de desenho (apesar de que, ironicamente, a sua finalização pode ser percebida como o oposto do que este meio é realmente). Software tem um alto valor de Modificabilidade porque os itens podem ser movidos, eliminados e copiados muito facilmente. Devido a isto, as apresentações projetadas (slides de Power Point, por exemplo) são mais facilmente modificáveis pelo grupo, porque a apresentação é controlada por um facilitador, e com frequência somente projetada na parede. Desenhar em papel (como em um <i>flipchart</i> ) com uma caneta não é um recurso altamente modificável porque os itens podem ser somente adicionados, mas não eliminados ou movidos, enquanto tais recursos seriam possíveis usando um lápis ou desenhando em um quadro branco.
7. Suporte à Interação de Grupo	A visualização ajuda a facilitar ou estruturar a interação de um grupo de pessoas?	Esta dimensão descreve a capacidade de manter a interação de grupo e a discussão no caminho certo. Se a visualização permite indicadores tais como o rastreamento de contribuição de participantes, avaliação progressiva do desenvolvimento da discussão, reprodução e modificação simultânea / seqüencial de padrões, então o Suporte à Interação de Grupo é alta. Se não há indicadores suportados, ela é baixa. Outros elementos que providenciam Suporte à Interação de Grupo são a possibilidade de referência aos itens no diagrama (apontamento ou chamada) para a facilitação aos participantes, da referência aos elementos da visualização, e documentação, que permite os participantes a voltar a certo ponto no tempo (história) ou reproduzir toda a interação.	Referenciabilidade: Reflete a facilidade com que quais participantes na conversação podem se referir a elementos da visualização Controlabilidade: Facilita a comunicação permitindo a um apresentador responder às questões do grupo de forma dinâmica	“É uma péssima contextualização se alguém toma algum papel para desenhar sobre ele; então não é possível compartilhar (este diagrama).”	Avaliação progressiva pode ser rastreada por uma metáfora visual, tal como uma seta atingindo um alvo, ou um termômetro. Rastreabilidade pode ser obtida, por exemplo, usando diferentes cores para a contribuição de cada pessoa, ou usando <i>templates</i> específicos que são projetados para suportar a coordenação focando em um modelo de contribuição pré-determinado.

Quadro 8: Fatores de Design: Descrição, Dimensões Similares, Citações de Praticantes, e Implicações de Design. Fonte: Bresciani et. al. (2008).

Título do Critério	Questão Orientadora	Contrapartidas (Tradeoffs)	Escala (Scale)	Exemplo de baixo valor (Example Low)	Exemplo de alto valor (Example High)
1. Impacto Visual	Quão atrativa é a visualização?	Baixo <i>Impacto Visual</i> pode desapontar ou produzir tédio. Uma quantidade apropriada de <i>Impacto Visual</i> pode suportar <i>Foco</i> direto (ícones e símbolos convencionais, por exemplo), mas um nível muito alto e inapropriado (tal como uma visualização que lembra uma peça de arte) pode distrair. A quantidade adequada pode não ser determinada <i>a priori</i> , mas depende da atividade e do objetivo da colaboração.	1: Genérica / baixa qualidade 2: básico 3: médio 4: distintivo 5: lembra um trabalho artístico	Alto: Metáfora visual com idéias dispostas em fogos de artifício 	Baixo: Gráfico de fluxo. Papel e Lápis 
2. Clareza	A visualização é facilmente compreendida com um baixo esforço cognitivo?	Existem divergências entre <i>Clareza</i> e <i>Impacto Visual</i> : uma visualização projetada para ser clara não deve incluir decoração excessiva ou elementos artísticos. Finalmente, alta <i>Clareza</i> provê uma compreensão rápida, mas pode conduzir a uma super simplificação.	1: confuso, 2: difícil de entender, 3: um pouco claro 4: muito claro 5: imediatamente claro	Alto: diagrama de Venn 	Baixo: dinâmica de sistemas 
3. Trabalho Final Percebido	A visualização convida a contribuições e modificações, ou ela lembra um produto finalizado e polido?	A <i>Percepção de Trabalho Terminado</i> é influenciada pelo <i>meio</i> , o <i>nível de modificabilidade</i> , e o <i>Impacto Visual</i> , porque um diagrama desenhado cuidadosamente que lembra um pedaço de publicidades de desktops, é percebido como sendo um trabalho terminado	1: percebido como em progresso 2: percebido como croqui 3: percebido como modificável 4: percebido como terminado 5: percebido como polido, ou otimizado	Alto: Onda de Forrester 	Baixo: Roadmap de planos de trabalho 
4. Foco Direccionado	A visualização dirige a atenção para os itens principais da discussão?	Uma quantidade moderada de <i>Foco</i> aumenta a <i>Clareza</i> , mas <i>Foco</i> em muitos itens reduz a <i>Clareza</i> , porque há uma competição por <i>Foco</i> .	1: nenhum 2: em muitos itens (disperso) 3: em muitos itens 4: em poucos itens 5: em um item principal	Alto: Quadrante Mágico modificado de Gartner 	Baixo: Ciclo de Moda de Gartner 
5. Facilitação de Ideias	As novas idéias são geradas como resultado da forma de visualização?	Procurar idéias de maneira freqüente conduz à redução de <i>Clareza</i> , por causa da maneira não convencional, ou não familiar de ver a informação. As formas de visualização que suportam a geração de inferências geralmente não são relacionadas, ou não respondem bem em termos de <i>Impacto Visual</i> .	1: nenhum 2: algumas idéias 3: muitas idéias 4: muitas idéias evidentes 5: várias idéias importantes	Alto: tabela de critérios para tomada de decisões 	Baixo: diagrama em cubo 
6. Modificabilidade	Os itens de visualização podem ser modificados em resposta à dinâmica da discussão?	<i>Modificabilidade</i> afeta suporte a <i>Interação de Grupo</i> e <i>Percepção de Término de Trabalho</i> : Um ambiente altamente modificável encoraja um alto nível de contribuição, porque as mudanças podem ser feitas facilmente, portanto as pessoas assumem mais riscos e contribuem mais freqüentemente.	1: não é possível 2: difícil 3: possível 4: fácil 5: fácil e rápido	Alto: post-its e comentários escritos a caneta sobre quadro branco 	Baixo: diagrama improvisado, caneta sobre papel 
7. Suporte à Interação de Grupo	A visualização ajuda a facilitar ou estruturar a interação de um grupo de pessoas?	O <i>meio</i> tem uma forte influência no <i>Suporte à Interação de Grupo</i> , como interações baseadas em <i>software</i> podem ser mais facilmente documentadas e freqüentemente suportam registros históricos; aplicações específicas para trabalho colaborativo tem um suporte especial para a avaliação progressiva, rastreabilidade e reprodução histórica.	1: nenhum 2: muito limitado 3: algum 4: grande 5: muito grande	Alto: template de poster com notas de post-it 	Baixo: apresentação projetada com gráficos embutidos 

Quadro 9: Fatores de Design: Contrapartida, Escala, Exemplo de Alto Valor, Exemplo de Baixo Valor. Fonte: Bresciani et. al. (2008).

<b>Autor</b>	<b>Dimensão</b>	<b>Descrição</b>
<b>Star e Griesemer (1989)</b>	Plástico-robusto	Estruturas gráficas plásticas a serem adaptadas a necessidades locais, estruturas robustas para manter uma identidade comum
	Fracamente/Fortemente estruturado	Fracamente estruturado em uso comum; pode se tornar fortemente estruturado em um uso individualista
	Abstrato-concreto	Abstrato refere-se à representação conceitual; concreto refere-se a imagens específicas
	Significado diferente – estrutura comum	Significado diferente em mundos sociais diferentes, mas a sua estrutura é comum o bastante para ser reconhecível
	Geral-específica	Específica para uma aplicação; geral para ser compreensível
	Convencionado-customizado	Convencional para ser reconhecível; customizado para ser mais significativo
<b>Green e Petre (1996)</b>	Abstração	Tipos e disponibilidade de mecanismos de abstração
	Dependências escondidas	Ligações importantes entre entidades não são visíveis
	Compromisso prematuro	Restrições na ordem de execução das coisas
	Notação Secundária	Informação extra em meios diferentes dos da sintaxe formal
	Viscosidade	Resistência à mudança
	Visibilidade	Habilidade de ver os componentes de forma fácil
	Proximidade de mapeamento	Proximidade de representação de domínio
	Consistência	Semânticas similares são expressas em formas sintáticas similares
	Difusividade	Verbosidade da linguagem
	Tendência a erros	A notação induz a enganos
	Operações mentais difíceis	Alta demanda de recursos cognitivos
	Avaliação progressiva	O andamento do trabalho pode ser verificado a qualquer instante
	Provisoriedade	Grau de compromisso com ações ou marcos
	Expressividade de papéis	O propósito de um componente é imediatamente inferido
<b>Blackwell (2001)</b>	Ambigüidade criativa	Uma notação encoraja o usuário a ver algo diferente
	Especificidade	Os elementos têm um número limitado de significados potenciais
	Detalhe em contexto	Como os elementos se relacionam com outro na mesma camada de notação
	Indexação	Inclui elementos para ajudar o usuário a encontrar partes específicas
	Síntese	Entendimento do todo. “chegue para trás e olhe”
	Navegação livre	Nova informação é gerada como resultado de ter seguido as regras notacionais
	Inconveniência útil	Interfaces inconvenientes forçam o usuário a refletir sobre a tarefa
	Disparidade	O sistema empurra suas idéias em uma certa direção
	Adaptabilidade a mudanças	A notação muda a forma rapidamente
	Permissividade	A notação permite vários caminhos diferentes de se fazer as coisas
<b>Hundhausen (2004)</b>	Saliência de programação	Qualquer coisa que o usuário foque durante a construção de uma visualização tende a se tornar o foco da discussão
	Provisoriedade	Extensão à qual a visualização remonta a um produto final
	Conteúdo histórico	Retrata os conceitos do domínio em termos de uma história
	Modificabilidade	Grau em que a visualização pode ser dinamicamente alterada em resposta à dinâmica da discussão
	Controlabilidade	Permite a um apresentador, dinamicamente, responder à audiência
	Possibilidade de referencia	Os participantes podem se referir a elementos da visualização
<b>Eppler (2004)</b>	Foco	Prestar atenção no recurso
	Coordenação	Estrutura passo a passo para organizar a interação
	Documentação	Os resultados alcançados são documentados
	Consistência	Os participantes podem fazer contribuições mais consistentes quando eles podem ver o que já existe compartilhado
	Responsabilidade por uma tarefa	As contribuições dos participantes são capturadas e documentadas
	Rastreabilidade	Reconstrução da interação e fluxo de idéias
<b>Karabeg (2006)</b>	Imediatismo visual	A primeira impressão; característica que permite o observador a perceber e reconhecer “de primeira”
	Ímpeto visual	Quão atrativo e convidativo à ação e posterior exploração
	Impedância visual	O impedimento; a implicação negativa freqüente e não intencional que é o motivo do observador ser menos receptivo

Quadro 10: Dimensões Cognitivas. Fonte: Blackwell et. al. (2001).



Blackwell *et. al.* (2001) criaram uma *framework*, denominada ‘Dimensões Cognitivas de Notações’(Quadro 10), com a intenção de auxiliar *designers* de ferramentas de visão de informação a avaliarem o impacto que seus trabalhos causarão em usuários dessas ferramentas. Os autores apresentam uma tentativa de caracterizar as formas em que os instrumentos da mente humana estão comprometidos pelas restrições impostas pelo nosso corpo e ambiente físico.

De início, os autores consideraram somente a visibilidade oferecida por essas ferramentas, onde o usuário apenas lê informações no *display* e aponta a necessidade de possuir uma notação de dimensões cognitivas para atividades mais complexas, tais como: criar, modificar e estruturas de informação, adicionar dados a essas estruturas ou explorar possíveis opções de *design* para estruturas de informação completamente novas.

Para Blackwell *et. al.* (2001), apesar de a visibilidade ser importante para todas essas atividades, muitas impõem restrições adicionais ao usuário, além da simples percepção e interação física. Alguns exemplos de restrições incluem a dificuldade em fazer pequenas mudanças na estrutura de informação (o que os autores definem como viscosidade), ou ainda as formas de como o usuário pode expressar partes de sua estrutura de informação que ainda não foram definidas (*provisionality* ou possibilidade de improvisação).

Os autores chamam esses atributos de Dimensões Cognitivas das Notações (*Cognitive Dimensions - CDs*). Da mesma maneira, assim como a visibilidade tem uma relação pré-determinada com a leitura, também as outras Dimensões Cognitivas podem ser usadas para prever as consequências do uso de artefatos de informação em outros tipos de atividade.

Eles recomendam o uso dessas dimensões porque profissionais (que projetam novos artefatos de informação: as tecnologias cognitivas) freqüentemente encontram os mesmos problemas, repetidas vezes.



## **4 MÉTODO DE CONSTRUÇÃO DO EDITOR DE ONTOLOGIAS MULTILINGUE**

### **4.1 IMPLEMENTAÇÃO DO PROTÓTIPO DE VISÃO DE REDES DE ONTOLOGIAS**

Antes da implementação do protótipo, deve-se considerar a sua finalidade. Inicia-se com a criação de um conjunto de ferramentas para atividades específicas, com foco na visualização e na comunicação das informações geridas pelos engenheiros do conhecimento.

#### **4.1.1 Requisitos funcionais**

Para que o sistema possa representar visualmente o conteúdo pretendido, na sua forma evolutiva, ele deve oferecer artifícios de compressão e expansão visual, ou seja: o agrupamento de vértices, arestas e conjuntos, compactados em um bloco único, também representado como um vértice.

Como a quantidade de dados contidos em um agrupamento desse tipo pode ser muito grande, com fins de manutenção de performance, o sistema deve oferecer de forma transparente ao usuário artifícios de gerenciamento de dados em memória, com registro temporário desses dados na memória de disco.

Como elementos básicos de representação, o sistema deve satisfazer as seguintes condições:

1. Permitir ao EC ver e manipular termos como vértices.
2. Permitir ao EC inserir informações visuais adicionais a um termo escolhido.
3. Permitir ao EC ver e manipular relações como arestas.
4. Permitir a criação de várias relações entre dois termos.
5. Permitir ao EC ver e manipular domínios como conjuntos.
6. Permitir o encadeamento de domínios em vários níveis.
7. Permitir incluir as arestas em vários domínios.
8. Permitir ao EC a parametrização dos elementos gráficos que descrevem termos, relações e domínios.

9. Dispor formas de representação de relações.
10. Possibilitar a criação de novas relações pelo próprio EC.
11. Possibilitar a criação de regras de inferência e *scripts* para automação da recuperação de informação.
12. Permitir a visão do encadeamento destas regras para diagnóstico do processo automático de recuperação.
13. Oferecer um ambiente integrado de desenvolvimento de ontologias e indexação de documentos.
14. Como funcionalidades básicas de comunicação da informação entre os EC's, o sistema deve satisfazer as seguintes condições:
15. Permitir o transporte e compartilhamento de dados das redes de ontologias entre os EC's.
16. Permitir a persistência/carga de repositórios de ontologias em vários formatos.
17. Permitir a concentração de dados em um ambiente distribuído (redes de máquinas - GRID) ou centralizado (*mainframe*).

#### **4.1.2 Prototipagem para testes de viabilidade**

Na comunicação de idéias entre pessoas, são usados meios e ferramentas comuns, digitais ou não, de representação da informação. Os meios e ferramentas usados são normalmente compostos por um espaço para representação geométrica e um elemento para escrita geométrica. (Ex.: papel e lápis, monitor e mouse para produção de desenhos, quadro e giz, quadro branco e caneta tinteiro.)

Durante as várias discussões de modelos sobre gestão de informação usando ontologias, foi percebido que as redes de ontologias e os domínios eram descritos no papel e no quadro branco. A partir dessas observações, surgiram necessidades de visão de informações de redes de ontologias, em formato digital, aplicando os mesmos elementos gráficos utilizados no papel.

De início, fez-se um levantamento prévio de quais elementos de representação eram necessários. Entretanto, não havia certeza da implementação desses elementos. Para isso, são necessários testes de verificação do ponto em que é possível criar os componentes visuais. Nesse contexto, formulou-se o seguinte questionamento: quais

elementos visuais podem ser usados para representar as ontologias e como esses elementos podem ser combinados?

### 4.1.3 Modelos de domínio para a ontologia multilíngüe

Neste item, são apresentados e justificados dois modelos de domínio, representados por diagramas de classe da *Unified Modeling Language*.

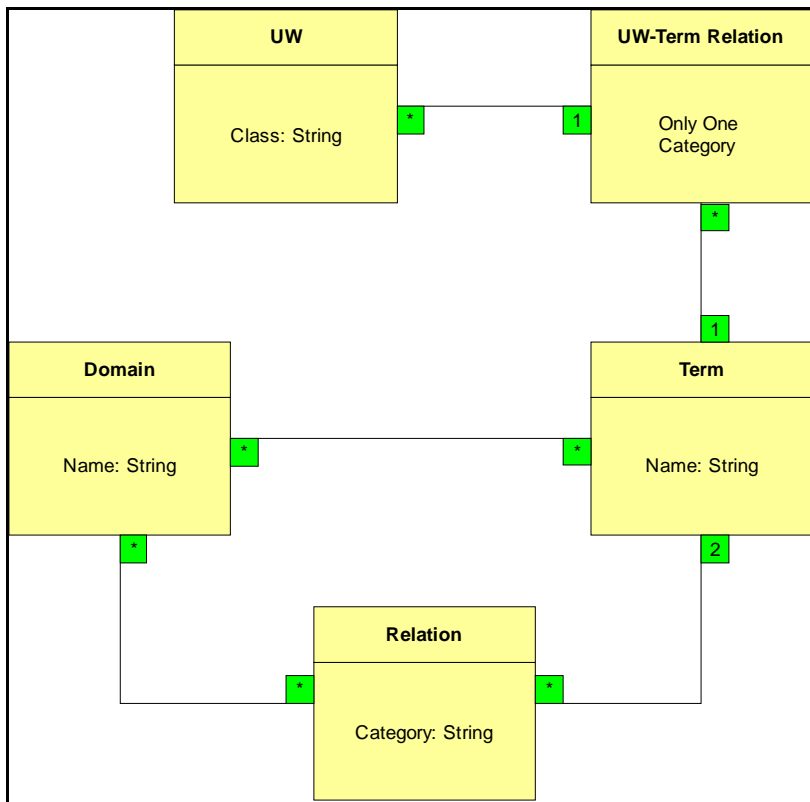


Figura 34:Primeira versão do modelo de domínio proposto para o Editor de Ontologias Multilíngües.

Na primeira versão do modelo de domínio (Figura 34), a palavra universal (UW) pode possuir conexões com N relações de equivalência com os termos nos vários idiomas. Um termo pode possuir conexões com N relações de equivalência com as palavras universais. Um termo pode pertencer a N domínios. E um domínio pode conter N termos. Uma

relação semântica deve estar conectando dois termos, e um mesmo termo pode conter N relações semânticas de ontologias. Um domínio pode conter N relações semânticas. E uma relação semântica pode pertencer a N domínios.

Neste modelo não há relações semânticas entre palavras universais que permitam uma equivalência entre ontologias em vários idiomas e a ontologia normalizada de palavras universais. Também não há o conjunto dos idiomas que contem seus respectivos domínios. Partiu-se, então, para o segundo modelo proposto de domínio para as ontologias multilíngües.

A segunda versão do modelo de domínio para o editor de ontologias multilíngües apresenta as seguintes características (Figura 35):

1. **Nomes de domínios representados como UW's:** Este modelo prevê que os nomes de cada domínio dentro da ontologia normalizada (Dominio-UW) também são representados por uma palavra universal. A relação UW entre duas palavras universais é a mesma relação entre os termos representados por estas palavras.
2. **Um domínio principal deve pertencer a um e somente um idioma:** O modelo contém uma lista de idiomas, em que cada idioma pode possuir N domínios, e um domínio deve estar contido em apenas UM idioma, representando uma relação de composição. Parte-se do principio de que, considerados dois idiomas diferentes, dois domínios que pertençam cada um a um idioma diferente nunca serão idênticos a ponto de permitir a simplificação em que um mesmo domínio pertença a esses dois idiomas. Por outro lado, qualquer domínio sempre representa um conjunto de termos, subdomínios e relações que têm um significado dentro de um idioma pré-estabelecido.

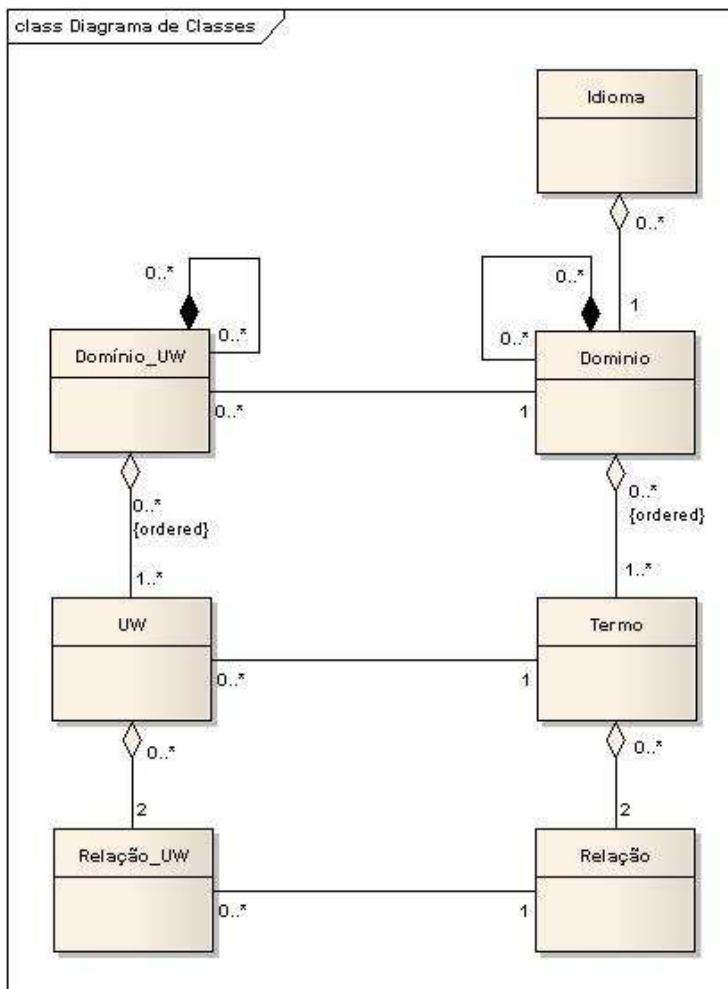


Figura 35:Segunda versão do modelo de domínio para o Editor de Ontologias Multilíngües.

3. **Um domínio pode conter N ou nenhum subdomínio, e um subdomínio pode estar contido em N domínios simultaneamente:** Um domínio (ou domínio-pai) pode conter vários subdomínios (ou domínios-filhos), e um domínio-filho pode pertencer a vários domínios-pais. De maneira equivalente, para a ontologia normalizada composta de palavras universais,

cada domínio-UW pode conter vários subdomínios UW, e cada subdomínio-UW pode pertencer a vários domínios-UW.




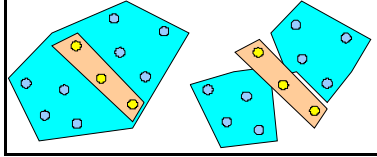
4. **Um domínio pode conter N ou nenhum termos, e um termo pode estar contido em de 1 a N domínios simultaneamente, tanto na ontologia de um idioma, quanto na ontologia normalizada:** Quanto aos termos, um domínio pode possuir de 0 a N termos, e um termo pode estar contido de 1 a N domínios. Dessa forma, não existem termos isolados. De maneira equivalente, na ontologia normalizada, uma UW pode pertencer de 1 a N domínios-UW.
5. **Sobre as relações semânticas:** Uma relação semântica deve estar conectada a dois termos, na ontologia de um idioma, assim como uma relação UW deve estar conectada a duas palavras universais.

#### **4.1.4 Elementos para representação de componentes das ontologias**

O processo usado para previsão de cenários de representação visual, antes do trabalho experimental e da construção do protótipo, foi enumerar os elementos visuais que poderão participar dos diagramas de representação de ontologias. Após o levantamento dos tipos de elementos participantes, uma análise de combinações foi elaborada, sem a exatidão ou rigor matemático que envolve a lógica ou que tenta levantar as possíveis conseqüências de cada tipo de combinação. A combinação desses elementos também foi feita visualmente, de maneira experimental, gerando algumas possibilidades. Somente uma pequena parte dessas combinações foi implementada no protótipo, já que o seu gerenciamento é bastante complexo, sendo garantido somente após extensas baterias de testes e correções.

A seguir, são apresentados os elementos visuais básicos que participam dos diagramas e suas combinações, com o objetivo de explicar algumas variantes que podem se combinar de diferentes formas.

## 4.1.5 Formas de representação

Item	Símbolo	Descrição
Espaço	∅	Lugar geométrico e abstrato que contém os elementos visuais e onde são aplicados operadores simbólicos (lógicos e matemáticos)
Vértice		<p>Basicamente é um elemento abstrato, contido em um espaço, que representa uma informação isolada, discreta. Este elemento também pode representar um grupo de outros elementos, mas, inicialmente, é usado para representar visualmente um objeto isolado, com características gráficas determinadas por seus parâmetros internos.</p> <p>O vértice pode representar um grupo de forma visual concentrada: Elementos pertencentes a um grupo são vistos como um vértice único, economizando espaço, e evitando a apresentação da informação interna irrelevante para uma dada análise.</p> <p>A desvantagem nesta forma de apresentação é que os dados internos não podem ser vistos de maneira imediata, necessitando de uma expansão dos dados internos do vértice. É um problema muito semelhante à procura de serviços por meio de menus: Não se pode ver o submenu de um menu, sendo necessária a procura a partir dos primeiros níveis do menu principal, pelo método de tentativa e erro.</p> <p>Para apresentar informações de grupo de forma dispersa, onde é possível ver o grupo e seu conteúdo, há a opção de ver grupos cercados por uma ENVOLTÓRIA.</p> <p>Finalmente, o vértice possui uma EXTENSÃO e uma FORMA. Ele também pode ser construído pela composição gráfica dos vários elementos que o constituem, tais como cores, tracejados, contornos (um vértice pode ser um ícone, um triângulo, ou mesmo o contorno de um mapa), espessuras de linha, dégradés e contrastes.</p>
Aresta		<p>Elemento abstrato e contínuo que designa alguma forma de relação entre dois elementos isolados: Os elementos, inicialmente isolados, que são ligados por uma aresta, representam elementos que compartilham uma propriedade em comum.</p> <p>Uma das necessidades satisfeitas pela existência de uma aresta é a direcionalidade: Dois elementos discretos e isolados compartilham uma propriedade que indica que um dos elementos é o inicial e o outro final, ou que um desses elementos é o elemento ativo e outro o elemento passivo, ou um é a causa e o outro a consequência.</p> <p>Neste caso, a aresta é representada com um desenho em um de seus extremos: uma seta, garra, estrela, tridente, entre outros, dependendo do significado desejado na representação.</p>
Envoltória		<p>Assim como em um dos casos de utilidade do vértice, a envoltória representa um grupo, um continente, de elementos discretos, estejam estes ligados ou não. A envoltória indica que os elementos dentro dela estão CONTIDOS no elemento abstrato representado por esta envoltória, que representa os limites gráficos de um conjunto.</p> <p>Em muitos casos, graficamente, uma envoltória pode conter um elemento discreto dentro de seus limites, apesar de este elemento não pertencer ao grupo representado por esta. Neste caso, há uma solução hipotética, onde a envoltória se dá pelo envolvimento de <i>clusters</i> de elementos discretos, evitando o envolvimento de elementos não pertencentes ao grupo. Um exemplo da solução é mostrado abaixo: na primeira figura, a envoltória azul limita os círculos azuis e a envoltória bege envolve os círculos amarelos. O analista tem a impressão de que tanto os círculos azuis, quanto a envoltória bege, com seus círculos amarelos, pertencem à envoltória azul claro. A determinação da envoltória por clusters (subgrupos de elementos determinados pela proximidade entre eles) esclarece a real pertinência entre círculos e envoltórias: Os círculos amarelos pertencem à envoltória bege. A envoltória bege não pertence a nenhuma envoltória, e os círculos azuis pertencem à envoltória azul claro.</p> 

Quadro 11: Formas de representação selecionadas.


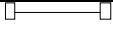


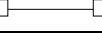

#### 4.1.6 Combinação das formas de representação

Após o levantamento dos elementos isolados, são previstas combinações dos elementos, de forma visual, para representá-los nas futuras ontologias multilíngües. As Tabelas Tabela 2 e Tabela 3 mostram as possíveis combinações entre os quatro elementos geométricos básicos: espaço, vértices, arestas, conjuntos. Esses quatro elementos estão dispostos nas linhas e nas colunas da tabela, para que sejam vistas as combinações de uma entidade com outra.

Notando que podem existir mais combinações gráficas entre cada entidade, duas a duas, todas as células da tabela de combinações foram ocupadas, de tal maneira que, dentro da célula (aresta/conjunto), foram colocadas algumas combinações desses elementos; nas mais simplificadas, e dentro da célula (conjunto/aresta), foram colocadas as variações mais complexas desses dois elementos.

A distribuição seguiu o mesmo método para as demais combinações (espaço-espaço, espaço-vértice, espaço-aresta, espaço-conjunto, vértice-vértice, vértice-aresta, vértice-conjunto, aresta-aresta, aresta-conjunto e conjunto-conjunto). Cada uma das células da Tabela 3 está referenciada com um índice matricial Linha X Coluna: L1 X C1, L2 X C1, etc. Para uma visão mais genérica, é mostrada a Tabela 2:


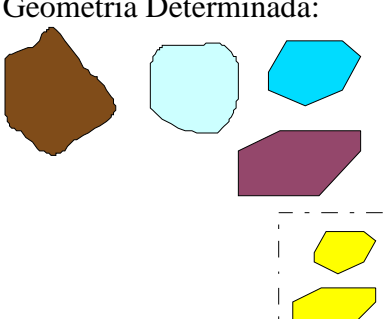
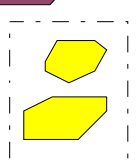
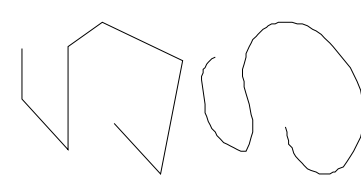
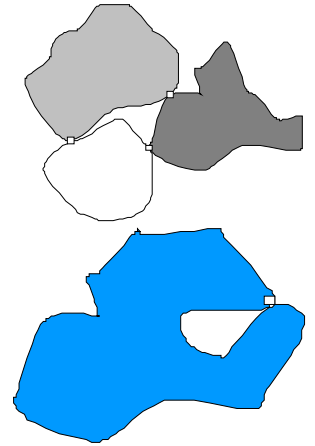
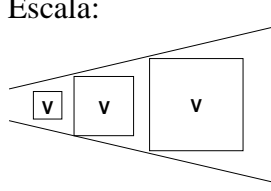
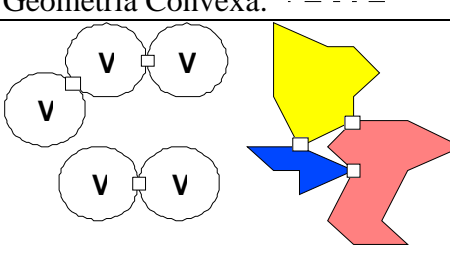
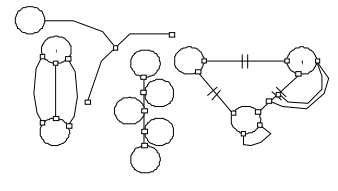
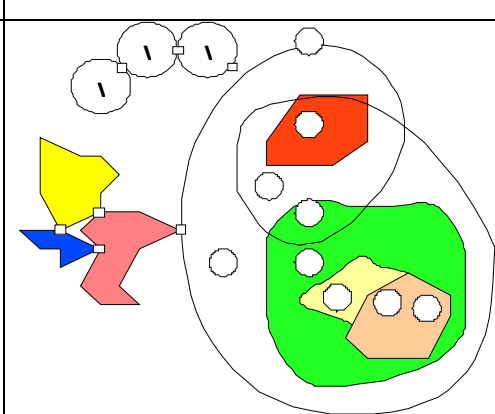


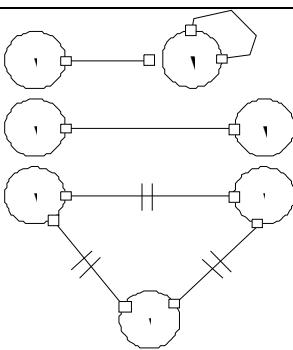
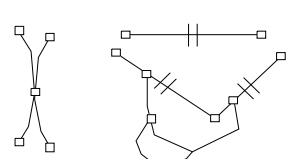
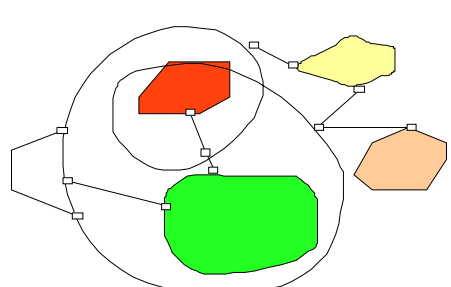
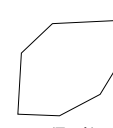
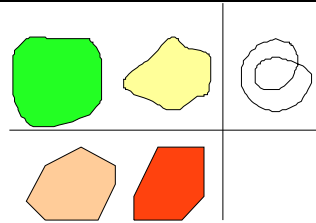
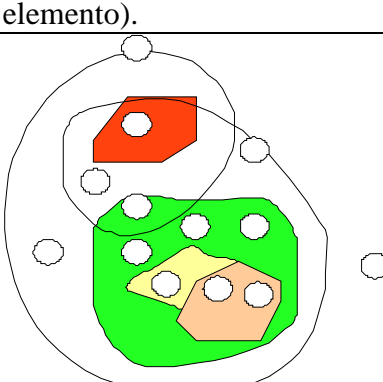
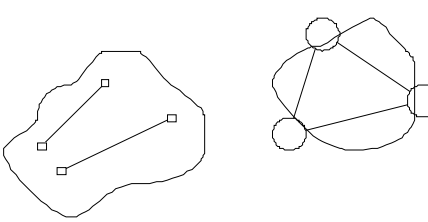
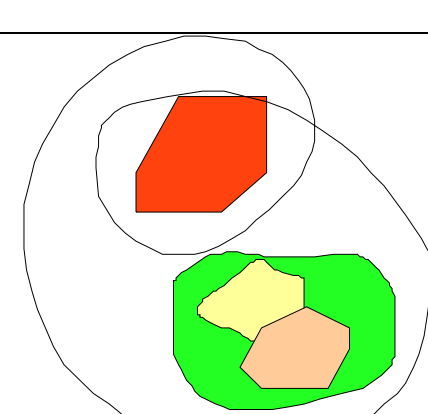
**Tabela 2: Visão geral das quatro tabelas de combinação de elementos visuais.**

	∅ (C1)	 (C2)	 (C3)	 (C4)
∅ (L1)				
 (L2)				
 (L3)				
 (L4)				

Na Tabela 3, são mostradas as possíveis combinações: entre dois espaços diferentes (C1 e L1), pode haver operações de fusão, intersecção, complemento ou subtração de dois diagramas. Entre um espaço e um vértice pode ocorrer a simples existência de um vértice em um espaço dado (L2 e C1) ou a existência de um vértice que esteja ampliado ou reduzido, nas dimensões do espaço cartesiano em que está inserido (também em L2 e C1).



Tabela 3: Combinações entre quatro elementos geométricos básicos: (espaço, vértice, aresta, envoltória) X (espaço, vértice, aresta, envoltória).

	$\emptyset$ (C1)	$\mathbf{V}$ (C2)	$\square$ — $\square$ (C3)	 (C4)
$\emptyset$ (L1)	Operadores matemáticos e lógicos: $\emptyset 0 + \emptyset 1$ ou $\emptyset 0 \cup \emptyset 1$ $\emptyset 0 \cap \emptyset 1$ não $\emptyset$ ou complementar de $\emptyset$ $\emptyset 0 - \emptyset 1$	Simplex: $\mathbf{v}$  Geometria Determinada:   Geometria Convexa: 		
$\mathbf{V}$ (L2)	Existência no espaço: $\mathbf{v}$  Escala: 	  Observar os pontos de contato entre os vértices, representados como pequenos quadrados.		
$\square$ — $\square$ (L3)	Simplex   Conectada a si mesma 	  Uma “hiperaresta” contendo três vértices. (Note os traços duplos Indicando a pertinência ao mesmo elemento).		
 (L4)				

Estendendo as possibilidades da combinação entre um espaço e um vértice (L1 e C2), um vértice pode não somente ser representado por um ponto, um ícone, ou uma geometria pré-definida (círculo, quadrado ou triângulo), mas sim por uma geometria mais genérica, que é determinada pelos seus pontos limítrofes.

A terceira variante de (L1 e C2) é a existência do vértice como uma geometria genérica, inclusive composta por várias regiões limítrofes (uma geometria convexa), não devendo ser confundido o vértice com geometria genérica com a envoltória, pois por definição, visualmente, este vértice não contém outros vértices dentro de si, tal como a envoltória.

Por último, as possíveis combinações entre vértice e vértice (L2 e C2) podem resultar em vértices **fixados** entre si através de pontos de contato. Esses vértices podem permanecer livres para a movimentação conveniente do usuário dentro do espaço de visualização dos dados, mas suas posições relativas estão constantes.

São mostradas, também, estas possíveis combinações: a combinação entre um espaço e uma aresta (L1 e C3) pode resultar na existência da aresta dentro desse espaço, seja esta aresta discreta, ou contínua.

A combinação entre um espaço e uma envoltória pode resultar na envoltória somente, ou ainda, as envoltórias que podem existir no espaço podem estar conectadas entre si (as envoltórias cinzas), através de pontos de contato, ou ainda uma envoltória que seja convexa pode ter pontos de contato entre os seus próprios limites (a envoltória azul).

A combinação entre vértice e aresta (L2 e C3) pode resultar em quatro casos: Várias arestas entre dois vértices; Uma aresta conectando um vértice a uma segunda aresta; Vários vértices conectados a uma aresta através de sua posição relativa; Uma aresta conectando um vértice a ele mesmo; ou ainda um elemento contendo mais de um vértice, e este elemento é representado por N arestas (para este caso, o número N é dois, e as arestas que o representam estão cortadas por dois pequenos traços paralelos).

Também são mostradas estas possíveis combinações: um pouco diferente do caso (L1 e C3), a combinação entre um espaço e uma aresta (L3 e C1) pode resultar em uma aresta simples, sem conexões em seus

extremos, ou uma aresta cujo extremo coincide com um ponto que pertence a esta mesma aresta.

A combinação entre uma aresta e um vértice (C2 e L3) pode resultar em uma conexão simples de uma aresta com um vértice em um de seus extremos, ou uma aresta conectada a dois vértices pelos seus extremos, ou uma aresta conectando um mesmo vértice a ele mesmo. Também tal como representado na Tabela 3, (combinação L2 e C3) há a possibilidade de representar um mesmo elemento com vários segmentos de reta entre um dado número de vértices: Uma hiperaresta.

A combinação entre uma envoltória e um espaço (L4 e C1) pode resultar na envoltória, simplesmente. Foi imaginada a representação do caso de uma envoltória que contém ela mesma, como se fosse uma recursividade (o desenho em formato de cardióide, na parte direita da célula L4, C1).

A combinação entre uma envoltória e um vértice (L4 e C2) pode gerar a situação em que uma envoltória contém um vértice (ou vários), ou ainda, em que um vértice está preso na linha limítrofe de uma envoltória.

Também são mostradas as seguintes combinações: a combinação entre arestas (L3 e C3) pode gerar situações em que duas arestas podem estar conectadas entre si através de um ponto intermediário, ou ainda, dado um caso em que um elemento contém N vértices e é representado por um número de segmentos de reta (representados por linhas com traços perpendiculares, tal como em L3 e C2 ou L2 e C3), pode existir uma aresta que tenha seus dois extremos conectados a este elemento.

A combinação entre uma envoltória e uma aresta (L4 e C3) pode gerar situações onde a envoltória contém a aresta, ficando ao redor dos pontos que determinam a extremidade das arestas que pertencem a esta envoltória.

A segunda forma de combinação entre uma envoltória e uma aresta (L3 e C4) pode gerar casos em que uma envoltória é ligada a outra envoltória, através de uma ou mais arestas, ou ainda, que uma mesma aresta esteja conectada a uma envoltória através de seus dois pontos extremos.

A combinação entre duas envoltórias (L4 e C4) pode representar somente o caso em que uma envoltória contém a outra em seu interior. Para o caso de uma envoltória que contém a si mesma (como visto no

caso L2 e C4), dentro do enésimo laço da envoltória recursiva podem estar contidas outras envoltórias.

## 4.2 DESCRIÇÃO DOS MODELOS ESTRUTURAIS E MATEMÁTICOS

O modelo estrutural usado foi construído sobre os componentes *VCL* da *Borland Incorporated*. Uma *framework* básica existente em linguagens tais como Delphi e C++ Builder.

Na Figura 36 está desenhado um diagrama contendo o relacionamento entre as classes. As linhas verdes – com setas – apontam da classe mãe para a classe derivada. As linhas marrons – com grampos – indicam que a instância da classe origem contém uma ou várias instâncias da classe de destino. As linhas vermelhas indicam que as instâncias das classes estão associadas: uma possui um ponteiro para a outra. A representação das classes da *framework* foi construída com o próprio aplicativo de construção de ontologias.

O Quadro 12, apresenta a descrição das classes usadas para a representação da *framework*, e do aplicativo de visão de redes de ontologias. As cores das células são apenas uma tentativa de agrupar as classes por nível hierárquico (*framework* / aplicação).

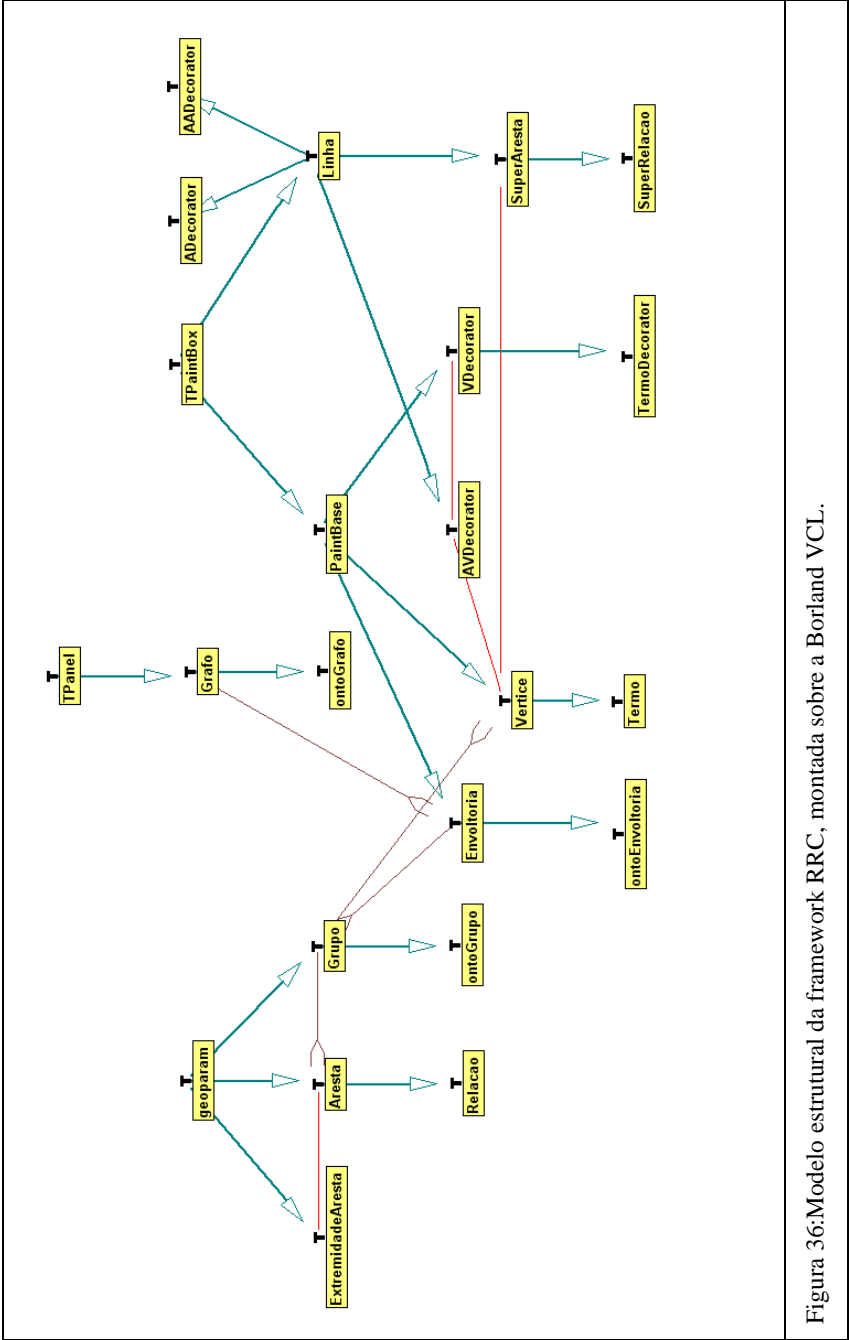


Figura 36: Modelo estrutural da framework RRC, montada sobre a Borland VCL.

TPanel (VCL)	É um container de componentes visuais.
Grafo (RRC)	Cria SuperAresta, Cria e destrói Arestas
ontoGrafo (Aplicativo)	Cria e Destrói vértices
geoparam (RRC)	Parâmetros geométricos de visualização: cor, espessura e tracejado
ExtremidadeAresta (RRC)	Uma poli-linha determinando o desenho de uma figura posicionada na extremidade de uma aresta
Aresta (RRC)	Uma poli-linha determina o desenho de uma figura entre dois vértices. Cada aresta contém uma lista dos grupos que a contem.
Grupo (RRC)	Uma classe contendo uma lista de Arestas, uma lista de vértices e uma lista também de grupos. Foi considerado que grupos podem estar dentro de grupos. Procurando alcançar a maior generalidade possível, um grupo pai pode conter vários grupos filhos, e um grupo filho pode estar contido dentro de vários grupos pais. Cada grupo contém uma lista dos grupos filhos contidos, e uma lista dos grupos pais que o contém. Um grupo é geometricamente determinado por oito pontos.
Relação (Aplicativo)	Esta classe pode ter várias instâncias dentro de uma classe SuperRelacao, portanto, cada uma das instâncias desta classe Relacao deve ser desenhada procurando-se destacar uma relação da outra. Para que tal aconteça, a classe relação tem seu desenho determinado por quatro pontos: Os dois pontos mais extremos coincidem com a posição dos vértices extremos da relação, e os dois pontos mais internos são posicionados a uma altura relativa ao número de relações entre os dois vértices.
ontoGrupo (Aplicativo)	Possui apenas uma alteração no método para desenhar o texto que o identifica
Envoltoria (RRC)	É um container gráfico onde são desenhados os contornos dos grupos contidos em sua lista.
ontoEnvoltoria (Aplicativo)	Cria um ontoGrupo determinado p/ um nome. A classe Envoltória cria uma instância da classe Grupo, dado seu nome sugerido
TPaintBox (VCL)	Usado para desenhar geometrias
PaintBase (RRC)	Usado para responder a eventos do mouse. Encapsula métodos de resposta a estes eventos, e também contém métodos de cálculo de limites de contorno
Linha (RRC)	Classe que desenha uma linha reta entre dois pontos nas coordenadas do espaço do Grafo. Dado que as coordenadas de uma instância desta classe podem se alterar de acordo com a posição dos seus vértices extremos, esta classe contém métodos que recalculam os dois pontos que resultam das coordenadas dos vértices, para que a linha entre estes dois vértices seja desenhada na direção correta.
Vertice (RRC)	Elemento responsável por responder a eventos do mouse, sofrer deslocamentos relativos, requisitar a atualização das envoltórias dos conjuntos, e da posição de suas arestas adjacentes, ser incluído e excluído de agrupamentos temporários.
SuperAresta (RRC)	É um container de arestas entre dois vértices. Responsável por métodos de criação e inclusão de instâncias de Arestas, atualização de posição dos pontos que definem as arestas, de acordo com o eixo definido pela posição de seus vértices extremos, e resposta a eventos do mouse.
ADecorator (RRC)	Classe responsável por conter informações a respeito de uma instância de uma classe SuperAresta. Uma mesma SuperAresta pode conter várias informações associadas, onde cada instância de informação pode ser representada graficamente por uma classe ADecorator. Esta classe não está sendo usada no aplicativo protótipo de visão de redes de ontologias.
AADecorator (RRC)	Classe responsável – inicialmente – apenas por desenhar uma linha entre uma SuperAresta e sua instância de classe ADecorator. Graficamente, esta classe serve apenas para informar graficamente ao usuário de que uma informação em forma de vértice, está associada a uma determinada SuperAresta.
VDecorator (RRC)	Considerando que uma classe Vértice é reconhecida graficamente por um ícone, é interessante que a framework ofereça formas de apresentar mais informações a respeito de um dado vértice, através de Decoradores (HEER[17]) deste vértice. Desta forma, é possível representar o vértice graficamente, e informações complexas associadas, com algumas versatilidades, tais como aparição e supressão destas informações gráficas mediante condições ou filtros de procura e alteração da forma gráfica destas informações em tempo real.
AVDecorator (RRC)	Classe que tem a função bem semelhante à classe AADecorator: Apenas para indicar graficamente – para o usuário – que um Vértice está ligado ao seu VDecorator. Vale lembrar que, assim como um vértice pode ter várias informações gráficas associadas, este também pode ter várias instâncias de classes VDecorators associadas, e este mesmo vértice possui uma instância de classe AVDecorator, para cada instância de classe VDecorator correspondente.
Termo (Aplicativo)	Contém dados do tipo de termo de ontologia (Se um termo ou Universal Word da UNL)
TermoDecorator (Aplicativo)	Contém métodos que atualizam seus parâmetros geométricos (cor, estilo e tamanho de caracteres, cor de fundo), métodos para redimensionar a caixa de texto, de acordo com o seu tamanho, e métodos para a captura inicial do texto contido no conceito representado pelo termo.
SuperRelacao (Aplicativo)	Contém métodos p/calcular a distância de cada instância da classe Relação, dentro de si, a partir do eixo determinado pelos vértices extremos.

Quadro 12: Descrição das principais classes estruturais da framework.

### 4.2.1 Descrição dos modelos matemáticos e algoritmos

Antes da implementação propriamente dita dos elementos visuais, devemos considerar que estes elementos, por sua própria natureza geométrica, têm seu funcionamento regido por fórmulas matemáticas.

Os itens abaixo descrevem os modelos matemáticos e/ou algoritmos utilizados na implementação dos cinco elementos geométricos básicos necessários para a existência da rede de relacionamentos e conjuntos: Vértices, várias arestas entre dois vértices, extremidades de arestas, conjuntos, e movimentação de vértices e conjuntos com o mouse.

#### 4.2.1.1 A Borland VCL e a programação orientada a eventos

As ações de periféricos sobre estes componentes visuais são programadas a partir de métodos, chamados de 'eventos'. Tais métodos já fazem parte da *framework VCL* da *Borland Incorporated*.

Dessa forma, são considerados eventos quaisquer atividades provenientes de manuseio de periféricos que agem dentro dos limites geométricos de um componente visual: pressionar e soltar uma tecla quando uma janela está ativa é um evento; passar com o ponteiro do mouse dentro dos limites de um componente visual é um evento; clicar com um dos botões do mouse, quando seu ponteiro está posicionado dentro dos limites geométricos de um componente é um evento; e assim por diante.

#### 4.2.1.2 Eventos do mouse

Na *framework RRC* proposta, o *mouse* possui um papel importante na manipulação de objetos visuais, já que este periférico é usado como meio caminho de recursos relevantes na manipulação de objetos.

Destacam-se três eventos básicos que guiam todos os processos de manipulação de elementos geométricos: 1) Pressão do botão do *mouse* quando seu ponteiro está sobre um componente visual (*MouseDown*); 2) Movimento do *mouse* quando seu ponteiro está sobre um componente visual (*MouseMove*); 3) Liberação do botão do *mouse* quando seu ponteiro está sobre um componente visual (*MouseUp*).

Por último, deve-se considerar que o movimento do *mouse* – quando seu ponteiro está sobre um componente visual da VCL – tem efeito somente quando o botão do *mouse* foi anteriormente pressionado sobre este componente, e continua abaixado quando o *mouse* é movimentado. Quando o botão é levantado, o componente não sofre mais o efeito do movimento do *mouse* (neste caso, se diz que o componente foi **liberado**. No caso de pressão do botão do *mouse* quando o seu ponteiro está sobre o componente móvel, diz-se que este componente foi **capturado** pelo *mouse*).

#### 4.3 MOVIMENTO DE VÉRTICES E CONJUNTOS, COM USO DO MOUSE

<b>X, Y</b>	coordenadas do ponto do cursor do <i>mouse</i> , sobre o componente visual, onde o botão é pressionado (EMouseDown) ou arrastado (EMouseMove)
<b>initdrag</b>	booleano que indica se o objeto foi capturado e esta sob o controle do ponteiro do <i>mouse</i>
<b>MouseDown</b>	ponto que indica o local, no componente container, onde o objeto visual foi capturado pelo <i>mouse</i>
<b>MouseMoveNew</b>	ponto que indica a posição do cursor do <i>mouse</i> enquanto está se movimentando sobre o componente visual
<b>Left</b>	Coordenada horizontal esquerda do componente visual capturado pelo <i>mouse</i>
<b>Top</b>	Coordenada vertical superior do componente visual capturado pelo <i>mouse</i>

Quadro 13: Variáveis usadas para o arraste dos componentes visuais com o uso do Mouse.



```
EMouseDown(X, Y){
```

```
initdrag = true;//este novo valor de variável indica que o componente visual foi capturado pelo mouse
```

```
MouseDown = TPoint(X,Y);//o ponto inicial, onde o ponteiro do mouse está, quando o botão do mouse é pressionado para a captura do componente.
```

```
...
```

```
}
```

```
EMouseMove(X, Y){
```

```
if(!initdrag)return;//se o componente visual não foi capturado antes, ele não é movimentado
```

```
MouseMoveNew = TPoint(X,Y);//a partir do momento em que o objeto visual é capturado, toda vez que o ponteiro do mouse é movimentado, esta variável toma a nova posição do cursor do mouse
```

```
//as coordenadas Left (ponto horizontal da extrema esquerda do componente) e Top (ponto vertical de topo do componente visual) são acrescentadas de um valor correspondente à diferença entre: o ponto atual do cursor do mouse em movimento (MouseMoveNew), e o ponto do cursor do mouse estava - nas coordenadas do componente – quando o botão do mouse foi pressionado pela primeira vez (MouseDown).
```

```
Left = Left + MouseMoveNew.x - MouseDown.x;
```

```
Top = Top + MouseMoveNew.y - MouseDown.y;
```

```
...
```

```
}
```

```
EMouseUp{
```

```
...
```

```
initdrag = false;//esta linha apenas indica que o componente foi liberado pelo cursor mouse, e que, se o cursor passar por este componente de novo, este componente não se movimentará junto com o cursor do mouse
```

```
...
```

```
}
```

#### 4.4 SOBRE A CLASSE SUPERARESTA (O CONTÊINER DE INSTÂNCIAS DA CLASSE ARESTA)

Inicialmente, antes desta classe SuperAresta ser projetada, a classe Aresta (Figura 37) era ela própria derivada de uma classe TPaintBox (ver **Erro! Fonte de referência não encontrada.**). Desta forma, os algoritmos orientados a objeto estavam escritos somente para que fosse possível desenhar somente UMA aresta entre seus dois vértices extremos.

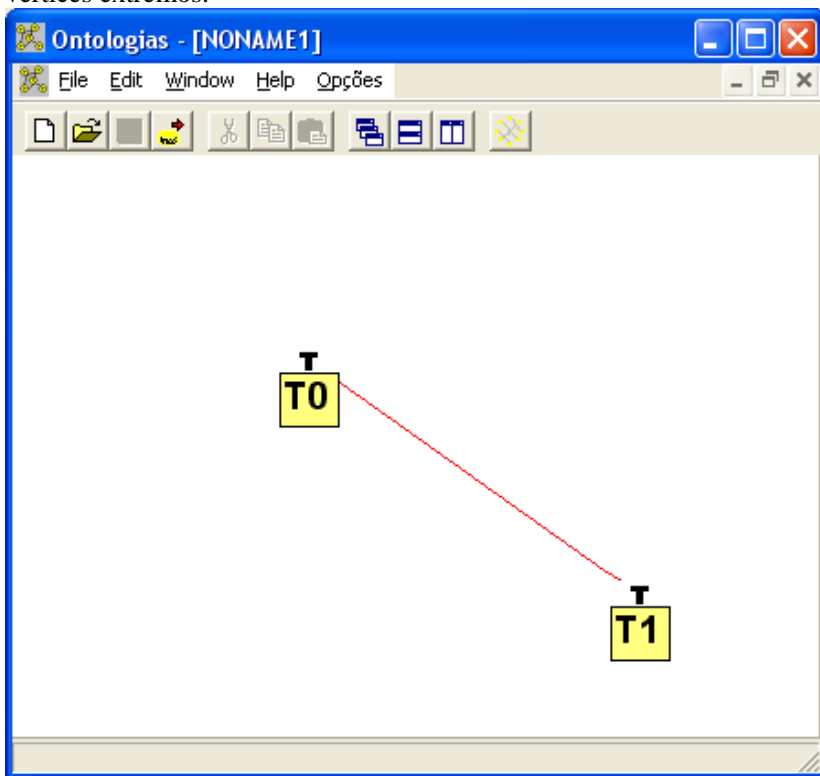


Figura 37: Representação gráfica da primeira versão da classe Aresta.

Posteriormente, surgiu a necessidade de exibição de várias arestas entre dois vértices. Foi decidido, então, que um mesmo contêiner entre dois vértices dados poderia exibir várias arestas. A classe **SuperAresta** foi criada para conter várias instâncias da classe **Aresta**, (Figura 38).

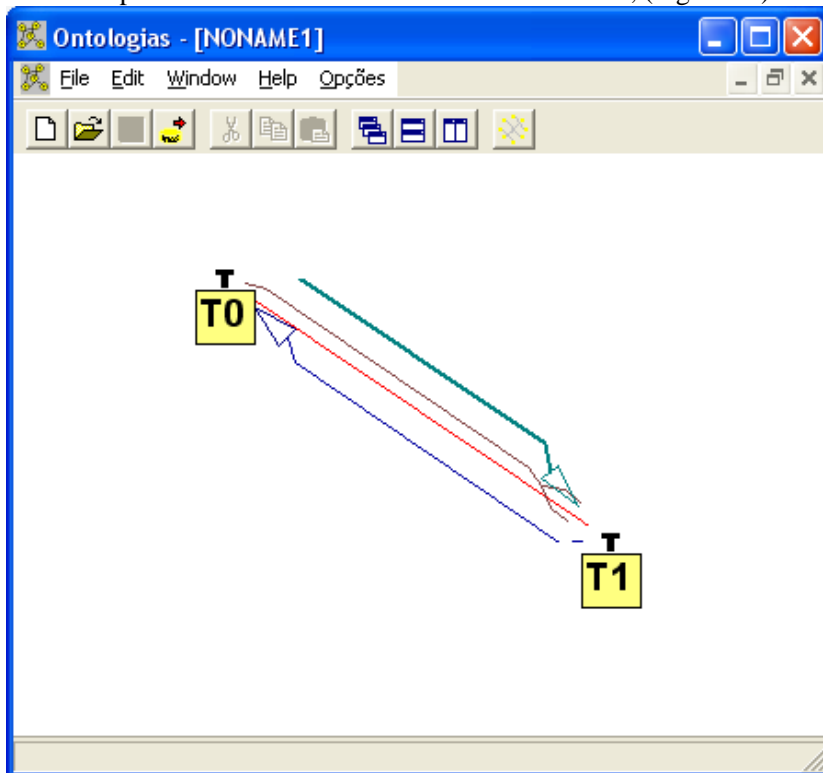


Figura 38: Classe visual SuperAresta, criada para conter várias instâncias da classe geométrica Aresta.

#### 4.5 DIFERENÇAS ENTRE OS SISTEMAS DE COORDENADAS DO MONITOR E DE UMA CLASSE SUPERARESTA

A classe `TPaintBox`, assim como qualquer outro componente visual pertencente à biblioteca *VCL* da *Borland Incorporated*, e, conseqüentemente, suas classes herdadas, é representada como um retângulo contendo quatro coordenadas: *Left* e *Top*, que são as coordenadas do ponto superior esquerdo do componente visual, e *Width* e *Height*, que são as dimensões de largura e altura do componente, respectivamente.

Quanto às coordenadas do vídeo, o ponto de origem da tela está no topo e à extrema esquerda da tela do monitor. A Figura 39 mostra estas coordenadas.

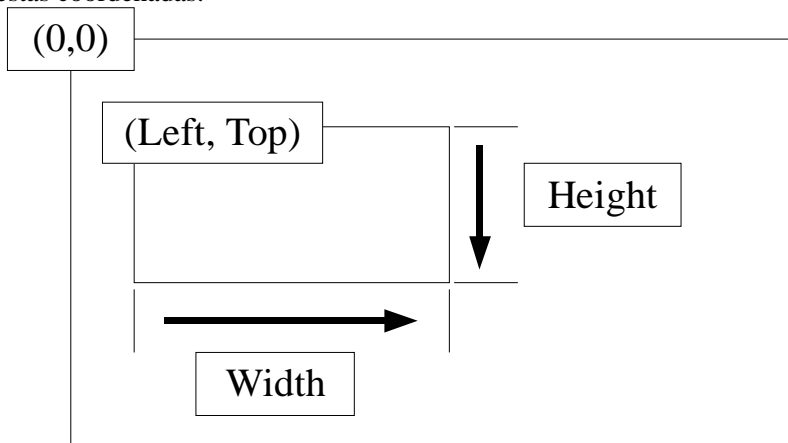


Figura 39: Sistema de coordenadas da tela e de componentes visuais da Borland VCL.

#### 4.6 DESENHO DE UMA LINHA RETA EM UM CONTÊINER DE ARESTAS

O desenho de uma linha reta, que representará a aresta, dentro de uma classe **SuperAresta** (ou **TPaintBox**), deve sempre levar em conta que o sistema de coordenadas deste contêiner gráfico é sempre constante, sendo o ponto (0,0) no ponto superior esquerdo.

Por sua vez, a aresta possui suas coordenadas dependentes das posições dos vértices de origem e de destino. Estas posições são dinâmicas (o usuário final pode manipular os objetos visuais e mudar suas posições relativas), e o sistema de coordenadas da aresta se move junto com as posições relativas dos vértices.

A Figura 40 mostra as nove possibilidades de posicionamento relativo, entre o vértice origem (início da linha com seta), e o vértice destino (fim da linha com seta). Por conveniência, os vértices não foram desenhados, mas somente a aresta ilustrativa.

O quadrado central (número 5) mostra a situação em que os dois vértices tem suas posições idênticas, ou muito próximas. Para este caso, é conveniente – tanto para o usuário, quanto para o algoritmo – que a aresta entre os dois vértices não seja desenhada.

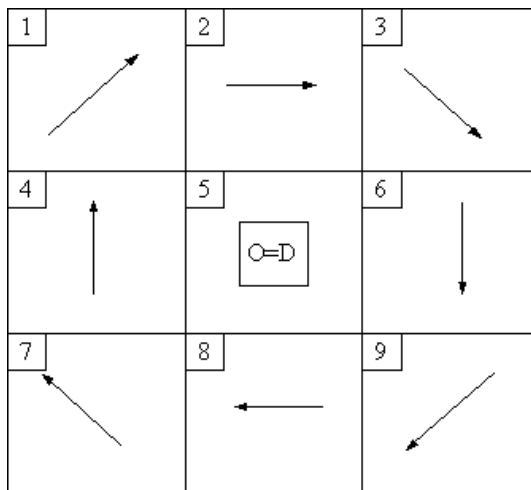


Figura 40: Nove possibilidades para o cálculo das coordenadas de uma aresta.

#### 4.7 ALGORITMO PARCIAL DE DESENHO DE UMA ARESTA ISOLADA

O algoritmo parcial de desenho de uma aresta isolada – dado que seus dois vértices extremos têm suas posições dinâmicas, e que o contêiner onde esta aresta é desenhada tem seu sistema de coordenadas constante – é mostrado nas páginas 117 e 118. Somente a parte de cálculo de coordenadas no eixo horizontal é mostrada, já que a parte de cálculo de coordenadas no eixo vertical é idêntica, mudando apenas as variáveis de eixo horizontal para as variáveis de eixo vertical. No quadro abaixo são mostradas as variáveis relevantes para o algoritmo:

<b>XREF, YREF</b>	Coordenadas do centro do vértice de origem
<b>X, Y</b>	Coordenadas do centro do vértice de destino
<b>tol</b>	Dado que o contêiner vai ter seu retângulo limítrofe adequado às coordenadas da aresta a ser desenhada entre os dois vértices, a variável <i>tol</i> é usada para dar um espaçamento adicional dos pontos extremos da aresta
<b>xi0</b>	Coordenada horizontal, no espaço do contêiner, calculada para o ponto <b>inicial</b> da aresta
<b>xi1</b>	Coordenada horizontal, no espaço do contêiner, calculada para o ponto <b>final</b> da aresta
<b>Left</b>	Coordenada horizontal esquerda do contêiner
<b>Top</b>	Coordenada vertical superior do contêiner
<b>Width</b>	Largura do contêiner
<b>Height</b>	Altura do contêiner

Quadro 14: Variáveis usadas para o desenho de uma aresta isolada.

```
CalcCoord(int XREF, int YREF, int X, int Y){//Os pontos dos vértices de origem e destino são
entrados
```

```
//Se a coordenada horiz. do vértice de destino estiver dentro da distancia de tolerância da
coordenada horizontal do vértice de origem, é considerado que a aresta é uma aresta vertical...
```

```
if((X >= XREF) && (X < XREF + tol)){
```

```
//A coordenada horizontal da SuperAresta (container) é deslocada tol pontos à esquerda do
vértice, como se fosse uma margem de segurança para o desenho completo da Aresta, sem perigos
de corte
```

```
Left = XREF - tol;
```

```
//A largura do container é o dobro da distancia da tolerância
```

```
Width= 2*tol;
```

```
//A coordenada horizontal do ponto inicial da aresta é calculado exatamente no centro horizontal
do container
```

```
xi0 = 0.5 * Width;
```

```
//A coordenada horizontal do ponto final da aresta TAMBÉM é calculado exatamente no centro
horizontal do container
```

```
xi1 = 0.5 * Width;
```

```
}
```

```
else
```

```
//Também neste segundo caso, se a coordenada horizontal do vértice de destino estiver dentro da
distância de tolerância da coordenada horizontal do vértice de origem, os mesmos procedimentos
vistos acima são usados
```

```
if((X < XREF) && (X > XREF - tol)){
```

```
Left = XREF - tol;
```

```
Width= 2*tol;
```

```
xi0 = 0.5 * Width;
```

```
xi1 = 0.5 * Width;
```

```
}
```

```
else
```

*//Em um terceiro caso, se a coordenada horizontal do vértice de destino for **maior** que a coordenada horizontal do vértice de origem, e não estiver dentro da distância de tolerância (se estivesse, haveria executado os procedimentos associados a esta condição)...*

**if(X > XREF){**

*// A coordenada horizontal esquerda do container coincide com o ponto do vértice de **origem***

**Left = XREF;**

*// A largura do container coincide com a distancia da coordenada horizontal do ponto do vértice de origem até a coordenada horizontal do ponto do vértice de destino*

**Width= (X - XREF);**

*//A coordenada horizontal do ponto **inicial** da aresta coincide com a coordenada inicial do container*

**xi0 = 0;**

*//A coordenada horizontal do ponto **final** da aresta coincide com a coordenada na extrema direita do container*

**xi1 = Width - 1;**

**}**

else

*//Em um quarto e ultimo caso, se a coordenada horizontal do vértice de destino for **menor** que a coordenada horizontal do vértice de origem, e não estiver dentro da distância de tolerância (se estivesse, haveria executado os procedimentos associados a esta condição)...**if(X < XREF){***

*// A coordenada horizontal esquerda do container coincide com o ponto do vértice de **destino***

**Left = X;**

*// A largura do container coincide com a distancia da coordenada horizontal do ponto do vértice de destino até a coordenada horizontal do ponto do vértice de origem*

**Width= (XREF - X);**

*//A coordenada horizontal do ponto **inicial** da aresta coincide com a coordenada na extrema direita do container*

**xi0 = Width - 1;**

*//A coordenada horizontal do ponto **final** da aresta coincide com a coordenada inicial do container*

**xi1 = 0;**

**}**



#### 4.8 EIXO DETERMINADO PELO SEGMENTO DE RETA ENTRE OS VÉRTICES INICIAL E FINAL

Dado que todas as arestas serão desenhadas dentro do contêiner determinado pela classe SuperAresta, e dado que estão determinados, a partir do algoritmo anterior, seus pontos de início e fim do eixo de referencia a partir do qual serão calculados os pontos de todas as arestas contidas nesta classe, e também dados os dois vértices extremos destas arestas  $\mathbf{Vo}(X_o, Y_o)$  e  $\mathbf{Vd}(X_d, Y_d)$ , temos a representação gráfica da Figura 41:

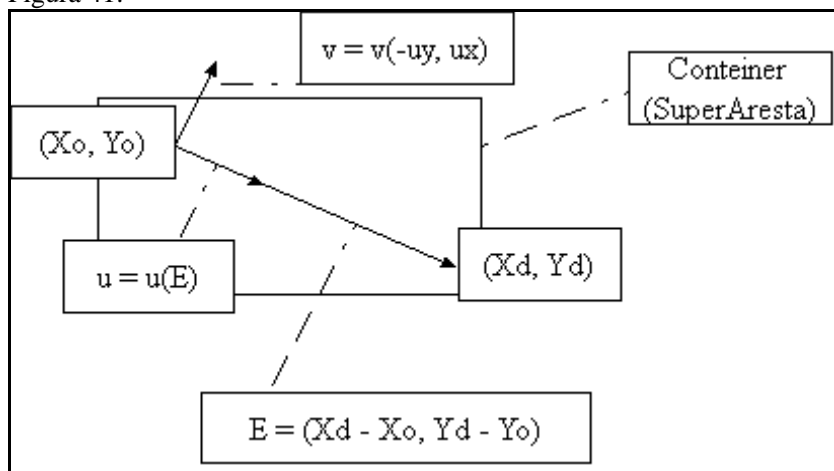


Figura 41: Representação gráfica do cálculo das coordenadas cartesianas tendo como base o eixo da aresta entre dois vértices.

A partir daí temos o vetor eixo principal da aresta, dado por:

$$\vec{E} = E (X_d - X_o, Y_d - Y_o)$$

**Equação 1: Vetor eixo principal da aresta.**

E seu módulo  $mE$  dado por:

$$mE = |\vec{E}| = \sqrt{(X_d - X_o)^2 + (Y_d - Y_o)^2}$$

**Equação 2: Módulo do vetor eixo principal da aresta.**

Temos então o vetor unitário do eixo formado entre os dois vértices:

$$\vec{u} = \frac{\vec{E}}{\|\vec{E}\|}$$

**Equação 3: Vetor unitário do eixo principal da aresta.**

E se aplicarmos a regra da mão direita sobre este vetor unitário, ou um operador de giro de  $90^\circ$  no sentido anti-horário sobre este vetor normal, temos o segundo eixo do espaço vetorial baseado neste eixo entre os dois vértices:

$$\vec{v} = v\vec{E} \left( -u\vec{E}_y, u\vec{E}_x \right)$$

**Equação 4: O eixo vertical do espaço determinado pela aresta entre os dois vértices.**

Dessa forma, caso o analista deseje determinar a posição de um ponto  $P_0(X_0, Y_0)$  no sistema de coordenadas do eixo formado entre os dois vértices, as coordenadas deste ponto serão combinadas com os vetores unitários como na Equação 5, gerando o ponto  $Q(X_q, Y_q)$ , no sistema de coordenadas do contêiner SuperAresta:

$$Q(X_q, Y_q) = (X_o, Y_o) + X_0 \cdot \vec{u} + Y_0 \cdot \vec{v}$$

**Equação 5: Determinação do ponto Q, no sistema de coordenadas da SuperAresta contêiner, dado o ponto  $P_0(X_0, Y_0)$ , no espaço do eixo determinado pela Aresta.**

Sendo mostrado como na Figura 42:

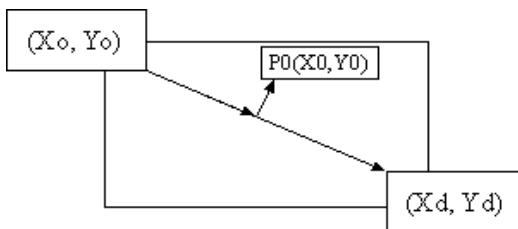


Figura 42: Determinação das coordenadas de um ponto no espaço do eixo entre os dois vértices.

#### 4.9 DESENHO DE UMA POLILINHA EM UM CONTÊINER DE ARESTAS

Dado que uma polilinha é um conjunto de pontos  $P(X_i, Y_i)$  no sistema de coordenadas do eixo formado entre os dois vértices, para que esta polilinha seja desenhada na SuperAresta contêiner, seus pontos  $P(X_i, Y_i)$  devem ser transformados para pontos  $Q(X_i, Y_i)$  do sistema de coordenadas da SuperAresta, tal como é mostrado na Equação 6:

$$\begin{aligned} \vec{Q}(X_0, Y_0) &= (X_o, Y_o) + X_0 \cdot \vec{u} + Y_0 \cdot \vec{v} \\ \vec{Q}(X_1, Y_1) &= (X_o, Y_o) + X_1 \cdot \vec{u} + Y_1 \cdot \vec{v} \\ \vec{Q}(X_2, Y_2) &= (X_o, Y_o) + X_2 \cdot \vec{u} + Y_2 \cdot \vec{v} \\ &\dots \\ \vec{Q}(X_n, Y_n) &= (X_o, Y_o) + X_n \cdot \vec{u} + Y_n \cdot \vec{v} \end{aligned}$$

**Equação 6:** Lista de equações para cada ponto  $Q(X, Y)$  no espaço de um contêiner SuperAresta, dadas as coordenadas dos pontos da Aresta, no espaço determinado pelo eixo entre  $V_o$  e  $V_d$ .

A Figura 43 mostra um exemplo gráfico de como somente uma polilinha de uma Aresta seria desenhada em um contêiner **SuperAresta**, dados os seus pontos dispostos no espaço determinado pelos vetores de origem e destino. Aqui pode-se notar que o ponto  $P_0(X_0, Y_0)$  será desenhado fora dos limites do contêiner **SuperAresta**, dando uma impressão de que parte da polilinha foi cortada:

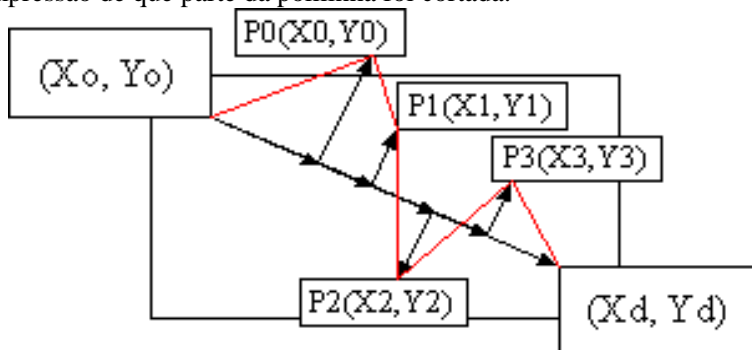


Figura 43: Determinação das coordenadas dos pontos de uma polilinha no espaço do eixo entre os dois vértices.

#### 4.10 O DESENHO DA GEOMETRIA DA EXTREMIDADE DE UMA ARESTA

Tal como no desenho da aresta, a extremidade de uma aresta é desenhada de forma vetorial, através de uma seqüência de pontos. Dado que a seqüência de pontos que determina seu tracejado está disposta em um sistema bidimensional de coordenadas, este sistema de coordenadas deve ser determinado.

Antes da determinação definitiva do sistema de coordenadas, um eixo horizontal deste sistema deve ser determinado. Como escolha inicial, foi definido que o eixo horizontal deste sistema seria o eixo formado pela linha imaginária entre o ponto final da aresta e o ponto do centro do vértice da extremidade correspondente desta aresta.

A Figura 44 mostra a representação gráfica do eixo horizontal, para o sistema de coordenadas de uma das extremidades de uma aresta, dados os pontos da extremidade da aresta, e do centro do vértice:

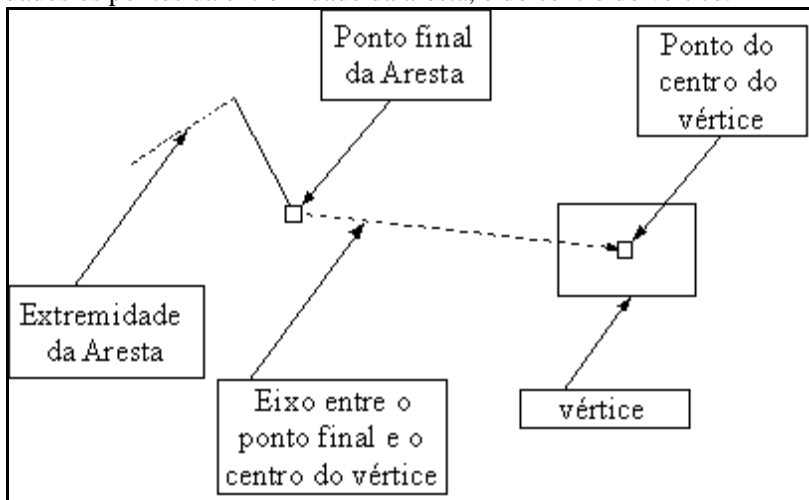


Figura 44: Elementos formadores do eixo principal da extremidade da aresta.

A Figura 45 mostra um diagrama de visão do sistema de coordenadas formado entre o ponto da extremidade da aresta, e o ponto central do vértice. O eixo vertical é formado a partir da aplicação da regra da mão direita sobre o eixo horizontal.

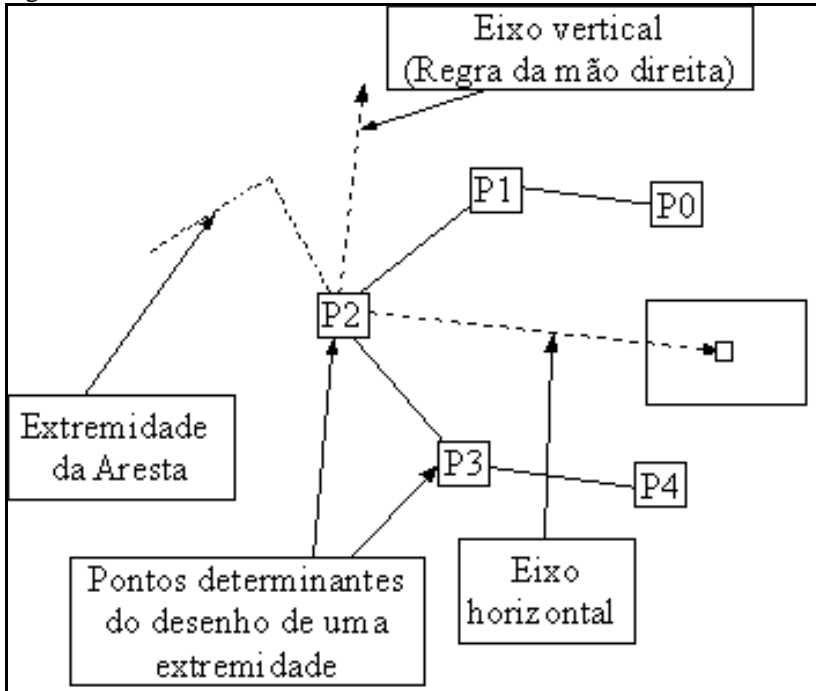


Figura 45: Posicionamento de pontos do desenho da extremidade de uma aresta.

A extremidade exemplo da aresta, com segmentos de reta com pontos **P0**, ..., **P4** são posicionados de acordo com o sistema de coordenadas da Figura 45.

Como equação básica para formação do sistema de coordenadas da extremidade da aresta, dados o ponto **P0** de início/fim da aresta desenhada, e o ponto **Pd** de centro do vértice origem/destino da SuperAresta, tem-se então:

$$\vec{D} = \vec{Pd} - \vec{P0}$$

**Equação 7: Vetor do eixo formado entre P0 e Pd.**

$$d = |\vec{D}|$$

**Equação 8: Módulo do vetor D, contido na variável d.**

$$u\vec{D} = \frac{\vec{D}}{d} = (dx, dy)$$

**Equação 9: Coordenadas dx, dy do vetor unitário uD do eixo formado entre os vértices de origem e destino.**

O algoritmo de posicionamento dos pontos das extremidades das arestas é mostrado no quadro abaixo:

<b>EhExtrOrigem</b>	Variável booleana indicando se a equação de cálculo de posição dos pontos que desenham a extremidade da aresta está na extremidade de início ou na extremidade de fim desta aresta
<b>x, y</b>	Coordenadas horizontal e vertical, no espaço da extremidade, do ponto atual a ser posicionado na classe SuperAresta
<b>xi0, yi0</b>	Coordenadas horizontal e vertical do ponto inicial da Aresta, desenhada dentro da classe SuperAresta
<b>dx, dy</b>	Coordenadas do vetor unitário que determina a direção da aresta, entre seus vértices de origem e de destino
<b>eeE</b>	Deslocamento, no sentido de fora para dentro da aresta, do ponto (0,0) do sistema de coordenadas onde será posicionada a geometria da extremidade desta aresta
<b>p</b>	Ponto que receberá as coordenadas, inicialmente x,y, vistas acima, no espaço da classe SuperAresta

Quadro 15: Variáveis usadas para o cálculo dos pontos das extremidades das arestas.

```
DetPontoExtr(float x, float y, TPoint &p, bool EhExtrOrigem, int eeE)  
{  
if(EhExtrOrigem){ //se a extremidade é de origem...  
p.x = xi0 + dx * (0 + eeE - x) - dy * y;  
p.y = yi0 + dy * (0 + eeE - x) + dx * y;  
return;  
}  
p.x = xi0 + dx * (d - eeE + x) - dy * y;  
p.y = yi0 + dy * (d - eeE + x) + dx * y;  
return;  
}
```

#### 4.11 DETERMINAÇÃO DA ENVOLTÓRIA

Inicialmente, por simplicidade na criação dos algoritmos, para a determinação da envoltória que representa um domínio ou um conjunto de vértices, deve-se considerar que esta envoltória deve possuir as seguintes propriedades: a) Esta envoltória é convexa; b) Esta envoltória é uma composição contínua de segmentos de retas; c) TODOS os vértices que estão contidos no conjunto determinado por esta envoltória devem estar dentro dos limites gráficos determinados por seus segmentos de reta

Os algoritmos de determinação dos pontos das envoltórias foram construídos com grau crescente de dificuldade. Inicialmente foi construído o algoritmo de envoltória quadrada. Posteriormente, sob o mesmo princípio, foi construído o algoritmo de envoltória octogonal.

Apesar de um algoritmo de desenho de envoltórias com o formato intermediário a estes dois tipos básicos ainda não estar implementado, é perfeitamente possível fazê-lo.

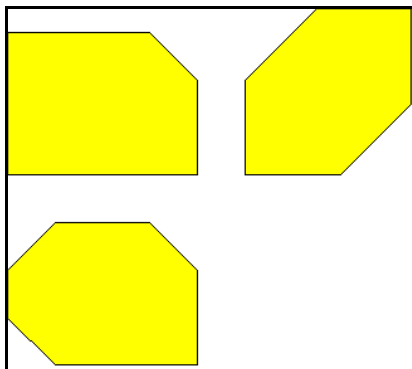


Figura 46: Envoltórias com formato intermediário entre quadrado e octogonal.



## 4.12 PRINCÍPIO BÁSICO DE CONSTRUÇÃO: A ENVOLTÓRIA QUADRADA

Para a construção de uma envoltória, o sistema usa um algoritmo de procura de valores máximos e mínimos, nas coordenadas horizontal e vertical do espaço onde as envoltórias estão desenhadas.

Para o caso da coordenada horizontal, o sistema captura os valores máximo e mínimo desta coordenada, respectivamente em duas variáveis **Xmin** e **Xmax**. Idem para os valores máximo e mínimo da coordenada vertical: **Ymin** e **Ymax**.

Dessa forma, é possível determinar os segmentos de reta que compõem a envoltória (linhas em cor azul, da Figura 47).

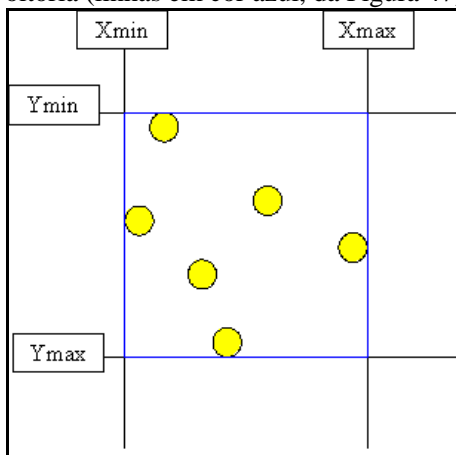


Figura 47: Princípio das equações para os segmentos de reta da envoltória quadrada.

#### 4.13 ENVOLTÓRIA OCTOGONAL

Semelhante ao processo de geração da envoltória quadrada, a geração da envoltória octogonal está baseada na procura de oito equações de reta. As oito equações de linha reta estão abaixo, onde C0...C7 são constantes, encontradas com os algoritmos de procura de vértices:

1.  $x_{\min} = c0$
2.  $x_{\max} = c1$
3.  $y_{\min} = c2$
4.  $y_{\max} = c3$
5.  $(x+y)_{\min} = c4$
6.  $(x+y)_{\max} = c5$
7.  $(x-y)_{\min} = c6$
8.  $(x-y)_{\max} = c7$

**Equação 10: Representação das linhas retas que formarão os segmentos da envoltória octogonal.**

O grupo da Equação 10 representa as seguintes retas da envoltória octogonal (Figura 48):

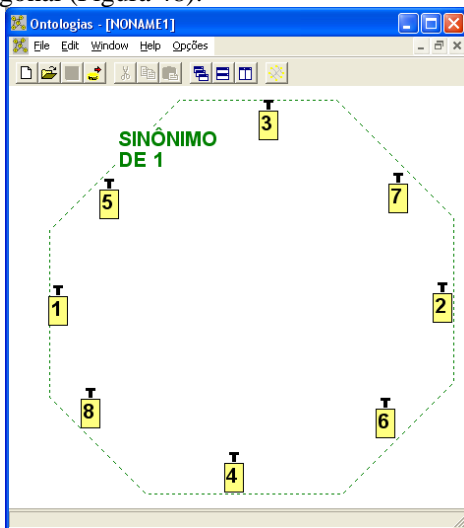


Figura 48: Segmentos de reta da envoltória octogonal, formados pelo cruzamento das retas do grupo da Equação 10.

## 5 RESULTADOS DA APLICAÇÃO DO MÉTODO

### 5.1 PRIMEIRA VERSÃO DO EDITOR DE ONTOLOGIAS 2D<sup>9</sup>

A primeira versão do editor de ontologias caracteriza-se por representar tanto relações semânticas entre termos quanto relações de pertinência de termos a domínios na forma de arestas.

A Figura 49 (página 130) demonstra a diferença entre relações entre termos e as relações de pertinência a domínio da seguinte forma: para as relações de pertinência a domínio, muitos vértices-termo apontam para somente um vértice que representa o nome de um domínio. Para as relações semânticas normais, a sub-rede de vértices se comporta como uma clique completa de arestas de mesma cor (para o caso de relações de sinônimos), ou uma clique incompleta com arestas de cores diferentes (relações de meronímia/hiperonímia e ocorrência conjunta de termos em documentos).

---

<sup>9</sup> Esta versão foi descrita por Stradiotto *et. al.* 2006



## 5.2 SEGUNDA VERSÃO DO EDITOR DE ONTOLOGIA 2D

### 5.2.1 Prototipagem das formas visuais de representação e suas combinações

Inicialmente, foram selecionados e combinados os elementos visuais para representar as ontologias. Na etapa seguinte, recorreu-se à experimentação, em que cada problema foi isolado, usando-se técnicas de prototipagem. Para isso, vários programas de pequeno porte, de nível experimental, foram feitos para representar cada uma das soluções implementadas: vértices e seus elementos anexados; arestas e seus elementos anexados; múltiplas arestas entre dois vértices; extremidades de arestas; envoltórias para a representação de conjuntos.

Abaixo são apresentados alguns resultados visuais em protótipos, de funcionalidades necessárias<sup>10</sup> para a implementação do aplicativo de edição de ontologias e forma de RRC e, conseqüentemente, da *framework*.

Neste experimento, foi utilizada a base de dados da ontologia do Sistema Ontojuris, que está armazenado de duas formas: em arquivos Excel e em bases de dados Postgres.

#### 5.2.1.1 Conexão com fontes de dados Excel

Um dos artifícios programados no editor de ontologias 2D foi a ferramenta de importação de termos, relações semânticas e domínios, a partir dos textos contidos nas células de arquivos Excel.

O princípio do processo baseia-se na leitura de arquivos Excel como uma fonte de dados ODBC. O processo de importação funciona assim:

1. O usuário escolhe o arquivo do qual faria a importação;
2. O usuário selecionava um grupo de células;
3. O usuário determina o papel dos textos contidos nestas células (se serão termos, domínios e subdomínios);
4. O sistema mostra as células pintadas com as cores correspondentes dos papéis determinados pelo usuário (Na Figura

---

<sup>10</sup> Embora os elementos anexados às arestas terem sido implementados em protótipos, eles não foram usados no sistema atual de edição de ontologias.

50, por exemplo, o domínio é mostrado em vermelho, o subdomínio é mostrado em roxo, o termo principal é mostrado em amarelo, e os termos relacionados a este termo principal, com relação de aparição conjunta (conexo) são mostrados em azul);

5. Após a seleção dos papéis, o usuário requisita a importação das relações para uma janela, em que será mostrada a instância da ontologia recém construída (Figura 50).

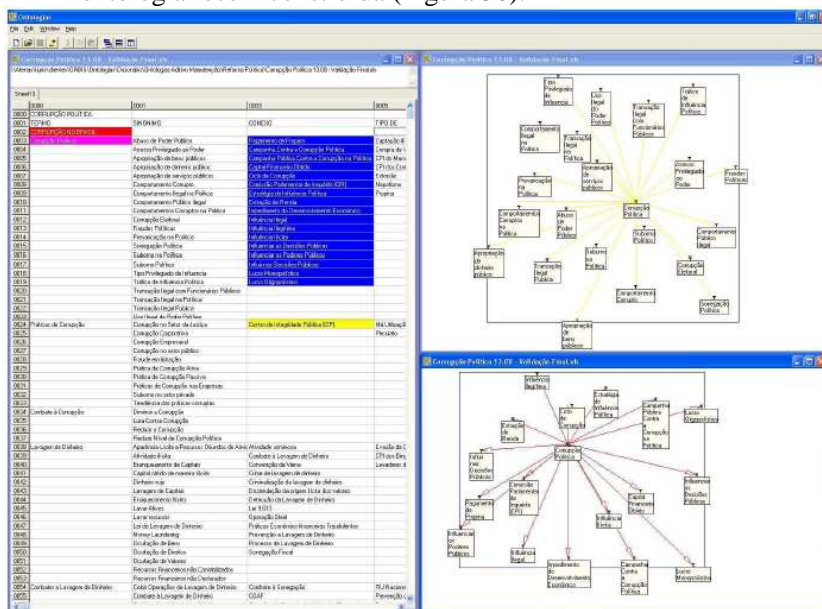


Figura 50: Aplicativo de construção de ontologias 2D, com o processo de importação de arquivos Microsoft Excel.

### 5.2.1.2 Comunicação com bases Postgres, com dados criptografados

Na experiência de comunicar o editor de ontologias com o Sistema de Gestão de Conhecimento (SGC), houve uma pesquisa de comunicação do C++ Builder com o Java, devido ao fato de que os dados do SGC estarem criptografados com uma das variantes do algoritmo Blowfish, combinada com mais uma camada desconhecida de criptografia.

Se fosse somente o algoritmo Blowfish, as chances de usar o mesmo algoritmo no Borland C++ Builder ou Delphi seriam bem maiores, pois o algoritmo está disponível nessas linguagens, na internet. O algoritmo usado para o SGC possui uma variante, usada pelo Java





### 5.2.1.4 Envoltórias

Envoltórias (Figura 52) são linhas desenhadas ao redor de vértices, arestas ou mesmo outras envoltórias. Esses elementos geométricos servem para representar conjuntos destes elementos; para tal, visualmente, devem ter seu desenho sempre feito ao redor de seu conteúdo.

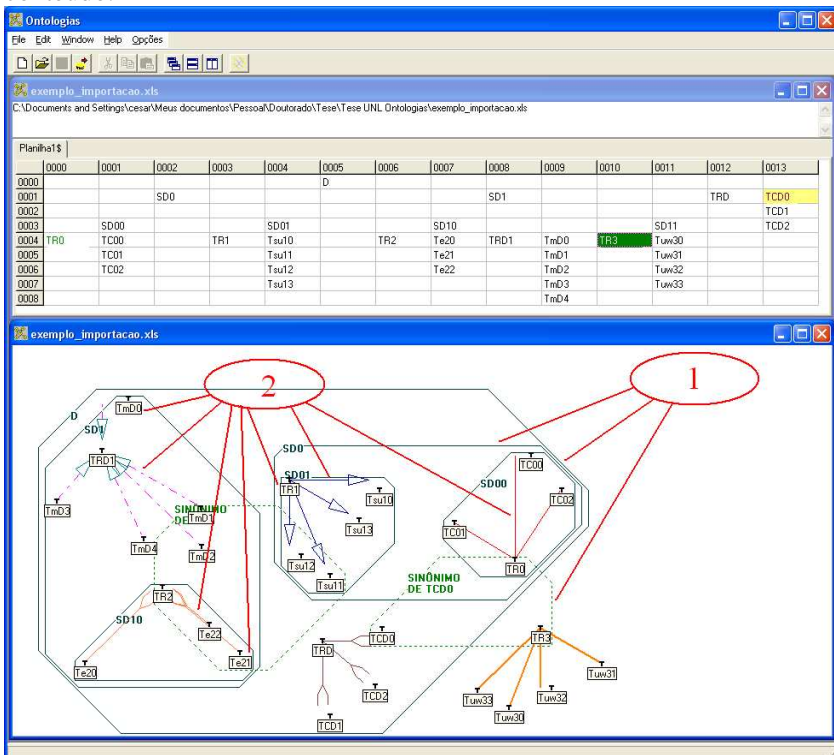


Figura 52: Demonstração de formação de conjuntos de vértices e arestas a partir de arquivos Microsoft Excel.

### 5.2.1.5 Múltiplas arestas entre dois vértices

É conveniente, para muitos sistemas de visão de informações, que exista uma representação de múltiplas ligações ou fluxos entre dois vértices. A *framework* é capaz de representar várias arestas entre dois vértices (Figura 53), de forma que seja possível ver, de um princípio, e caso desejado, todas as ligações existentes entre dois vértices apresentados. A vantagem desta forma de visualização é a disponibilidade imediata da visão de todos os elementos ocorridos ou



que ligam os dois vértices. Um segundo elemento de representação (referente às arestas) é a possibilidade de alterar a geometria de suas extremidades. Desta forma, há um aumento significativo de possibilidades de representação de informações referentes a conexões, fluxos ou causas entre dois elementos isolados.

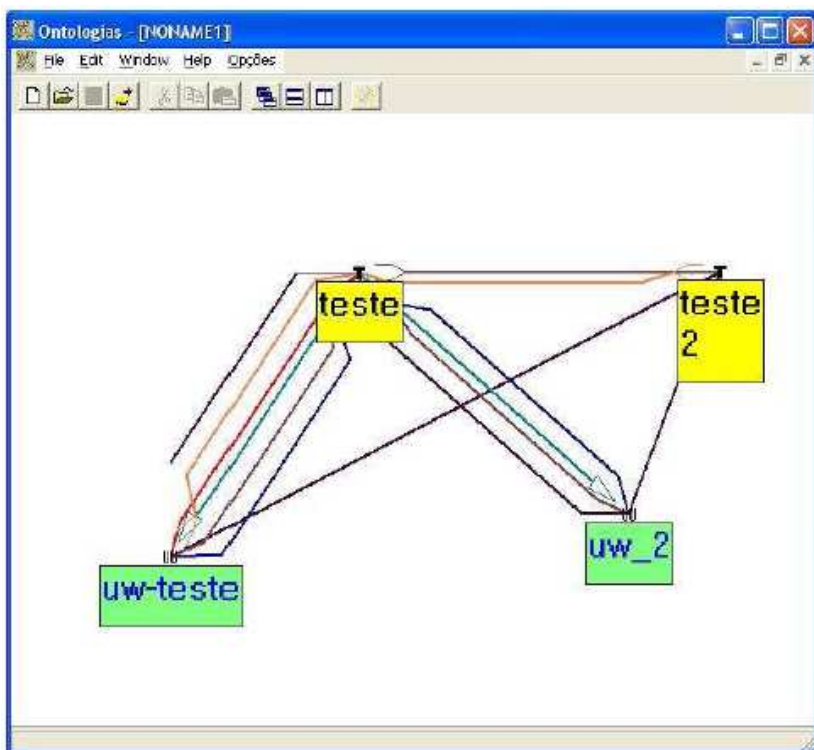


Figura 53: Protótipo com demonstração de múltiplas arestas entre vértices.

### 5.2.2 O protótipo da segunda versão do editor de ontologias

Este item descreve a implementação de um sistema de construção de ontologias para um sistema multilíngüe de busca de documentos. O sistema possui artifícios de interface para edição dos componentes visuais propostos, baseados protótipos vistos no item 5.2.1.

O modelo matemático permite a inclusão de várias envoltórias dentro de uma envoltória mãe e a inclusão de uma envoltória dentro de várias envoltórias (relacionamento de 1 para N e N para 1) (Figura 54). O modelo permite a inclusão de arestas dentro de envoltórias (a representação da pertinência de uma aresta a uma envoltória é feita

mostrando a envoltória ao redor dos vértices que estão nas extremidades da aresta contida).

Permite também representar palavras universais do projeto UNL e associá-las aos termos do modelo de ontologias de Bueno (2008) (Figuras Figura 54, Figura 55 e Figura 56). Para que os dados sejam guardados em disco, a interface gráfica está associada a uma base de dados com a estrutura proposta (modelo de domínio) explicada na Figura 34, na página 97.

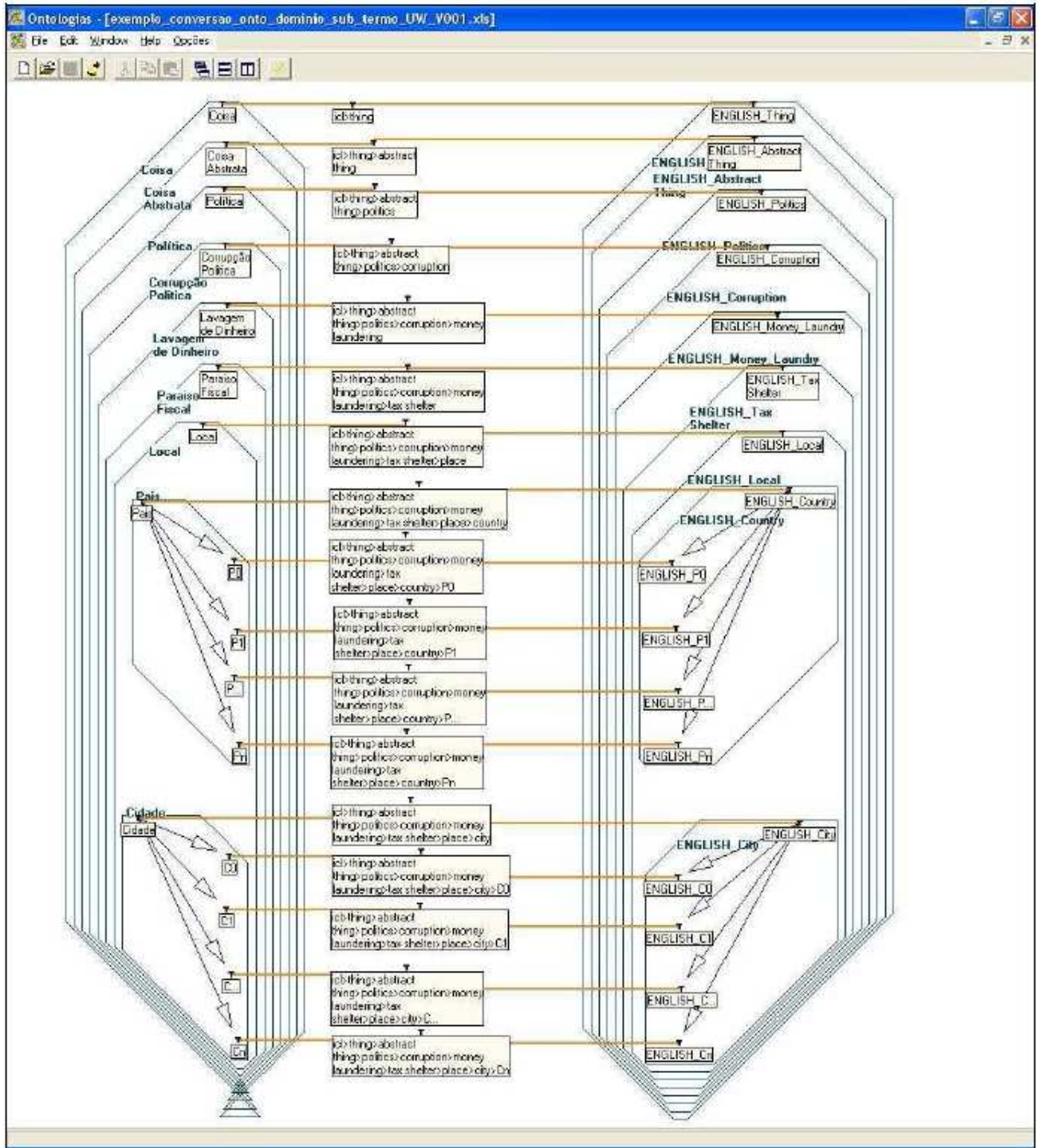


Figura 54: Interface do editor mostrando uma janela com vários níveis de domínios.

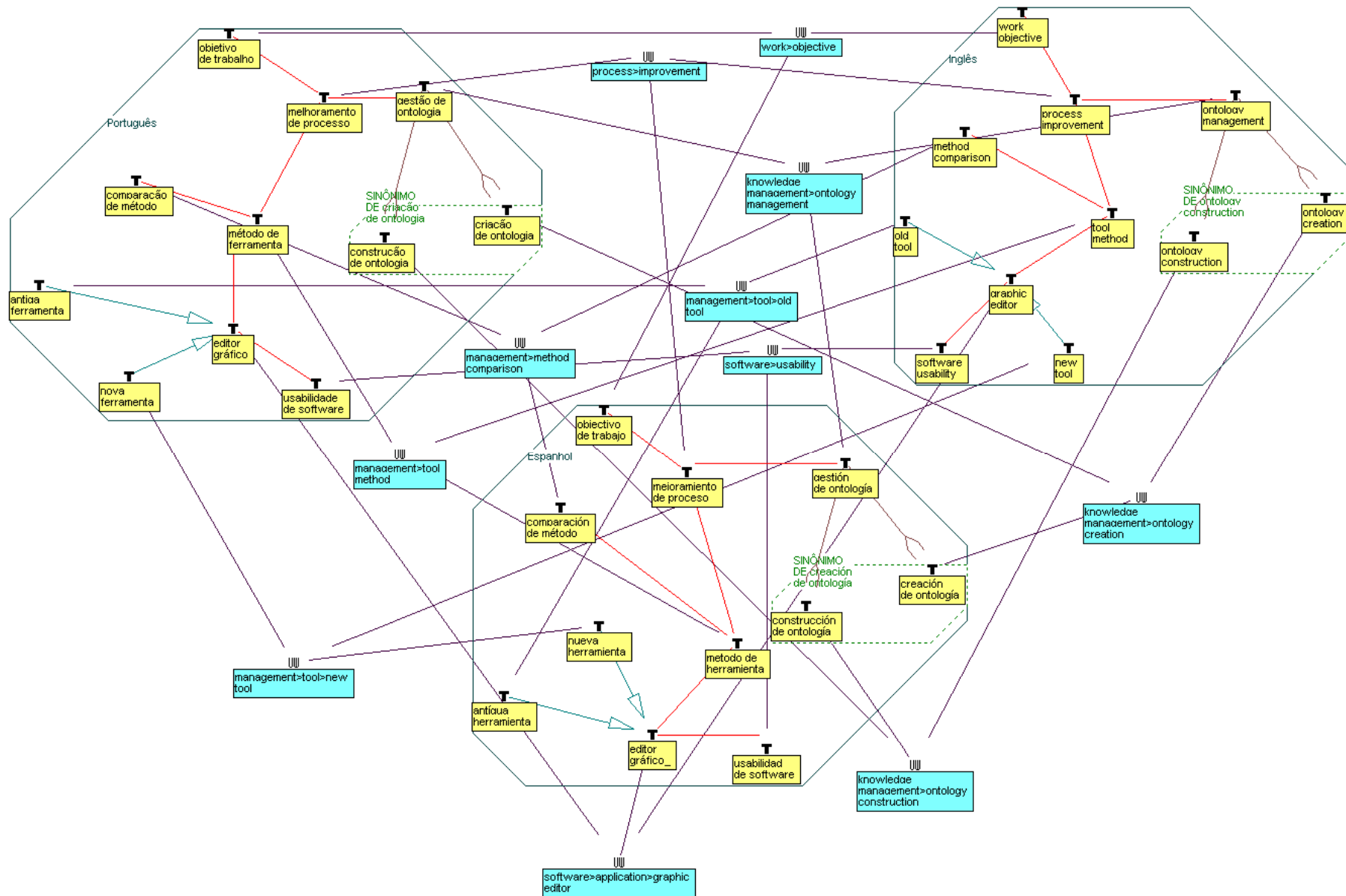


Figura 55: Interface de uma das janelas do editor de ontologias multilíngües, mostrando os termos em três idiomas (Português, Espanhol e Inglês, em amarelo), palavras universais (em azul), as relações e as envoltórias representando os domínios.

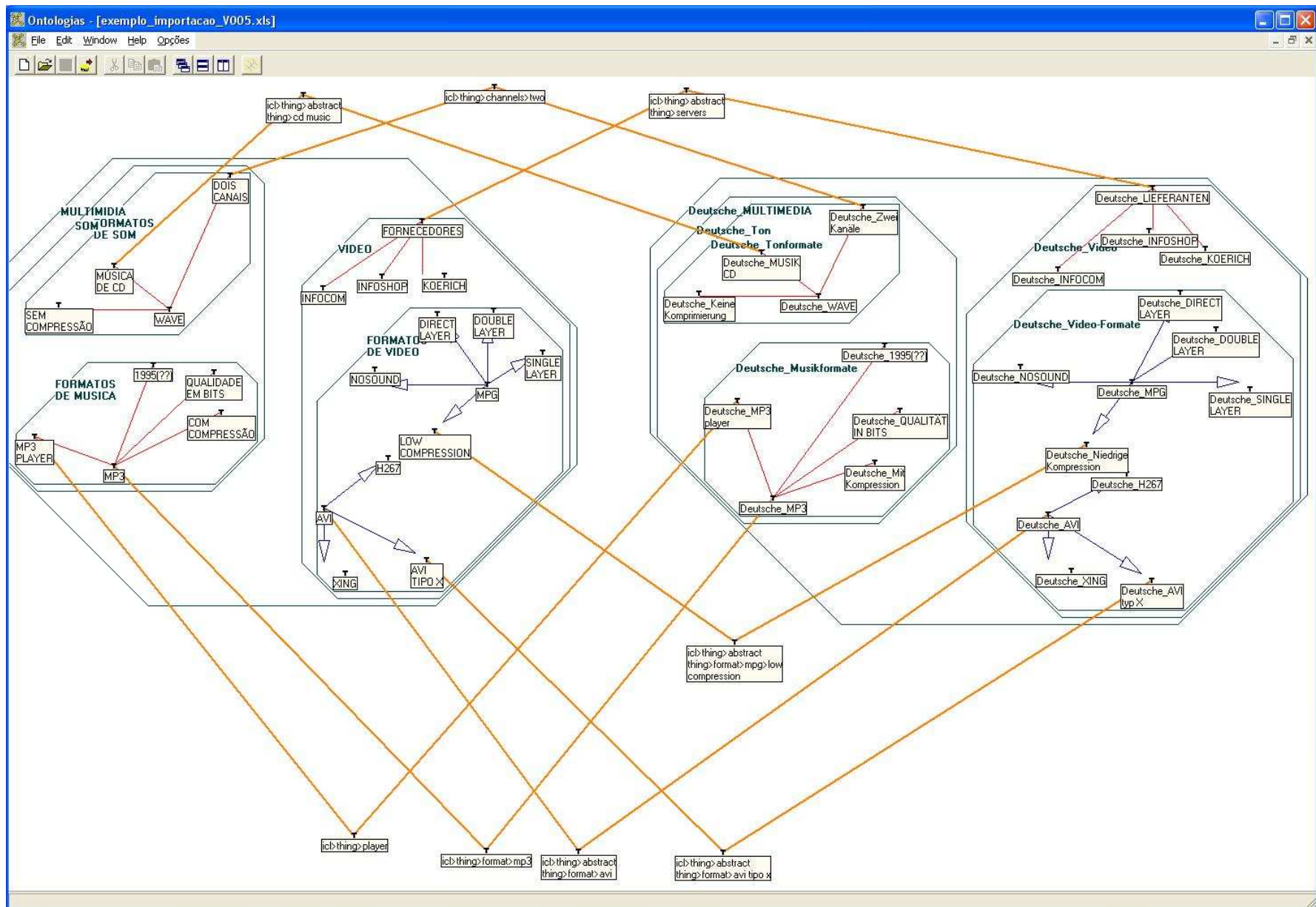


Figura 56: Interface exemplificando termos (conectados por palavras universais) dentro de domínios, em vários níveis.

## 5.3 TERCEIRA VERSÃO DO EDITOR DE ONTOLOGIAS: VISÃO 3D

### 5.3.1 Prototipagem das formas visuais de representação e suas combinações

#### 5.3.1.1 Elementos básicos

Para explicar os experimentos que confirmaram o funcionamento de cada uma das etapas na construção do editor, são apresentados os sete itens que fazem parte deste método:

1. O sistema de coordenadas 3D como tuplas de três elementos ( $x$ ,  $y$ ,  $z$ ): é o sistema cartesiano de coordenadas, representando, respectivamente, o eixo horizontal, positivo à direita, o eixo vertical, positivo para cima, e de profundidade, positivo indo em direção à câmera. Todos os pontos que fazem parte dos objetos são posicionados com base neste sistema;
2. O espaço vazio, com uma cor de fundo determinada (inicialmente de cor preta): a existência deste espaço é a primeira garantia de que todas as operações computacionais para o surgimento do espaço 3D foram feitas de forma correta. Caso este espaço não apareça, significa que alguma instrução está faltando, está com um parâmetro errado, ou está em ordem invertida;
3. As coordenadas de origem do espaço (posição  $(0, 0, 0)$ );
4. Um objeto na origem (ponto  $(0, 0, 0)$ ) do sistema de coordenadas: nos testes iniciais, este objeto ajuda a ver o comportamento do experimento, com variações na posição e direção do observador, e com variações na posição e direção do objeto observado. É muito comum, nos experimentos, determinar uma posição para o objeto e para a câmera, e como resultado aparecer somente o fundo escuro, representando o espaço do ambiente. Caso esse fenômeno ocorra, deve ser feito um diagnóstico sobre a direção inicial da câmera, a existência ou não de luz, para que seja refletida pelo objeto, ou se o objeto tem uma cor que não reflete luz (cor preta), ou ainda se o objeto está posicionado fora da pirâmide de visão determinada para a câmera;

5. A câmera, posicionada sobre o eixo de profundidade positiva (0, 0, 100), e direcionada com o vetor inverso (0, 0, -100) (que, para efeitos de direção, equivalente ao vetor normalizado (0, 0, -1));
6. A luz ambiente, sendo um vetor de eixo qualquer, ou que acompanhe a orientação da câmera: caso a luz tenha direção e sentido constantes, no momento em que o usuário navega no ambiente, este usuário percebe que os objetos sempre refletem a luz de um mesmo lado, independente da direção em que a câmera esteja. Existe a possibilidade de o usuário não ver um objeto se estiver com a câmera apontando para o lado do objeto que está a favor da luz incidente. Caso a luz acompanhe a câmera, qualquer objeto que este usuário veja de frente estará iluminado;
7. A navegação no ambiente. Esta navegação abrange três operadores básicos: Translação do Ambiente, que é o deslocamento de todos os objetos ao longo de qualquer um dos três eixos; Rotação do Ambiente, que é o giro de todos os objetos do ambiente ao redor de qualquer um dos três eixos básicos; e a Escala (aqui também chamado de Aproximação, ou *Zoom*) dos objetos do ambiente, resultando em um efeito de aproximação ou afastamento dos objetos da câmera. Vale notar que essas três formas de operação podem ser aplicadas tanto nos objetos do ambiente quanto na câmera do observador: Exemplo: o efeito de girar o conjunto de objetos no sentido horário é equivalente a girar a câmera no sentido anti-horário; o efeito de mover o conjunto de objetos para a direita é equivalente a mover a câmera para a esquerda; e o efeito de deixar a câmera parada e afastar o conjunto de objetos é equivalente a deixar o conjunto de objetos parado e afastar a câmera.

### *5.3.1.2 Teste de visão de ambientes 3D com Java 3D em múltiplos documentos (MDI)*

Até meados de 2008, a sobreposição de janelas com um universo 3D dentro apresentava um problema: Quando uma janela interna com um ambiente 3D era aberta, ao mudar o tamanho da janela ou mover a janela pela sua borda superior, o retângulo onde estava o ambiente 3D permanecia parado, e a janela mudava de tamanho ou de posição, dando um efeito antiestético. A Figura 57 mostra um exemplo no qual o ambiente 3D – contendo um livro – sobrepõe um menu, apontado pelo cursor do mouse.





Figura 57: Efeito do componente *heavyweight* do Java 3D sobre componentes *lightweight* do Swing. (Sobreposição do ambiente 3D sobre o menu do aplicativo).<sup>11</sup>

A atual versão do Java (JRE 6 Update 12) corrigiu este problema. A Figura 58 demonstra uma aplicação, feita em Java 3D, na qual existem várias janelas internas, e cada janela mostra uma esfera (exemplo de ambiente tri-dimensional). Caso o problema de sobreposição persistisse, os ambientes 3D sobreporiam às bordas das janelas.

Quanto ao uso de texturas e localização de objetos no espaço, a mesma Figura 58 mostrou exemplos bem comportados de esferas em uma posição fixa e sem rotação de câmera.

Já a Figura 59 apresenta os resultados de um primeiro teste de navegação por rotação, de um quadrado com uma textura qualquer. Este experimento inicial teve por objetivo resolver cinco questões: 1) Garantia do posicionamento da câmera (também chamada de observador); 2) Efeito inicial dos primeiros experimentos sobre operadores de navegação (translação, rotação e aproximação); 3) Efeito da aplicação de textura quadrada sobre uma face também quadrada; 4) Amostragem do lado visível de uma face; 5) Aplicação da luz sobre um objeto.

<sup>11</sup> Fonte: <http://codeidol.com/java/swing/Rendering/Add-a-Third-Dimension-to-Swing/>



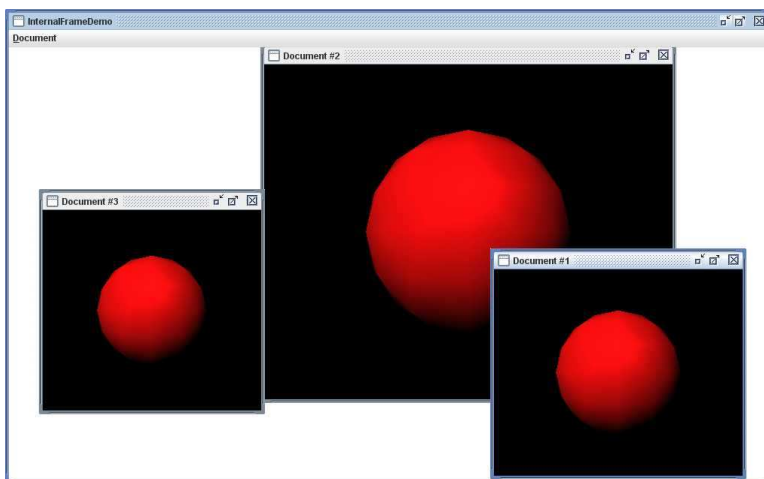


Figura 58: Imagem de protótipo para teste de sobreposição de várias janelas contendo seus respectivos ambientes 3D.

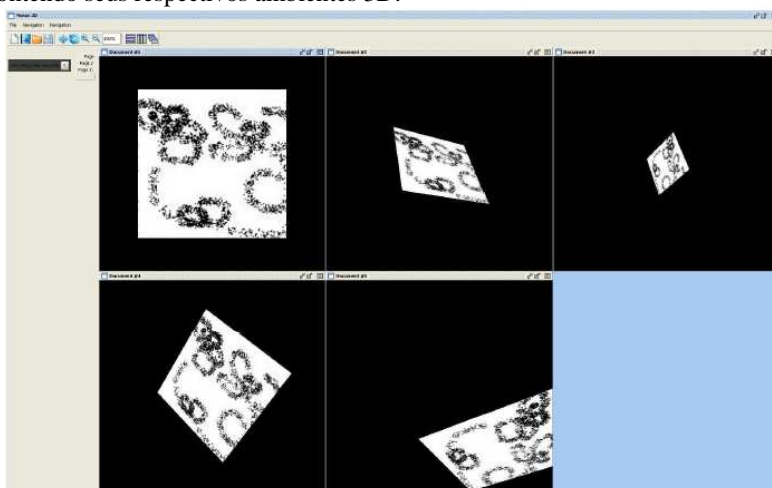


Figura 59: Resultados do segundo experimento, sobre posicionamento de câmera, navegação, textura, lado visível de uma face, e aplicação de Luz.

### 5.3.1.3 Testes de desempenho para vértices

Uma instância de ontologia multilíngüe, vista como uma rede de relacionamento e envoltórias e com objetos 3D associados, exige a amostragem de uma grande quantidade de objetos. Usando a biblioteca Java 3D para mostrar esses objetos, foram feitos experimentos no qual se empregam apenas quadrados de tamanho fixo, em grande quantidade.

A máquina utilizada possuía 1.5 GB de memória RAM. O teste consistiu em aumentar o número de quadrados a mostrar até que a Máquina Virtual Java (JVM) apontasse falta de memória.

A Figura 60 apresenta o resultado de uma amostragem de  $2 \times 320 \times 320 = 204.800$  quadrados, em somente uma janela, mostrados de frente para o usuário, renderizados pelo Java 3D em tempo real.

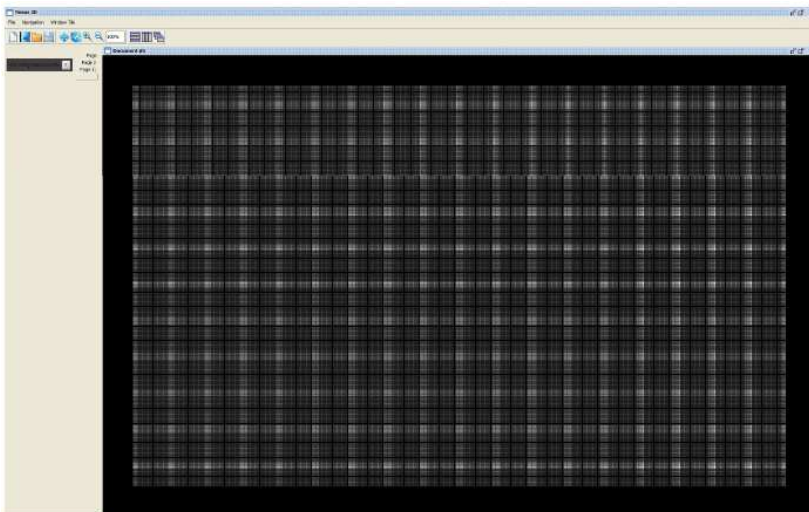


Figura 60: Amostragem de  $2 \times 320 \times 320 = 204.800$  quadrados, renderizados em tempo real, pelo Java 3D.

### 5.3.1.4 Testes de desempenho para vértices e arestas

A Figura 61 apresenta o protótipo com duas janelas contendo dois conjuntos de 40 X 40 (1600) quadrados, um conjunto de frente para o outro, e com seus quadrados já contendo texturas, e cada quadrado de cada conjunto é conectado ao quadrado em frente por uma linha. Cada linha formada entre dois quadrados é pintada com uma cor variando do amarelo ao branco.

Devido à inclusão do Billboard e da textura para cada quadrado, cada um desses quadrados ocupa mais memória do que simplesmente o quadrado branco fixo. Portanto, dentro do espaço de memória da máquina utilizada para os testes, com 1.5 GB de memória RAM, há lugar apenas para 3.200 quadrados com textura e 1.600 linhas. Nesta etapa, as setas das arestas ainda não foram programadas.

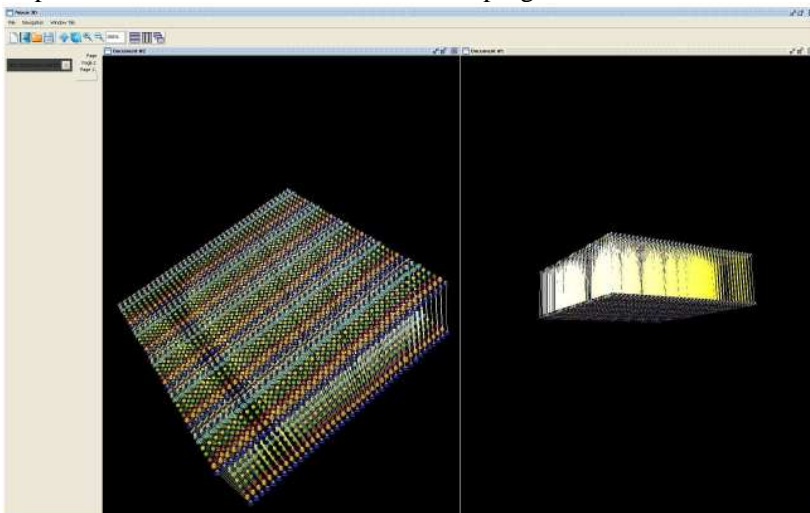


Figura 61: Teste de desempenho com duas janelas, cada uma com 3200 quadrados com oito texturas diferentes, e 1600 arestas. As arestas foram desenhadas com um degradê entre as cores branca e amarela.

### 5.3.1.5 Testes de elementos de arestas e espessura de arestas

Considerando que algumas relações semânticas são direcionais, bidirecionais ou simples, foi testado o uso de linhas nas extremidades de cada aresta, representando suas setas. Inicialmente, estas setas são tetraedros com altura de três pontos e largura de um ponto (Figura 62).

Este teste também mostra a modificação de valores de espessura de cada uma das arestas e seu impacto no desempenho. Durante o experimento foi notado que as arestas mais espessas impactam o rendimento da máquina, mas a demora provocada é tolerável.

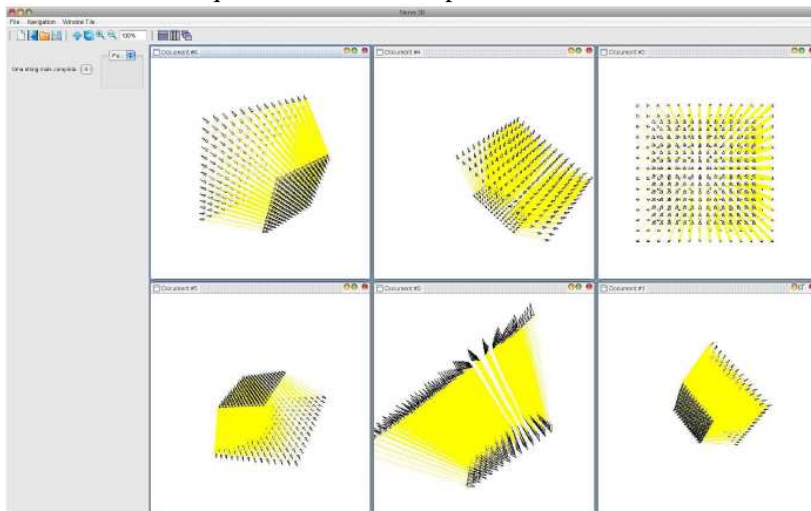


Figura 62: Protótipo com seis janelas com o mesmo grupo de arestas, com suas respectivas setas, cores e espessuras.

### 5.3.1.6 Testes de carga de imagens da base de dados para o espaço 3D

Apesar de os vértices de uma rede de relacionamentos de ontologias possuírem um desenho padrão, dependendo da classe de vértice que representa (se é um termo ou se é uma Palavra Universal), dá-se ao usuário final a possibilidade dele escolher a imagem que representa uma classe de vértice. Desse modo, o usuário pode personalizar os elementos visuais de sua instância de ontologias, na forma de redes de relacionamento.

Para que tal ocorra, foi programado um recurso que permite ao usuário escolher uma imagem (Figura 63) de um sistema de arquivos e armazenar esta imagem em uma base de dados. Essa comunicação com a base de dados *Postgres* é feita através da API JDBC (*Java Database Connectivity*), da plataforma Java.

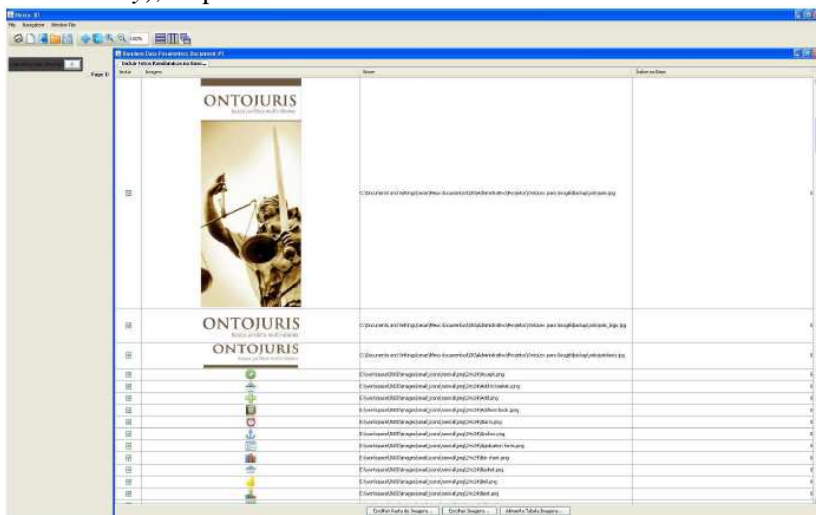


Figura 63: Interface que permite ao usuário escolher uma imagem presente em um sistema de arquivos, e armazená-la em uma base de dados *Postgres*.

Após este experimento, foi possível alimentar a base experimental com algumas fotos. A etapa seguinte do experimento foi a carga destas imagens no espaço 3D, como texturas em vértices, a partir da base de dados já alimentada.

Cada imagem foi armazenada na base como uma matriz de pixels. Essa matriz foi armazenada em cada uma das células da tabela de

imagens; do momento de sua carga como textura sobre uma superfície, essa mesma matriz de pixels foi copiada.

A Figura 64 mostra um conjunto de vértices com texturas carregadas a partir de imagens gravadas em base de dados.

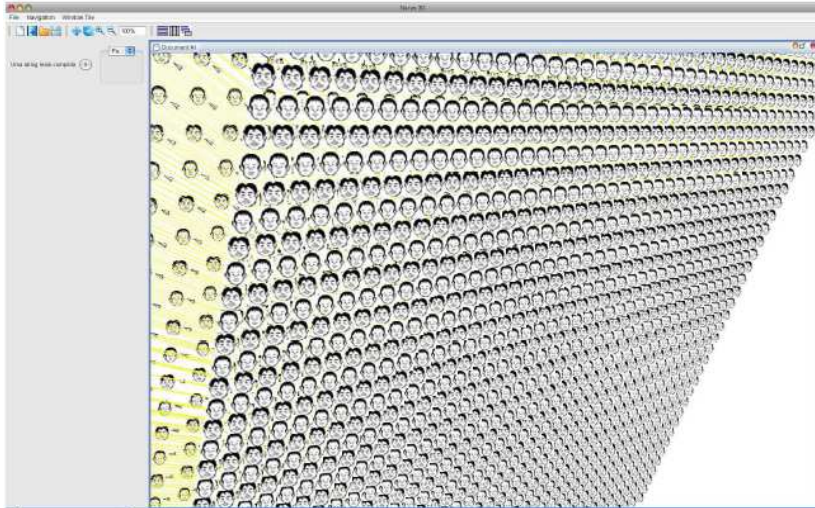


Figura 64: Conjunto de vértices com texturas carregadas a partir de base de dados.

### 5.3.1.7 Rótulos de vértices e efeito de Billboard

Considerando que os vértices da rede de relacionamentos das instâncias de ontologias representam os termos destas instâncias e que contém texto, tais vértices devem oferecer a possibilidade de mostrar o texto que o termo representado contém. O texto anexado a cada um dos vértices é o seu rótulo (Figura 65).

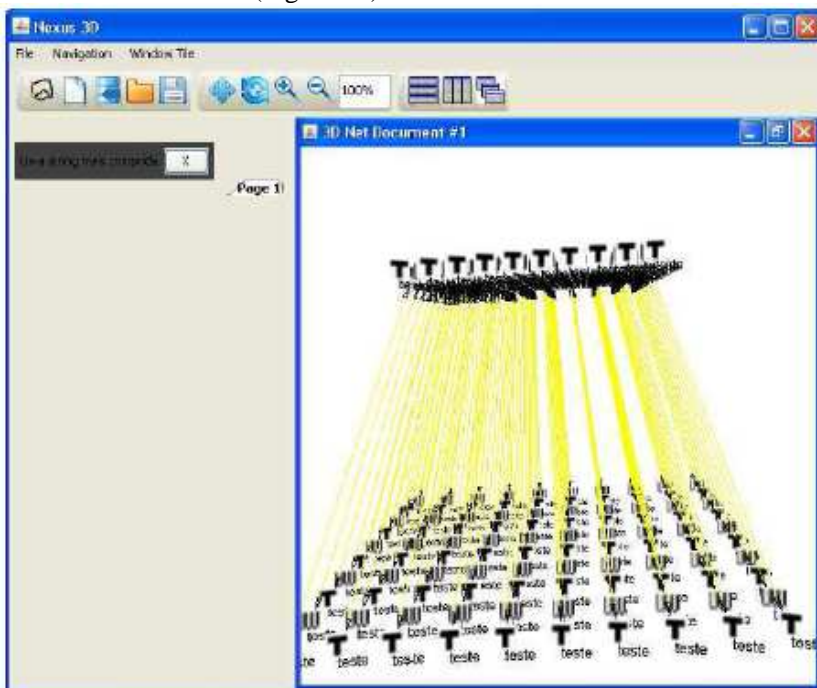


Figura 65: Protótipo do editor mostrando várias instâncias de um mesmo vértice experimental com o seu rótulo.

Nesta etapa, cada quadrado tem o seguinte comportamento: ele sempre fica de frente para o usuário (este tipo de objeto é chamado de *Billboard*). No caso da Figura 65, ao girar o ambiente tridimensional, como os quadrados não são forçados a ficar de frente para o usuário, este não vê as costas do quadrado, que é invisível.

### 5.3.1.8 Carga de termos de ontologias para o espaço 3D

Após o processo de modelagem dos vértices, foi implementada a carga de termos a partir da base de dados de ontologias. Na sua primeira forma (Figura 66), o quadrado contendo o ícone representativo do vértice é posicionado na extrema esquerda, em cima do vértice. O texto é contido em somente uma linha, sem quebras. A Figura 66 também mostra uma janela contendo uma disposição helicoidal dos termos, no espaço 3D. Os dados foram carregados ainda criptografados.



Figura 66: Disposição helicoidal dos termos de ontologias ainda criptografados, a partir da base de dados.



A Figura 67 apresenta um resultado de carga somente de termos, a partir da escolha de um domínio. Os termos se sobrepõem devido à simplicidade do algoritmo de posicionamento.

Por causa da semelhança dos métodos de implementação dos editores 2D e 3D, as interfaces de carga de domínios também são semelhantes. Uma diferença marcante entre a interface dos dois aplicativos é o fato de que a plataforma Java utiliza os recursos gráficos de placa aceleradora, disponibilizando vértices em tempo real.

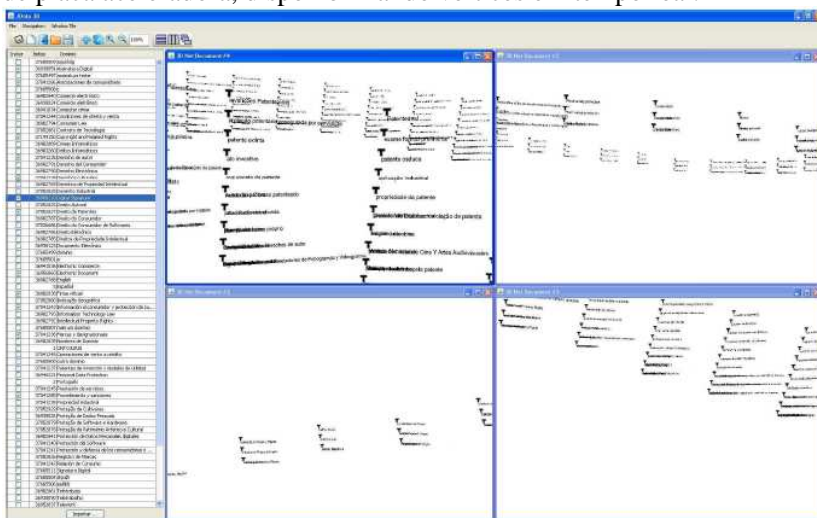


Figura 67: Posicionamento somente de termos, dado o domínio escolhido pelo usuário, carregados no ambiente 3D.

### 5.3.1.9 Dependência do posicionamento das arestas com os seus vértices conectados

No modelo de rede de relacionamentos disposta no espaço 3D, foi assumido que o posicionamento das arestas seria dependente do posicionamento de seus vértices conectados, de origem e destino, tal como no modelo do editor 2D. Para tal, cada par de vértices deveria estar conectado dinamicamente (enquanto sendo reposicionado) por uma aresta. A Figura 68 mostra os primeiros resultados da tentativa de conectar pares de vértices com uma aresta.

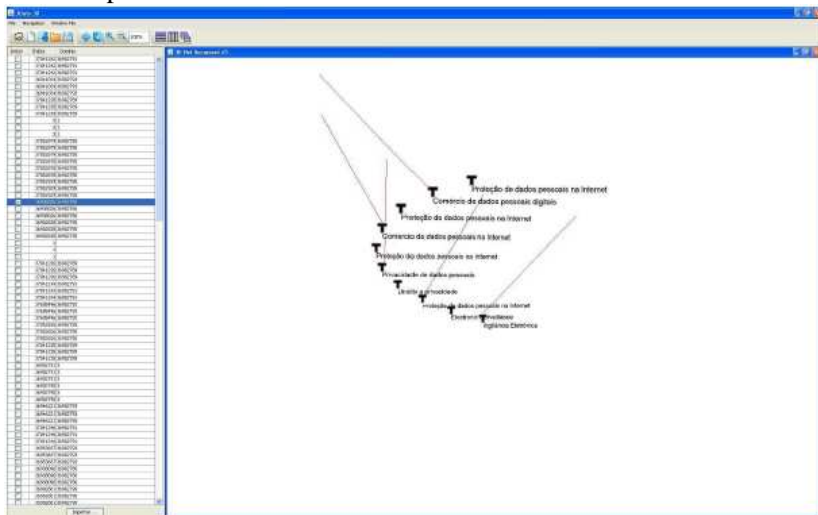


Figura 68: Tentativa inicial de conectar pares de vértices através de arestas.

### 5.3.1.10 Personalização do formato de arestas

O processo de determinação dos pontos que constituem as arestas foi programado para ser configurado. Em seu modelo básico, cada aresta é determinada pelos seus pontos de extremidade, que coincidem com a posição dos dois vértices conectados, e por pontos intermediários que determinam o formato de uma aresta<sup>12</sup>.

A Figura 69 mostra um conjunto de pares de vértices conectados por arestas personalizadas com uma trajetória padrão: um ponto mais alto, resultando em um formato em “v”. Embora o exemplo mostre todas as arestas com o mesmo formato, o sistema permite a personalização particular de cada uma destas arestas.

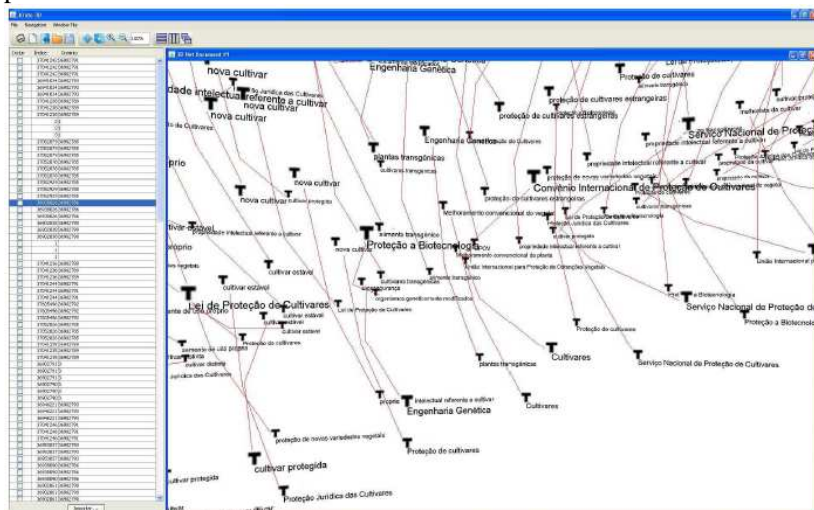


Figura 69: Exemplo de arestas personalizadas, com o posicionamento de seus pontos resultando em um formato com uma quebra central (formato em "v").

<sup>12</sup> O cômputo destes pontos para o desenho final da aresta é semelhante ao algoritmo mostrado nos itens 1.7 a 1.11 do apêndice, para arestas 2D.

### 5.3.1.11 Inclusão de setas

Conhecendo que cada uma das arestas pode ter seu formato determinado, podendo ser diferente da linha reta normalmente conhecida, a próxima etapa foi a construção de setas em diferentes formatos, e o seu alinhamento com o segmento de reta da extremidade da aresta hospedeira.

A Figura 70 mostra as setas desenhadas nas extremidades das arestas, ainda não alinhadas com seu segmento de reta correspondente. O cômputo das coordenadas das extremidades das arestas é semelhante ao processo explicado na seção 1.11 do Apêndice, para arestas 2D.

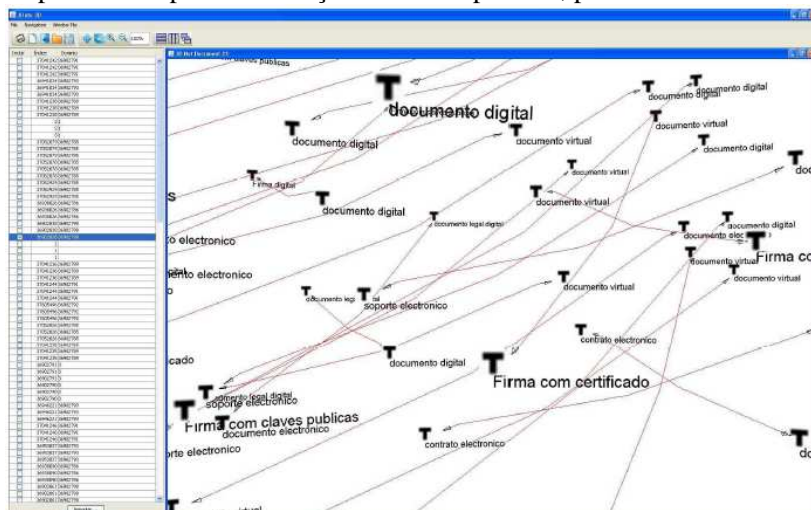


Figura 70: Demonstração do uso de setas nas extremidades das arestas.

### 5.3.1.12 Múltiplas arestas entre dois vértices

Prevedo o reuso das classes que formam a base para a construção do aplicativo (uma *framework*), foi programada em sua estrutura uma classe contêiner que permite a inclusão e apresentação de várias arestas, personalizadas, entre dois vértices. Dessa forma, é possível representar múltiplas formas de associação entre dois vértices, de acordo com a necessidade de futuros aplicativos.

Durante a construção do editor, foi percebido que, devido à simetria em relação ao ponto médio entre dois vértices, as arestas entre esses dois vértices formaram uma constrição no mesmo ponto médio.

A Figura 71 apresenta uma janela contendo vários pares de vértices contendo o mesmo conjunto de arestas múltiplas (com cores diferentes) entre si, com este problema de constrição. Já a Figura 72 apresenta as múltiplas arestas com o problema corrigido.

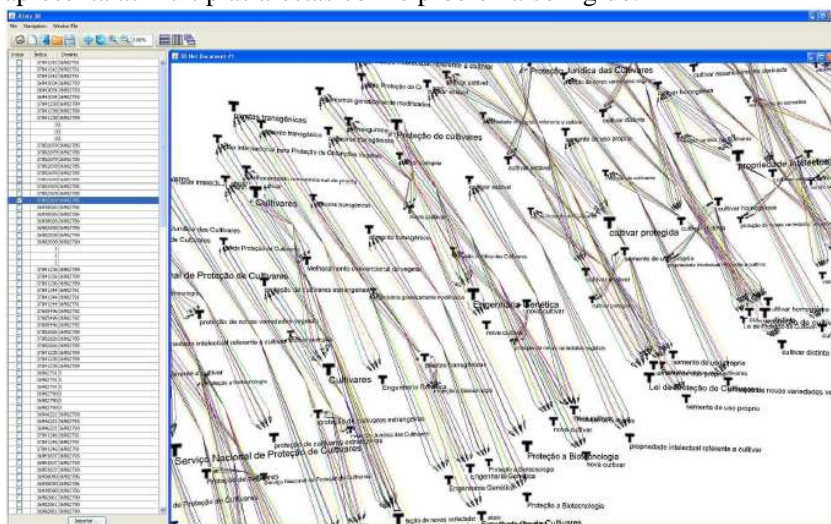


Figura 71: Múltiplas arestas com o problema da constrição.

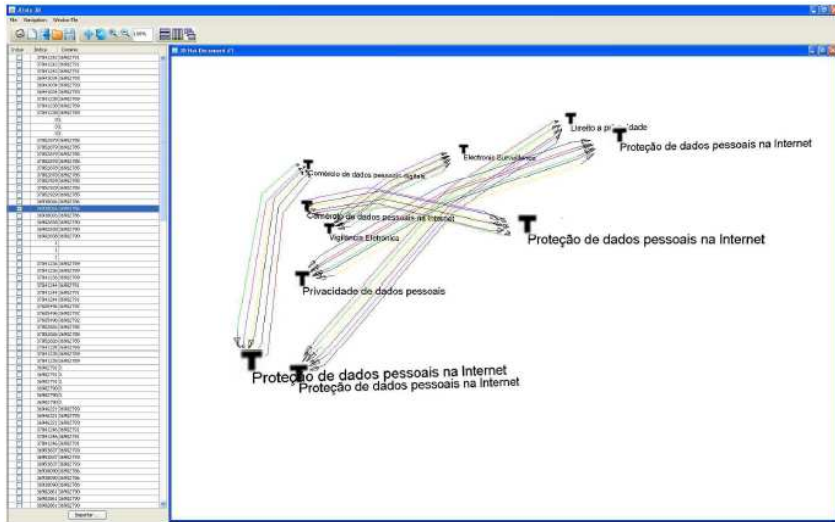


Figura 72:Construção das múltiplas arestas corrigida.

### 5.3.1.13 Arestas conectando arestas entre si e arestas conectando arestas com vértices

Novamente prevendo o reuso das classes da *framework*, foi construída uma estrutura de dados que permite que um grupo de arestas conecte outros dois grupos de arestas, através de pontos com posição relativa às suas duas extremidades. O posicionamento das extremidades das arestas conectoras é feito pelo cálculo do comprimento de arco das arestas conectadas. Na Figura 73 é mostrado um exemplo de conexão de arestas com arestas.

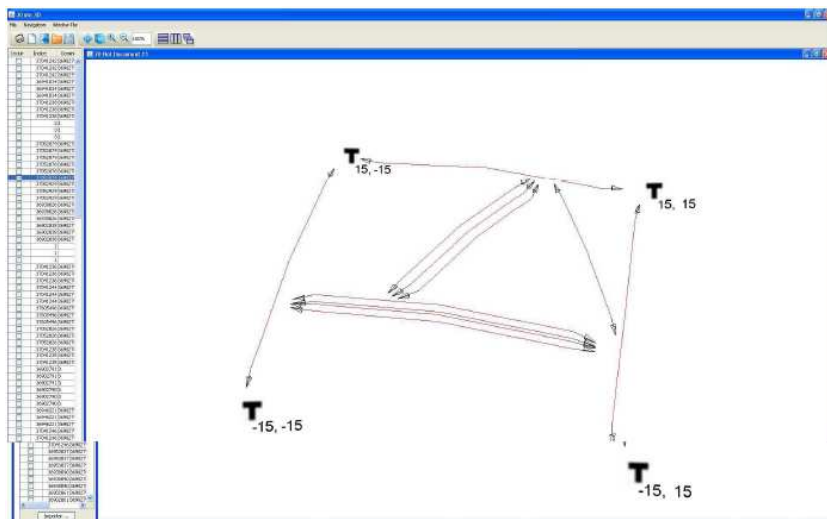


Figura 73: Exemplo de arestas conectando arestas.

Também foi programada uma estrutura que permite a conexão de um vértice com um grupo de arestas através de outro grupo de arestas. Para este caso, o posicionamento do grupo de arestas conectoras também tem a posição de sua extremidade dependente do comprimento de arco da aresta conectada.

A Figura 74 mostra um exemplo de conexão de arestas e vértices através de arestas.

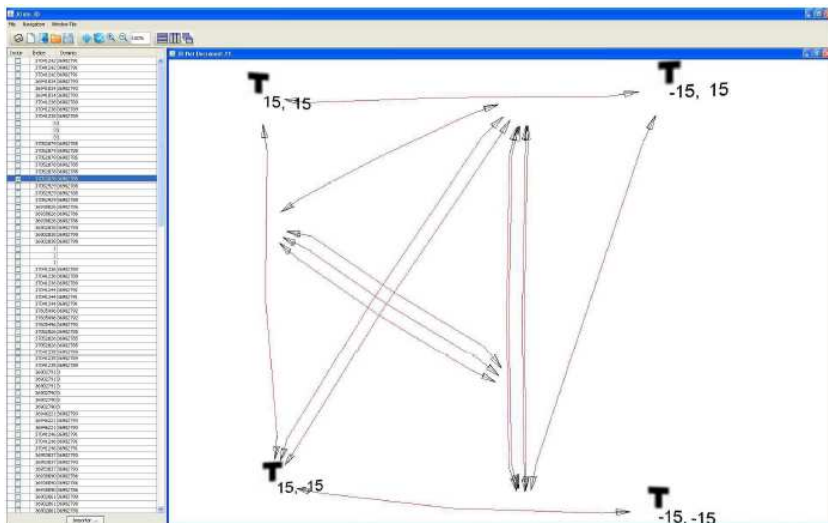


Figura 74: Exemplo de arestas conectando arestas com vértices.

No exemplo, é possível observar que não há um elemento visual indicando para onde a aresta conectora está apontando. Ela apenas é posicionada relativamente a uma proporção do comprimento da aresta conectada. Todas as arestas têm suas posições atualizadas de maneira dinâmica, em relação à posição dos vértices.

#### 5.3.1.4 Programação de envoltórias 3D

Nos itens 4.11, 4.12 e 4.13, são apresentados algoritmos para o cômputo de envoltórias quadradas e octogonais ao redor dos vértices das instâncias de ontologias. Pelos resultados visuais, nota-se que a envoltória octogonal tem um melhor aproveitamento do espaço, passando linhas diagonais sobre regiões que não contêm vértices. Este aproveitamento não é feito nas envoltórias quadradas, pois as linhas da envoltória são somente verticais ou horizontais, podendo abranger grandes espaços que não possuem vértice algum.

Partindo dessa premissa, foi iniciada uma pesquisa sobre a construção de uma envoltória que oferecesse o mesmo aproveitamento do espaço 3D. A partir de uma página que oferece formas de poliedros regulares de Platão e Arquimedes (Interactive Polyhedra, 2010), foi escolhido um poliedro que serviria de base para as envoltórias 3D. O



poliedro escolhido foi o rombicuboctaedro (na Figura 75, marcado por um quadrado), já que a sua aparência, vista de cada um dos três eixos, assemelha-se a um octógono, utilizado como forma base das envoltórias no editor 2D.

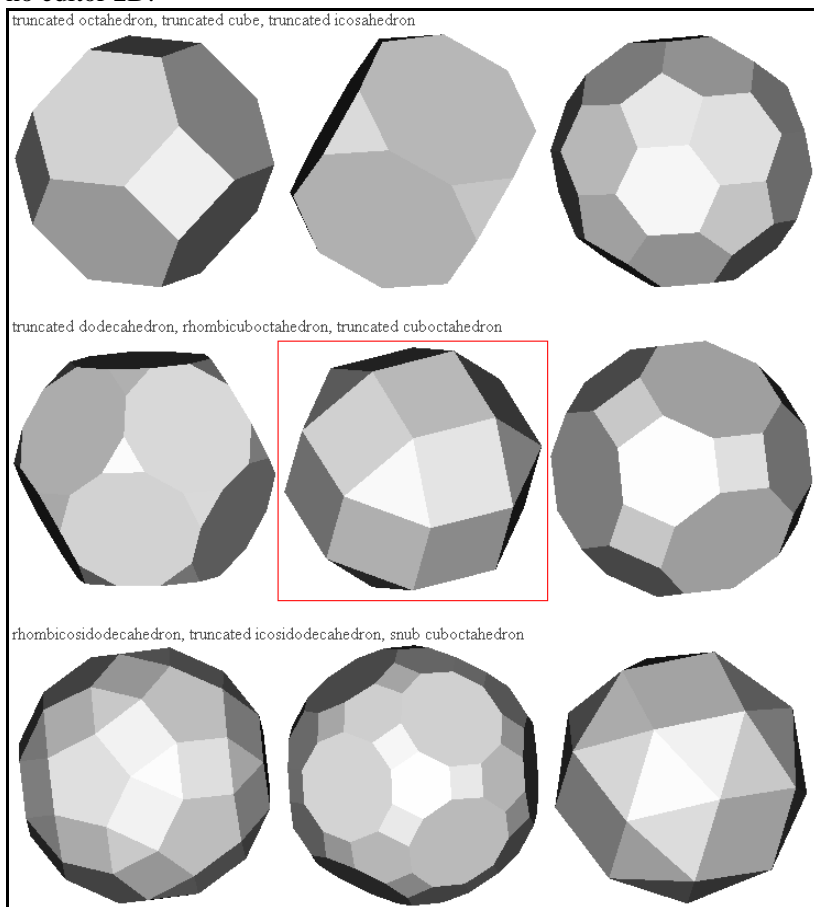


Figura 75: Poliedros de Arquimedes.

Inicialmente, a envoltória foi construída de maneira manual, ponto a ponto, para posteriores testes de posicionamento de seus pontos de construção (Figura 76).

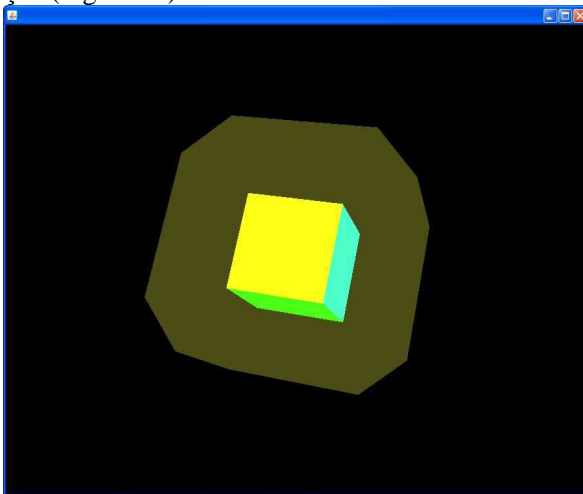


Figura 76:Envoltória rombicuboctáédrica transparente (contendo um cubo colorido em seu interior para facilitar a identificação de suas faces e pontos constituintes).

A Figura 77 mostra um exemplo bem comportado de envoltória.

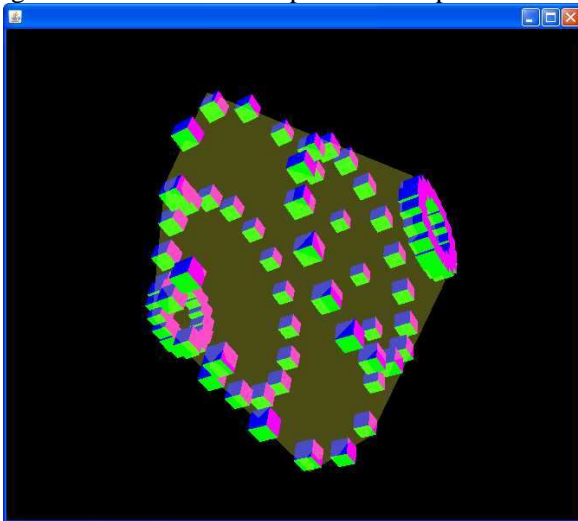


Figura 77:Exemplo bem comportado de envoltória de base rombicuboctáédrica. Sua disposição está ao redor do centro de cada cubo.

A Figura 78 mostra um exemplo de envoltória com ‘rebarbas’ (os triângulos sobrando ao redor da envoltória). Nota-se que a envoltória permanece ao redor do ‘centro geométrico’ dos cubos coloridos, e não ao redor dos limites de cada cubo. Para que a envoltória ficasse ao redor também dos limites dos cubos, de maneira perfeita, deveria ser computado também os oito pontos representados pelas extremidades destes cubos.

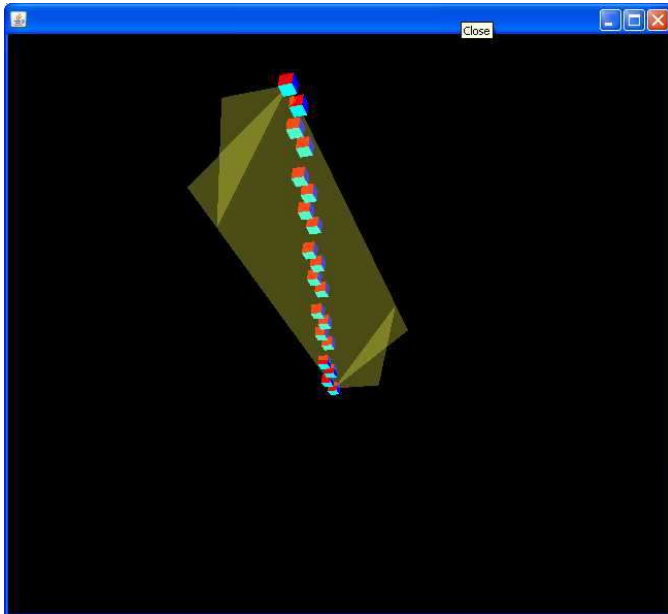


Figura 78:Exemplo mal sucedido do mesmo modelo de envoltória.

### 5.3.1.15 *Envoltória rombicuboctaédrica sobre vértices*

Garantido o processo computacional para o calculo das coordenadas da envoltória, foram feitos os primeiros testes de envoltórias envolvendo vértices posicionados pseudo-aleatoriamente.

A Figura 79 mostra o editor de ontologias contendo seis janelas filhas com 24 pontos gerados aleatoriamente dentro de um intervalo. Na figura, nota-se que as envoltórias possuem rebarbas, de tamanho pequeno, porque os vértices envolvidos estão distribuídos de maneira uniforme, quase formando uma esfera. Nos casos em que os pontos ficam distribuídos de forma mais plana, as rebarbas resultantes se tornam maiores, resultando em um efeito bastante antiestético.

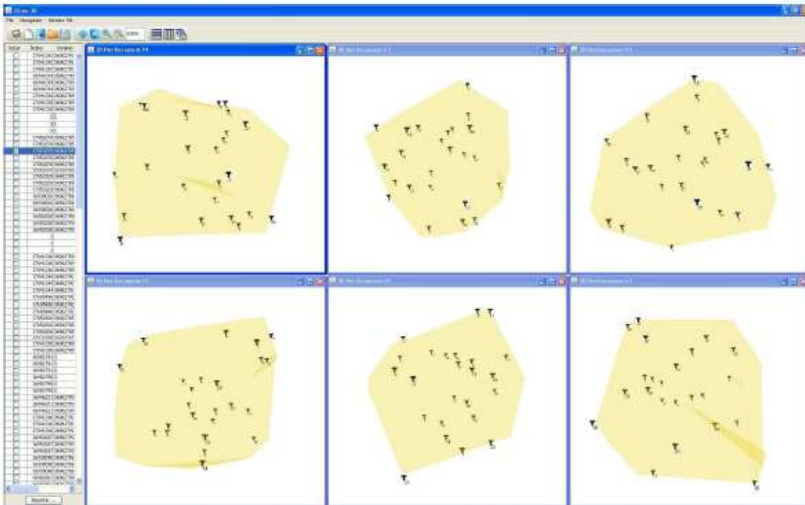


Figura 79: Experimento com seis janelas contendo uma envoltória de base rombicuboctaédrica, sobre vértices posicionados aleatoriamente ao redor de um ponto.

### 5.3.1.16 Combinando envoltórias e vértices para representar domínios e termos

Posteriormente foi programado o processo de carga de domínios com envoltórias ao redor de termos, das instâncias de ontologias, a partir da base de dados (Figura 80). As envoltórias são transparentes e os vértices são dispostos em espiral. A representação visual da inclusão de um domínio dentro de outro não foi programada.

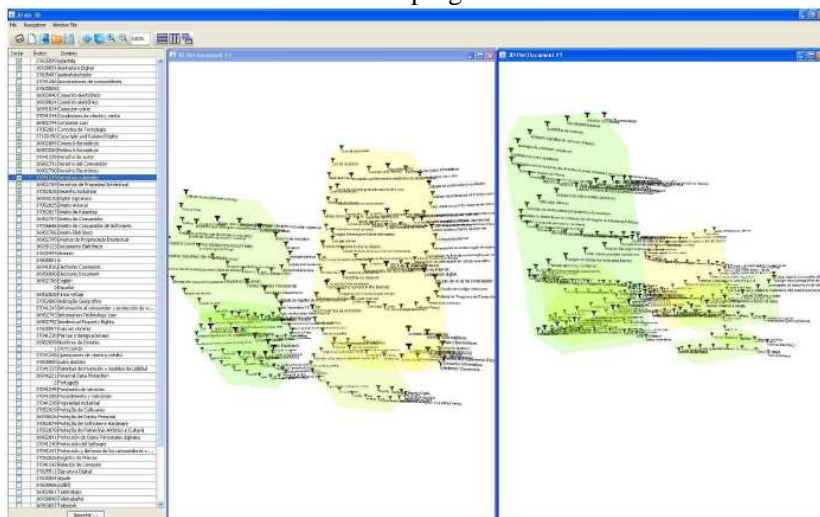


Figura 80: Combinação de envoltórias de base rombicuboctaédrica (representando os domínios) e vértices (representando os termos) no espaço 3D.

### 5.3.1.17 Simplificações no algoritmo de cálculo da envoltória paralelepípedicas

Após alguns experimentos, as envoltórias de base rombicuboctaédrica foram substituídas por envoltórias paralelepípedicas. A substituição se deu por três motivos: a) Houve a redução do número de pontos a controlar, de  $6 \times 4 = 24$  pontos, para apenas 8; b) Conseqüentemente, o número de faces passou de  $6 + 2 \times 4 + 2 \times 4 = 22$  para apenas 6; c) A envoltória paralelepípedica não apresenta rebarbas, tem programação mais simples e é atualizada mais rápido.

A única desvantagem apresentada pela envoltória paralelepípedica foi o aproveitamento reduzido do espaço, comentado no item 4.3.1.14 Programação de Envoltórias 3D.

### 5.3.1.18 Arestas entre envoltórias paralelepípedicas

Previendo cenários que exijam a representação visual de associações entre grupos, implementou-se uma conexão entre duas envoltórias através de uma aresta (ou de um grupo de arestas). A aresta conectora (ou o grupo) foi planejada para ter seus extremos coincidindo com a superfície das duas envoltórias.

A tentativa inicial de limitar a extensão das arestas conectoras entre a superfície limítrofe das envoltórias conectadas não foi bem sucedida. Como solução substituta para duas envoltórias, o grupo de arestas que as conecta tem seus limites apontando para o centro geométrico das envoltórias conectadas (Figura 81).

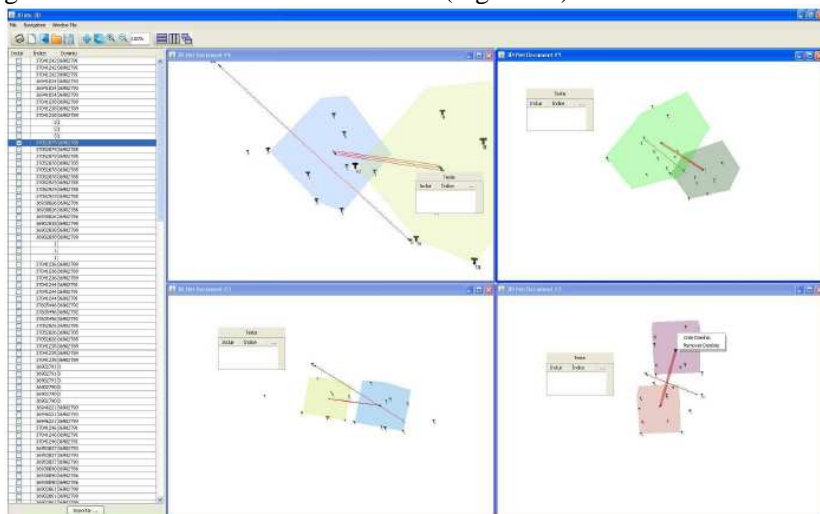


Figura 81:Exemplos de envoltórias paralelepípedicas.

### 5.3.1.19 Navegação

Os operadores de navegação usados são o de translação do ambiente, rotação do ambiente e *zoom*. A navegação do ambiente baseia-se nos componentes do mouse. Dessa forma, o usuário necessita de somente um componente para conseguir manipular os elementos de uma instância de ontologia, no espaço 3D. Para que o usuário altere o seu modo de navegação entre translação, rotação e *zoom*, basta clicar sobre botões no ‘menu’ do aplicativo.

As interfaces disponíveis são pressão de botão e arraste do mouse para movimentos de rotação e translação, e o uso do botão de rolagem

para o *zoom*:

**a) Translação:** o resultado conseguido para a translação não foi julgado suficiente. O problema ocorre quando do movimento do objeto no espaço, apesar de ocorrer no plano perpendicular ao eixo de visão da câmera, ser desproporcional ao deslocamento do mouse, o que provoca freqüentes fugas do grupo de objetos do eixo de visão do usuário. Uma solução paliativa é deslocar levemente o mouse, no plano, evitando fugas dos objetos. O ideal seria um deslocamento WYSIWYG (“O que você vê é o que você tem”);

**b) Rotação:** No início das prototipagens sobre rotação, um problema recorrente foi a perda de coerência entre o movimento do mouse no plano e o giro do grupo de objetos no espaço 3D. Partiu-se do princípio de que o giro resultante dos objetos no espaço, com o movimento do mouse, deveria dar a impressão de que o mundo virtual gira como se o usuário pegasse a ponta do objeto que está mais próximo de si, e movesse esta ponta para os lados. Este comportamento é chamado de *Orbit Behavior* (Comportamento de Órbita) e é usado em editores gráficos (O editor gráfico 3D Studio 5, da AutoDesk, foi usado como referência).

Entretanto, o resultado inicial da implementação não foi como o esperado: após o usuário fazer um, dois ou três giros, o ambiente já não acompanhava mais o movimento natural do mouse; era comum o usuário mover o mouse em uma linha vertical reta, para baixo, e o ambiente girar para a esquerda.

Para resolver esse problema, duas soluções estavam disponíveis: a primeira, usar uma classe da própria biblioteca do Java 3D permite a execução de comportamento de órbita, com a desvantagem de sobrepor o comportamento de arraste dos objetos isolados. Inicialmente, tal solução não permite a capturar dos objetos isolados para movê-los, e alterar o código fonte seria trabalhoso e com risco de não apresentar resultados a tempo; a segunda, utilizar uma classe de terceiros que não fizesse parte da própria biblioteca do Java 3D (JAVA2S, 2010).

A segunda opção foi escolhida por estar disponível e apresentar resultados imediatos.

**c) Zoom:** A aproximação ou afastamento da câmera se baseia no princípio de que a câmera está posicionada em um ponto positivo qualquer sobre o eixo de profundidade, e sempre apontando para o centro (ponto (0, 0, 0)). O afastamento significa o aumento do valor no

eixo de profundidade. A aproximação significa a redução do valor neste mesmo eixo. O controle do afastamento é feito pelo botão de rolagem do mouse.

Quanto ao manuseio dos elementos, procurou-se ver as instâncias de ontologias de forma mais detalhada. Assim foi implementado o controle sobre o posicionamento (translação) particular de cada um dos vértices (termos, palavras universais e envoltórias) da rede de relacionamentos correspondente.

A translação de cada um dos vértices é executada com o arraste do mouse. O resultado desta implementação foi parcialmente bem sucedido, embora ainda persista o problema de fuga de foco no movimento de translação.

#### *5.3.1.20 Menus e formulários*

A próxima etapa na construção do editor foi a programação dos menus dependentes de contexto (as opções oferecidas pelo menu são modificadas de acordo com o elemento onde ocorre o clique do mouse, podendo ser sobre uma parte vazia do espaço virtual 3D, vértices, arestas, envoltórias e extremidades de arestas). Nesse caso, de menus flutuantes, não há sobreposição do ambiente 3D (*heavyweight*) sobre o menu flutuante (*lightweight*). A etapa seguinte foi a programação da disposição de formulários móveis sobre o ambiente 3D<sup>13</sup>.

---

<sup>13</sup> A Figura 68 mostra cada uma das quatro janelas contendo um formulário simples sobre o ambiente tridimensional.



### 5.3.1.21 Criação de vértices (*Termos ou Palavras Universais*)

A programação do processo de construção de vértices (Figura 82), pelo usuário, ainda possui duas limitações: a) Os vértices criados não estão associados à base de dados. Se o aplicativo é fechado, os vértices criados se perdem. b) Quando um vértice é criado, ele é sempre posicionado na origem do espaço 3D (0, 0, 0), quando o ideal deveria ser o posicionamento no mesmo ponto clicado com o uso do mouse.

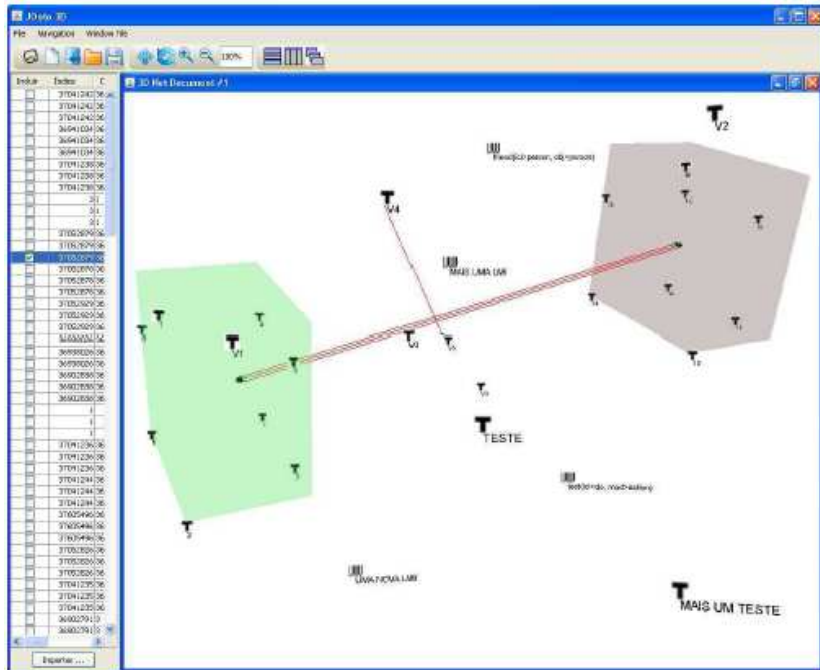


Figura 82: Resultado do processo de criação de vértice (termo nomeado como "TESTE") nas coordenadas (0, 0, 0).

### 5.3.1.22 Criação de arestas (Relações Semânticas)

O procedimento para que o usuário construa uma aresta entre dois vértices é dado a seguir: 1) O usuário clica com o botão direito do mouse sobre um vértice; 2) Escolhe Criar Relação... \Conexo (por enquanto só dispara esta funcionalidade com o submenu 'Conexo', e com mais nenhuma outra; 3) Uma aresta fantasma acompanha a posição do mouse, tal como no modelo 2D (Observou-se que o posicionamento da aresta no espaço, de acordo com a posição do mouse na tela, não é tão fácil de programar); 3) O usuário posiciona o ponteiro do mouse sobre um vértice e clica o botão: a aresta entre os dois vértices é criada; 4) Caso o usuário clique no espaço vazio, ou pressione a tecla ESC, durante o processo de construção da aresta, o processo é cancelado. A Figura 83 mostra uma tela com duas janelas contendo arestas criadas pelo usuário.

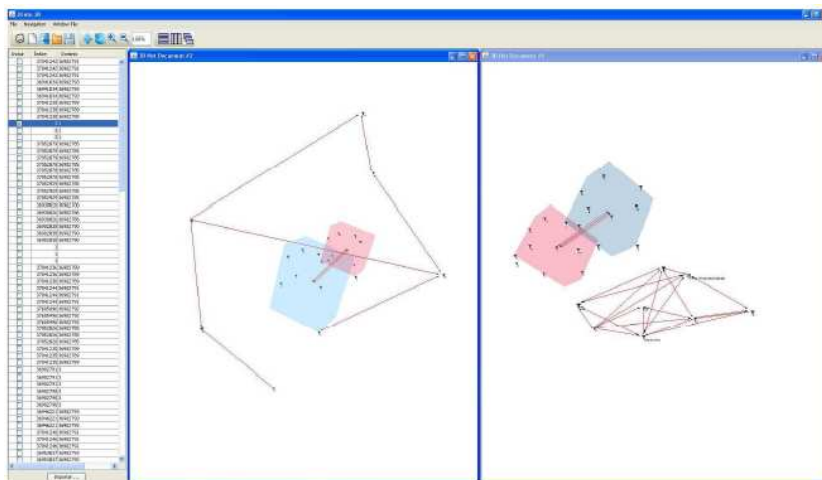


Figura 83:Tela do aplicativo com duas janelas mostrando o resultado do processo de criação de arestas.

### 5.3.1.23 Carga de modelos 3D externos

A carga de modelos 3D externos foi programada com quatro finalidades:

1. A possibilidade de associação destes modelos com conceitos escritos, no idioma original, ou de maneira normalizada, através de Palavras Universais. Dessa forma, caso haja dificuldades do engenheiro do conhecimento, ou de um leigo, entender ou avaliar um conceito escrito em outro idioma, o conceito pode estar conectado a um objeto 3D, dentro do ambiente, para que um segundo indivíduo possa reconhecer o conceito pela aparição virtual de seu objeto associado;
2. A disposição de objetos e suas partes constituintes (visão explodida, muito comum em projetos), mostrando no espaço 3D as relações de parte (meronímia), e classe (hiperonímia) através desses objetos. Assim, engenheiros de conhecimento e leigos poderão entender sobre conceitos que fazem parte de outros conceitos, ou são origem ou grupos de outros conceitos;
3. A associação de conceitos com pontos ou regiões geográficas, para inferências em sistemas de informação geográfica;
4. A possibilidade de personalizar elementos geométricos (setas das extremidades), com estes modelos 3D. Tais extremidades podem ter formato de seta triangular, tridente, quadrado, Letra T ou qualquer outro que o usuário desejar.

Para o caso de personalização da geometria das extremidades, uma solução alternativa foi a possibilidade de permitir ao usuário editar a geometria de seu interesse, mas esta possibilidade foi descartada por ser mais trabalhosa do que a inclusão de modelos já prontos, que podem ser preparados com editores específicos para este fim.

O protótipo pode importar geometrias no formato *VRML (Virtual Reality Modeling Language)*, com algumas limitações, já que esse formato é bastante comum e disponível na Web, e pode ser exportado pelo editor gráfico utilizado para as experimentações, o 3D Studio Max.

A Figura 84 mostra um modelo simples, construído no aplicativo 3D Studio, e importado para dentro do ambiente 3D do editor de ontologias multilíngües. Para que tal carga de um arquivo VRML fosse possível, pelo aplicativo, foi necessária a instalação de uma biblioteca de licença BSD, feita em Java (J3D-VRML97, 2006).

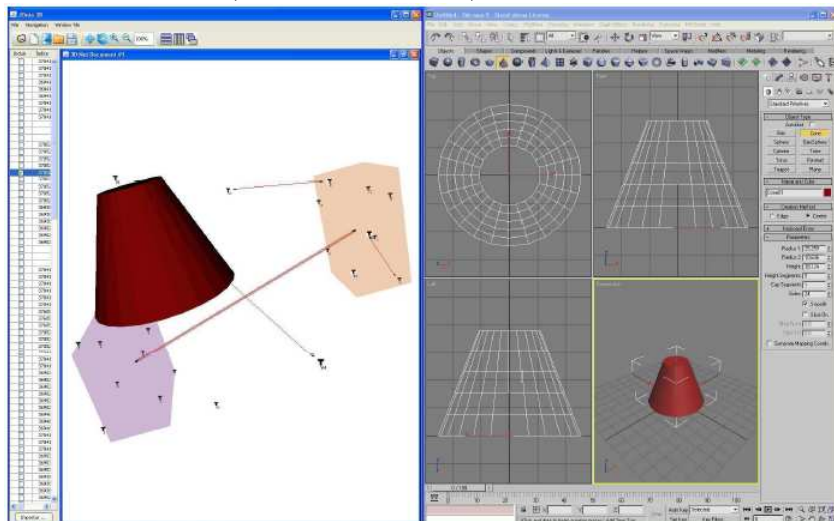


Figura 84: Janelas lado a lado mostrando a construção de um tronco de cone no 3D Studio MAX, e a inclusão desta geometria no ambiente 3D do editor de ontologias multilíngües.

### 5.3.1.24 Parametrização das extremidades das arestas

As extremidades das arestas se baseiam em geometrias armazenadas em arquivos de formato VRML. Desta maneira, suas geometrias podem ser modificadas em editores independentes e depois carregadas para o editor. É possível também disponibilizar uma biblioteca de geometrias para as extremidades das arestas.

Um ponto importante a determinar nessas geometrias é o ponto de origem. Na geometria armazenada em arquivo, esse ponto terá coordenadas (0, 0, 0), sendo os demais pontos de coordenadas positivas. Desse modo garante-se que a extremidade será desenhada fora da extensão de sua aresta hospedeira e conectada ao ponto final desta aresta.

Comparadas com as geometrias a serem importadas ao espaço 3D do editor, as geometrias das extremidades são relativamente simples. Para facilitar sua construção, utilizou-se um editor de coordenadas de geometrias no formato VRML: o aplicativo *VRMLPad* (Figura 85), da *Parallel Graphics* (VRMLPad, 2010).

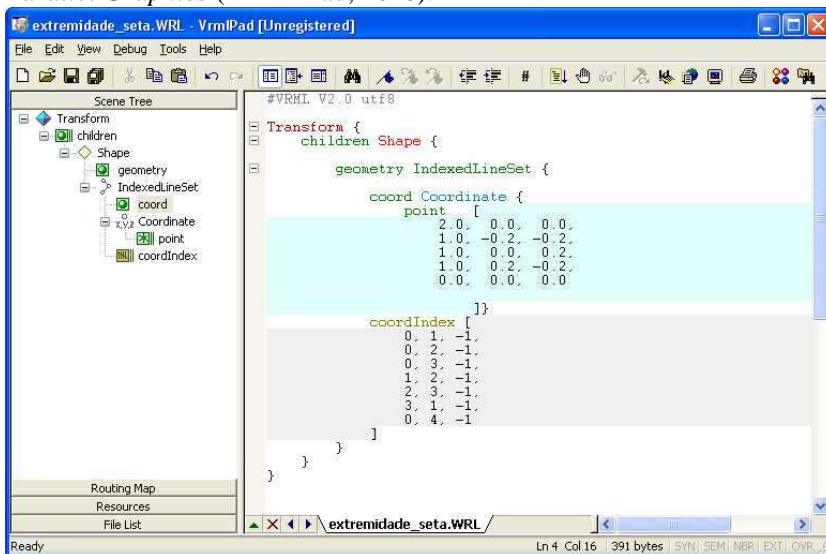


Figura 85: Interface do aplicativo VRMLPad, para a edição de geometrias no formato VRML97.

### 5.3.1.25 Parametrização das arestas

Neste editor, as arestas podem ser caracterizadas em sete propriedades diferentes: 1) Nome do tipo de relação; 2) Cor; 3) Espessura; 4) Tracejado; 5) Geometria (Posição de seus pontos constituintes); 6) Forma da extremidade de origem; 7) Forma da extremidade de destino.

Foi usada a política de guardar os valores em arquivos de configuração de formato próprio. A Figura 86 mostra um exemplo de conteúdo de arquivo de configuração de arestas. Este arquivo determina um tipo de aresta para cada linha. No exemplo, cinco tipos de representação geométrica são representados, um para cada relação semântica: Ocorrência conjunta de termos dentro do texto (Conexo); Sinonímia; Classe; Meronímia; e Equivalência de termos e UW's.

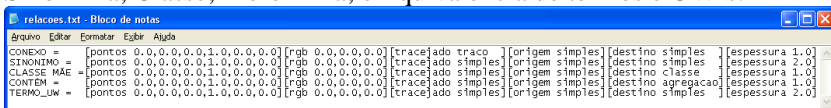


Figura 86: Conteúdo exemplo de arquivo de configuração de arestas.

O arquivo tem uma linha de texto representando cada relação. Cada uma destas linhas tem o seguinte formato: nome da relação semântica representada; pontos determinantes da geometria da aresta; cor da aresta no formato *Red/Green/Blue* (RGB); tipo do tracejado: simples, traço, traço-ponto ou traço-dois pontos; tipo da extremidade de origem; tipo da extremidade de destino; e espessura da aresta em pontos.

Na Figura 87, a aresta somente de pontos equivale à relação de continência de um termo em outro, a aresta sólida representa a relação conexa, a aresta de traço ponto é de relação termo/UW, e a aresta tracejada azul-claro é de relação de classe.

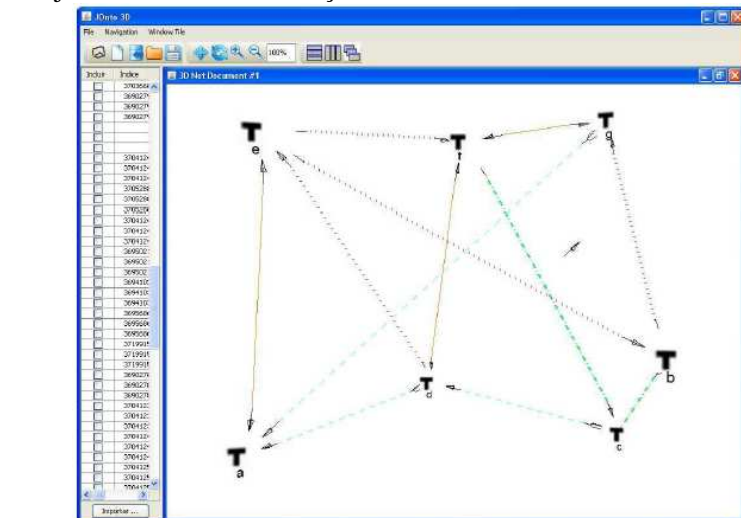


Figura 87: Exemplos de arestas parametrizadas.

### 5.3.1.26 Equivalências sugeridas entre relações semânticas e relações do diagrama de classes UML

Sabendo que as relações podem ser parametrizadas, a representação visual das relações semânticas foi adaptada à simbologia do diagrama de classes da UML, seguindo o mesmo modelo de representação de Bueno (2008).

Neste modelo, inicialmente, existem cinco tipos de relações semânticas. Assim, foi representada cada uma das relações que tenham o seu significado o mais próximo possível do significado das relações do Diagrama de Classes da UML:

<b>Tipo de Relação Semântica</b>	<b>Relação equivalente do Diagrama de Classes</b>	<b>Motivo</b>
Ocorrência conjunta (conexo), ou à relação de equivalência entre termo e palavra universal.	Associação	Um objeto sabe da existência do outro. A linha é contínua, para mudar a representação visual podem ser alterados as cores ou as espessuras
Parte / Todo	Agregação	As duas classes agem como uma unidade, e uma delas é a controladora, e a outra classe é a controlada.
Parte / Todo, mas aqui há o diferencial da condição de existência do todo, para que a parte possa existir	Composição	Tal como na Agregação as duas classes agem como uma unidade, uma delas é a controladora, e a outra classe é a controlada, mas a controlada não existe sem a controladora.
Tipo, ou Classe/Subclasse	Generalização	Em PENDER, Cap. 6, é explicado que existe uma sutil diferença entre tipo e classe. Por enquanto, para o modelo do editor de ontologias, estes conceitos são considerados equivalentes.
Sinonímia	Associação	É o terceiro caso de uso de representação visual de Associação.

Quadro 16: Significado das relações do Diagrama de Classes da UML.

Entretanto, a relação de sinonímia entre termos tem uma desvantagem: caso houver vários termos que sejam sinônimos de um termo inicial, dentro de um dado domínio, todos esses termos serão sinônimos entre si. Visualmente, todos os termos sinônimos entre si estariam conectados por arestas, as chamadas cliques (se existem N vértices, existirão  $N \times N$  arestas), o que sobrecarrega a rede da instância

da ontologia com muitas arestas. Uma solução alternativa para este problema foi representar os termos sinônimos envolvidos pela mesma envoltória, reduzindo a demanda computacional para a sua representação visual. Este modelo foi usado no protótipo 2D do editor de ontologias.



5.3.1.27 Modelo de domínio

A Figura 88 sugere um modelo de domínio para o editor de ontologias.

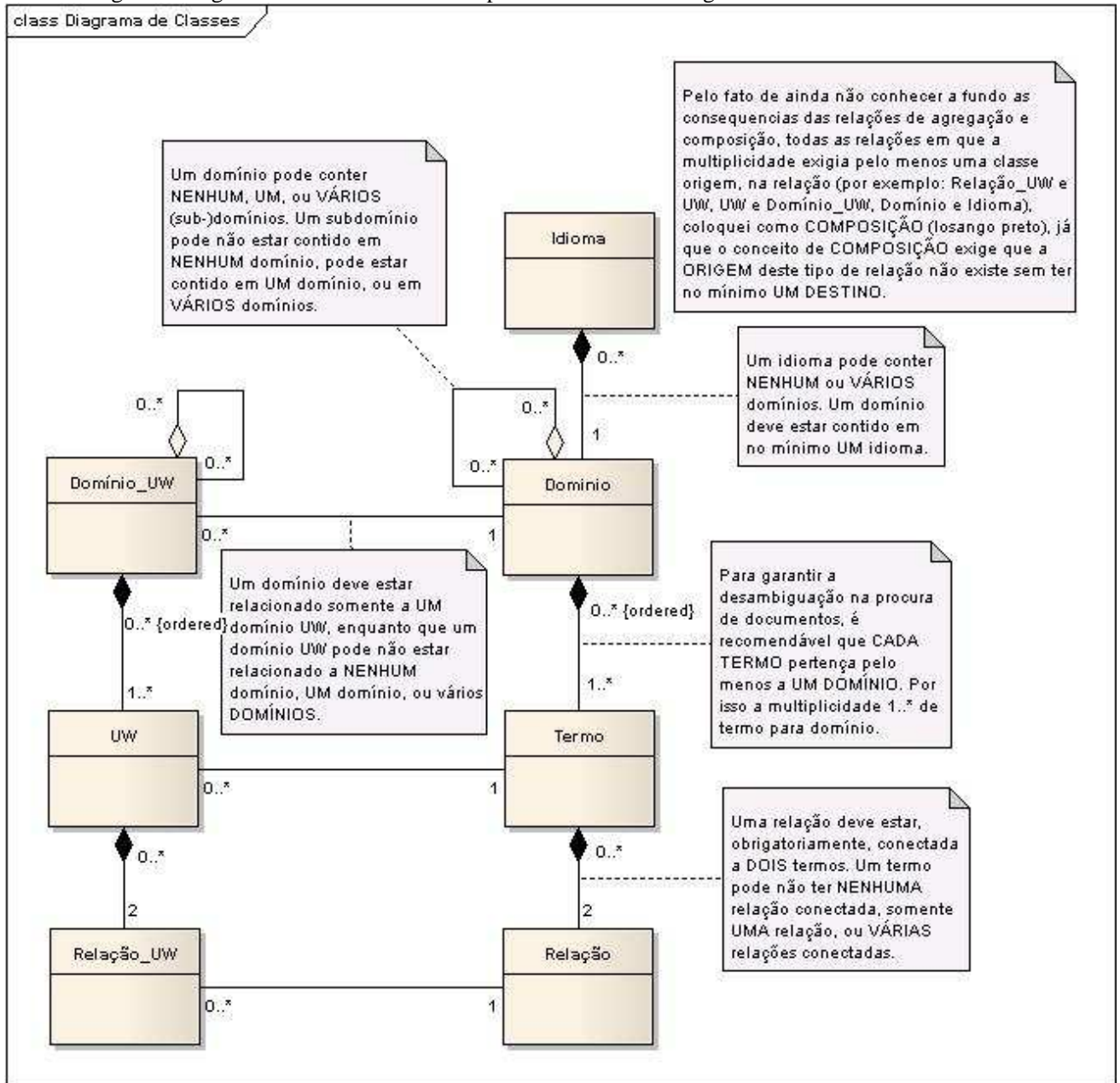


Figura 88:Modelo de domínio sugerido para o editor de ontologias.

### 5.3.1.28 Primeiros resultados com simbologia UML

A etapa seguinte foi a programação das arestas para serem representadas como as relações do Diagrama de Classes da UML.

A Figura 89 mostra o resultado inicial com os vértices e relações explicados abaixo:

1. T é classe-mãe de  $T_3$  (ou  $T_3$  é sub-classe de T) (representada por uma seta de T a  $T_3$ )
2.  $T_2$  é classe-mãe de T (ou T é sub-classe de  $T_2$ ) (seta)
3.  $T_2$  é conexo de  $T_3$  (sem extremidades: associação)
4.  $T_3$  contém  $T_1$  (ou  $T_1$  é parte de  $T_3$ , ou  $T_1$  está contido em  $T_3$ , agregação: diamante preenchido em  $T_3$ )
5. T contém T1 (ou T1 é parte de T, ou T1 está contido em T, agregação: diamante preenchido em T)
6.  $T_2$  é equivalente a  $UW_0$  (sem extremidades: associação. Aresta diferenciada da relação de Conexo devido à sua maior espessura).

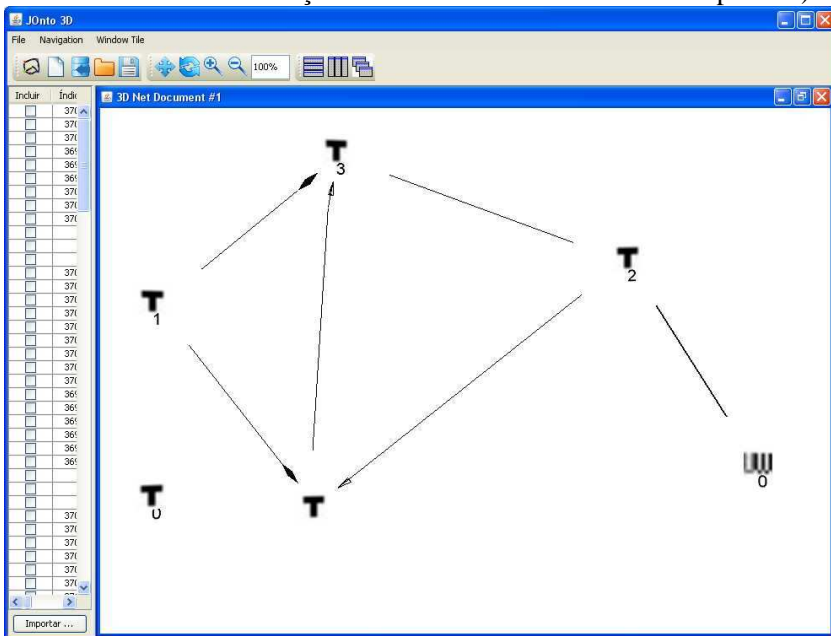


Figura 89: Exemplo de representação das relações semânticas do modelo de Bueno (2005, 2008) como relações do Diagrama de Classes da UML.

### 5.3.1.29 Primeiro resultado com a carga de domínios, termos e relações, com representação UML

Na Figura 90, coloca-se a foto do primeiro resultado referente à carga de domínios, termos e relações entre esses termos, representadas por relações UML. Como as relações usadas da UML são todas de cor negra e contínuas, a única forma de diferenciar tais relações é pela observação das extremidades e espessura da linha, explicados a seguir: a) Sem extremidade e espessura dupla para sinônimos (apesar de a representação em forma de envoltória estar disponível, ela ainda não foi programada no espaço 3D para sinônimos); b) Sem extremidade e espessura simples para conexos; c) Com extremidade rômbrica (ou em diamante cheio) escura para relação de parte; d) Extremidade de seta para relação de classe. (A representação de domínios dentro de domínios ainda não foi programada).

Observando a Figura 90, nota-se que somente o formato das setas não ajuda na identificação das relações.

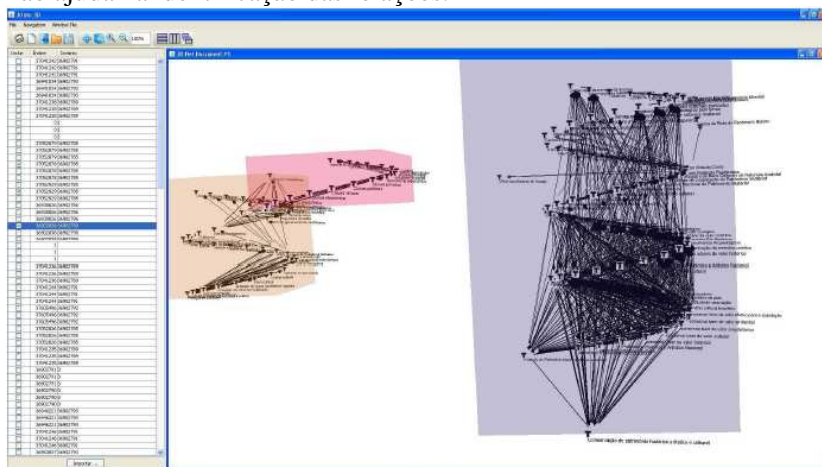


Figura 90: Representação visual das instâncias de ontologias, com as relações no formato UML, e com domínios.

### *5.3.1.30 Tentativa de inclusão de relevos e mapas*

Um mapa ou um relevo são objetos no espaço 3D. Uma utilidade relevante para o editor é poder disponibilizar aos usuários uma explicação sobre a localização de elementos dentro do espaço terrestre.

Sabe-se que os termos de um idioma têm seu significado referente ao local onde vivem as pessoas que se utilizam desses termos para a sua comunicação. Sabe-se também que pessoas que vivem em regiões diferentes usam termos diferentes para definir o mesmo objeto, ou, no mínimo, objetos muito semelhantes.

Uma das vantagens de se pode representar um conceito a um elemento em um espaço virtual 3D, é que é possível associar o elemento 3D a um espaço geográfico, também em três dimensões, que pode ser associado a um espaço maior e assim sucessivamente. Dessa maneira, caso um usuário veja um conceito que não conheça, mas veja também o objeto 3D representado por esse conceito, este usuário poderá descobrir o que significa o conceito antes desconhecido e poderá ver de onde surgiu o conceito, baseando-se na representação geográfica do local associado ao objeto.

No editor, ainda não é possível importar informações de aplicativos especializados na representação de elementos geográficos, mas tal limitação pode ser contornada com a transformação dessa informação geográfica em um modelo 3D no formato VRML, visto acima, e a sua importação para o espaço do editor.

A Figura 91 apresenta um exemplo de um relevo improvisado manualmente, que foi originado de uma foto de topo da cidade de Florianópolis – Santa Catarina – Brasil, tirada com o auxílio do aplicativo Google Maps. A manipulação do relevo foi feita com ajuda do aplicativo 3D Studio Max, da AutoDesk. Ao relevo é possível associar conceitos normalizados (palavras universais), ou termos originais dos idiomas. Ainda não é possível posicionar os pontos das relações conectadas, mas esta limitação será resolvida em uma próxima iteração.

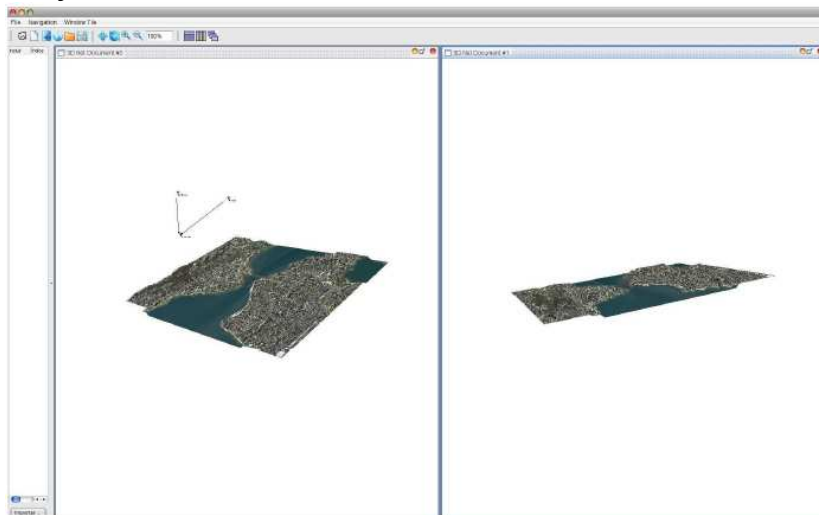


Figura 91: Experimentação de carga de arquivo VRML contendo um relevo improvisado da cidade de Florianópolis.

### 5.3.1.31 Carga e movimentação de múltiplos objetos 3D por janela

O editor 3D pode carregar vários objetos tridimensionais por janela, fazer conexões entre esses objetos e, sob algumas limitações, ser movimentados. Ainda não é possível girar cada objeto em separado ou mudar a escala (o tamanho) do objeto.

Na Figura 92, na janela à esquerda, foi feita a carga de uma flor, uma cabeça, um veículo e uma peça de motor de carro. Esses elementos estão em escalas desproporcionais. Na janela da direita, há dois modelos de veículos que também estão em escalas diferentes. Também ainda não é possível separar os objetos em seus pedaços, para se ter uma visão explodida de um objeto, muito útil para demonstrar relações de meronímia.

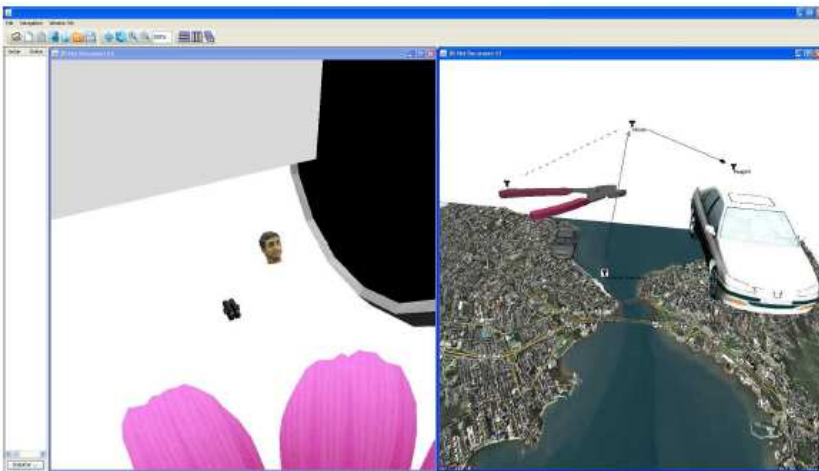


Figura 92:Exemplo mostrando múltiplos objetos 3D por janela.

### 5.3.1.32 Interface Web do Editor

Prevendo o uso deste editor como uma ferramenta de construção compartilhada de ontologias, foram feitos experimentos de funcionamento do editor em páginas web. O experimento comprovou que é possível criar aplicações web no espaço 3D, com múltiplos documentos (*MDI*) e com formulários 2D sobre o espaço 3D (Figura 93). O funcionamento do aplicativo web 3D foi possível com as devidas permissões de acesso da máquina Java.

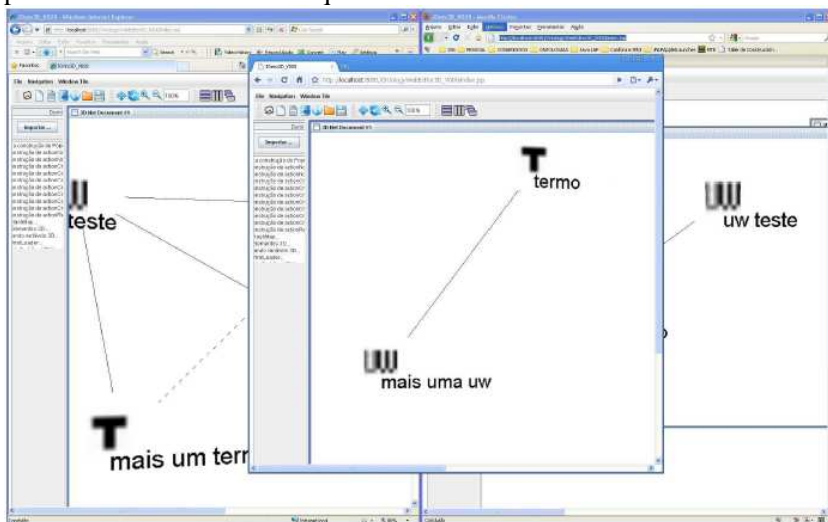


Figura 93: Experimento de funcionamento do editor 3D de ontologias multilíngües rodando com *Applet* em três navegadores Web diferentes: Microsoft Internet Explorer 8, Mozilla Firefox 3.6 e Google Chrome 6.

## 6 PERCEPÇÃO DO EDITOR PELOS USUÁRIOS ENTREVISTADOS

O Quadro 17 mostra os Fatores de Design de Bresciani et. al. (2008), usados como critérios de usabilidade de software, para a avaliação dos aplicativos de construção de ontologias multilíngües. Esta tabela contém sete conceitos básicos, suas correspondentes questões orientadoras, e explicações sobre sensações provocadas nos usuários de interfaces. Os sete critérios são: Impacto Visual, Clareza, Trabalho Final Percebido, Foco Direcionado, Facilitação de Idéias, Modificabilidade, e Suporte à Interação de Grupo.

Tais Fatores de Design foram escolhidos pelos seguintes motivos:

- a) São em baixo número;
- b) São de interpretação e uso simples, portanto, são claros durante a sua aplicação;
- c) Os autores explicam formas de evitar possíveis confusões na interpretação de cada fator;
- d) Também são explicados fenômenos de *trade-off*: Quando o alto valor na avaliação de um destes critérios está ligado a um valor baixo em outro critério, o que ajuda a entender alguma disparidade inesperada em avaliações de usabilidade feita por usuários;
- e) O documento fonte (Bresciani et. al. (2008)) está facilmente disponível na Web.

Quanto aos participantes entrevistados, todos fazem parte do mesmo laboratório (i3G<sup>14</sup>), que foi também o local das entrevistas. Seu perfil profissional / acadêmico foi bastante variado: Um com formação acadêmica em Economia, um com formação acadêmica em Direito, dois com bacharelado em Biologia, e um com bacharelado em Biblioteconomia. Todos possuem experiência na construção de ontologias pelo modelo de BUENO (2005), com uso do sistema OntoSinapse, e também já trabalharam com aplicativos de visão de informações na forma de redes de relacionamento, em espaço 2D, mas não em espaço 3D.

---

<sup>14</sup> Instituto de Governo Eletrônico, Inteligências e Sistemas – i3G (<http://www.i3g.org.br/>)



Quanto ao escopo de aplicação, estes critérios de usabilidade foram usados nos três editores de ontologias existentes dentro do laboratório do i3G: OntoSinapse, Editor RR 2D, e Editor RR 3D. Os critérios foram aplicados considerando que o modelo de ontologias tratado é o mesmo para estas três ferramentas, contendo domínios (representados por envoltórias), relações semânticas (representadas por arestas), e termos (representados por vértices). O processo de construção desta ontologia também passa pelas mesmas etapas nas três ferramentas: Criação dos termos, criação das relações entre termos, e construção dos domínios que contem os termos e relações, e contem outros domínios.

A seqüência de tarefas desde a pesquisa inicial dos critérios de usabilidade até a aplicação destes critérios e compilação das respostas foi a seguinte:

- a) Compreensão dos critérios
- b) Seleção dos critérios
- c) Disponibilização dos três sistemas de construção de ontologias (OntoSinapse, Editor RR 2D, e Editor RR 3D) para os usuários a serem entrevistados
- d) Aplicação das entrevistas
- e) Compilação das respostas

Quanto à aplicação da Tabela de Critérios, tal aplicação está bastante associada ao processo de entrevistas com os usuários dos três editores, já que foi esta tabela que serviu como referencia a todo o processo de comparação de usabilidade entre os três aplicativos. As etapas de aplicação da Tabela de Critérios foram as seguintes:

- a) Preparação dos questionários na ordem da tabela de Fatores de Design (Bresciani et. al. (2008)), que foram usados como Critérios de Usabilidade
- b) Escolha da ordem de entrevistas pela disponibilidade dos participantes
- c) Os participantes foram Informados sobre a entrevista particular
- d) As entrevistas foram feitas com cada participante, isolado dos demais
- e) Para cada participante:

- a. A tabela de critérios era mostrada, com as três colunas iniciais contendo o Título do Critério, Questão Orientadora, e Descrição do Critério, mais as três colunas de resposta para cada um dos três editores de ontologias.
  - b. O usuário/entrevistado lia a descrição do critério de usabilidade a ser respondido. Caso necessário, era feita mais uma explicação verbal sobre o significado do critério e onde critério impactaria em cada um dos editores.
  - c. O usuário/entrevistado começava com um critério, e comentava suas observações verbalmente – para cada um dos três editores. Esta explicação verbal era transcrita para a célula correspondente ao par critério / editor avaliado.
  - d. Ao terminar todas as observações sobre os três editores para o mesmo critério, o processo de entrevista / transcrição passava para o critério seguinte.
  - e. O processo de entrevistas para cada par critério / editor não era forçosamente seqüencial: Caso o usuário entrevistado se recordasse de alguma observação referente a um par critério / editor já respondido, sua célula correspondente era completada com estas novas observações.
- f) Após o término das entrevistas com todos os usuários, foi feita uma compilação dos dados: Dado que cada usuário possui sua tabela de respostas (Quadros 18 até 22), foi construída uma tabela final, onde cada célula continha o acumulado de TODAS as respostas das células correspondentes ao mesmo par critério / editor nas demais tabelas.
- g) Para cada célula desta tabela final:
- a. No lugar das observações semelhantes era deixada somente UMA descrição de uma observação que representasse o conjunto destas observações.
  - b. As observações diferentes eram mantidas em suas células.
  - c. Algumas funcionalidades necessárias para cumprir com critérios de usabilidade não foram implementadas nos editores mais novos (2D e 3D). Para cada um destes casos, foi feito um comentário – pelo pesquisador – sobre o que é possível implementar nos editores para cumprir com a funcionalidade que impactaria no critério avaliado.

A tabela final com as observações dos entrevistados, e comentários sobre funcionalidades, é mostrada no Quadro 23.

Título do Critério	Questão Orientadora	Descrição ( <i>Explanation</i> )
Impacto Visual	Quão atrativa é a visualização?	Esta dimensão está relacionada com as características gráficas da visualização. O Impacto Visual é alto se a <b>visualização é capturada imediatamente</b> , se é <b>passível de ser recordada, emocionalmente passível de ser recordada</b> , ou se <b>relembra um trabalho artístico</b> . O Impacto Visual é <b>baixo</b> se a visualização é <b>simples ou de baixa qualidade</b> . Um Impacto Visual <b>alto</b> pode ser contra produtivo porque pode dispersar atenção. Uma <b>visualização prazerosa</b> é mais passível de atrair a atenção do observador e <b>criar um efeito de aura nos demais aspectos</b> da visualização, <b>inspira criatividade e dá emoções</b> , conseqüentemente <b>aumentando seu suporte mnemônico</b> .
Clareza	A visualização é facilmente compreendida com um baixo esforço cognitivo?	Clareza é alta se o significado da visualização pode ser compreendido em uma primeira vista. Ela é baixa se a visualização requer tempo e concentração de forma a ser compreendida. Baixa clareza deve ser necessária quando o conceito representado é muito complexo, e se uma alta clareza pode conduzir a uma super simplificação. Este fenómeno é relacionado com a familiaridade de audiência com as convenções, com a complexidade e completeza, e com a consistência dos elementos. <b>Dificuldade em compreender não é sempre negativa, já que a super simplificação da realidade pode conduzir à compreensão imprecisa, com conseqüentes erros em julgamentos.</b>
Trabalho Final Percebido	A visualização convida a contribuições e modificações, ou ela lembra um produto finalizado e polido?	Esta dimensão é alta se a visualização se parece com uma ilustração final ou uma figura de um livro. Quando a percepção final é baixa, a visualização lembra um croqui ou um trabalho provisório, por exemplo, feito com notas de <i>post-it</i> ou com lápis. Este tipo de apresentação influencia fortemente a boa disposição do grupo a interagir, questionar e modificar o diagrama. A percepção de trabalho terminado deveria ser consistente com a atual possibilidade de modificação.
Foco Direcionado	A visualização dirige a atenção para os itens principais da discussão?	A direção do foco é alta quando a atenção dos participantes é dirigida somente para os itens principais (relevantes) da discussão. Quando não há foco, ou o foco é disperso em muitos itens, o Foco Direcionado é baixo. Baixo Foco Direcionado pode ser útil quando são requisitados pensamentos divergentes. Um gráfico inteligente que foca em um, ou em poucos itens, pode ajudar o grupo a manter o centro da discussão. A saliência é um recurso fundamental a ser considerado, já que pode alternar a atenção do conteúdo para a forma gráfica. O foco pode ser dispersado de forma útil quando há a procura por pensamentos divergentes ou alternativas diferentes de necessidades a serem consideradas. Como conseqüência de um foco não convencional, novas descobertas podem surgir (conduzindo ao suporte de alta inferência).
Facilitação de Idéias	As novas idéias são geradas como resultado da forma de visualização?	Quando uma nova compreensão é gerada "livremente" usando visualizações ou mudando o foco (dê um passo atrás e veja a grande imagem), ou mudando as restrições de representação, então a Facilitação de Idéias é alta. Se a visualização não ajuda no ganho de novas idéias ou compreensões, ela tem uma baixa Facilitação de Idéias. Esta dimensão está relacionada com o potencial de uma visualização de ajudar a descobrir novos padrões ou relacionamentos. Ela descreve a habilidade da visualização em ajudar a pensar e a descobrir processos. Facilitar idéias é um diferenciador vital e um valor agregado na visualização sobre textos: Ela permite ganhar novas compreensões "de maneira livre" somente mudando o tipo de visualização, o foco, ou as restrições de representação.
Modificabilidade	Os itens da visualização podem ser modificados em resposta à dinâmica da discussão?	A Modificabilidade é alta se cada item da imagem puder ser mudado facilmente (mover, eliminar, substituir). Se os itens são difíceis de mudar, ou se as mudanças não são possíveis, então a Modificabilidade é baixa. Baixa modificabilidade pode ser útil quando encoraja outros modos de pensar antes que as modificações sejam feitas, enquanto a alta modificabilidade melhora a possibilidade de interação.
Suporte à Interação de Grupo	A visualização ajuda a facilitar ou estruturar a interação de um grupo de pessoas?	Esta dimensão descreve a capacidade de manter a interação de grupo e a discussão no caminho certo. Se a visualização permite indicadores tais como o rastreamento de contribuição de participantes, avaliação progressiva do desenvolvimento da discussão, reprodução e modificação simultânea / seqüencial de padrões, então o Suporte à Interação de Grupo é alta. Se não há indicadores suportados, ela é baixa. Outros elementos que providenciam Suporte à Interação de Grupo são a possibilidade de referência aos itens no diagrama (apontamento ou re-chamada) para a facilitação aos participantes, da referência aos elementos da visualização, e documentação, que permite os participantes a voltar a um certo ponto no tempo (história) ou reproduzir toda a interação.

Quadro 17: Critérios de Usabilidade.

## Entrevistado 1:

<b>Título do Critério</b>	<b>Questão Orientadora</b>	<b>Web</b>	<b>Desktop</b>	<b>3D</b>
<b>Impacto Visual</b>	Quão atrativa é a visualização?	Não agride Facilita a leitura (há uma confiabilidade de não cometer erros) Cores ajudam a lembrar a ordem dos elementos Não parece um trabalho de arte: É uma tabela, um documento, é nítido, é claro Mais um documento chat!!	Há um impacto visual: Que confusão!! Uma novidade	Parece uma arte final Uma novidade
<b>Clareza</b>	A visualização é facilmente compreendida com um baixo esforço cognitivo?	Alta clareza: Os elementos estão ordenados por tipo, e em seqüência linear Super-simplificação: não pode mostrar mais de dois níveis	Baixa clareza, há muitos elementos de uma só vez. Não há uma seqüência linear Pode mostrar mais de dois níveis	Baixa clareza pelo fato de não haver uma ordenação compreensível a princípio Necessita de ajustes. Elementos ficam concentrados em um mesmo ponto, dificultando a leitura Não há seqüência linear Relações não implementadas
<b>Trabalho Final Percebido</b>	A visualização convida a contribuições e modificações, ou ela lembra um produto finalizado e polido?	A necessidade de uso adequado das cores é perceptível. As conexões são mostradas em forma tabular	Parece um trabalho final. Existem as conexões, mas estão confusas. Existem os domínios mas se confundem com as relações.	Parece um trabalho final. Não existem as conexões. Mas é perceptível que dados termos pertencem a dados domínios.
<b>Foco Direcionado</b>	A visualização dirige a atenção para os itens principais da discussão?	Maior, não há necessidade de procuras, pelo fato de existir a ordem por tipos e seqüência linear.	Há uma dificuldade em encontrar um termo específico. Um termo focado deveria ter uma cor diferente.	Há uma dificuldade em encontrar um termo específico. Um termo focado deveria ter uma cor diferente. Há um foco artificial quanto ao efeito da profundidade (perspectiva: mais próximo está com maiores dimensões)
<b>Facilitação de Idéias</b>	As novas idéias são geradas como resultado da forma de visualização?	Documento como outro qualquer (chato!!) Bandeiras Degrades / paleta de cores (RGB, CMYK)	Lembra um mapa Conexões Sinapses	Lembrou pássaros voando Relembra matas, postes de energia elétrica Ciclones Espirais DNA Luzes de discoteca
<b>Modificabilidade</b>	Os itens da visualização podem ser modificados em resposta à dinâmica da discussão?	Há de se navegar três páginas para alterar um termo e uma relação	A inclusão / alteração é imediata	Não implementado
<b>Suporte à Interação de Grupo</b>	A visualização ajuda a facilitar ou estruturar a interação de um grupo de pessoas?	Não implementado	Não implementado	Não implementado

Quadro 18: Observações do entrevistado 1.

## Entrevistado 2:

<b>Título do Critério</b>	<b>Questão Orientadora</b>	<b>Web</b>	<b>Desktop</b>	<b>3D</b>
<b>Impacto Visual</b>	Quão atrativa é a visualização?	Mais ameno, agradável. As informações aparecem de maneira mais imediata	Este possui mais informação que o 3D (relações)	Visualização mais agradável. Parece mais artístico
<b>Clareza</b>	A visualização é facilmente compreendida com um baixo esforço cognitivo?	Mais claro. Mais organizado (ordem linear e seqüencial) Mais simples Super-simplificação: não pode mostrar mais de dois níveis	Confuso. Muita informação ao mesmo tempo. Um elemento ofusca o outro (há uma sobreposição em camadas) Pode mostrar mais de dois níveis	Muita informação ao mesmo tempo. Um elemento ofusca o outro (há uma sobreposição em camadas) Relações ainda não implementadas
<b>Trabalho Final Percebido</b>	A visualização convida a contribuições e modificações, ou ela lembra um produto finalizado e polido?	Estão mais acostumados com a visão tabular (o trabalho parece mais completo, porque as sempre havia a funcionalidade desejada, quando requisitada)	Falta uma finalização na disposição dos elementos: Um algoritmo de reposicionamento é necessário	Visualmente, dá uma impressão de término das funcionalidades (por causa do 3D), mas deve ser testado
<b>Foco Direcionado</b>	A visualização dirige a atenção para os itens principais da discussão?	É apresentada menor quantidade de informação: Maior o foco sobre um termo referencia e seus relacionados	O fato de apresentar várias relações e termos ao mesmo tempo causa fadiga e desinteresse. Muitos itens apresentados ao mesmo tempo	Baixo foco: Muitos termos apresentados ao mesmo tempo. A sobreposição dá uma leitura difícil: Perda de foco
<b>Facilitação de Idéias</b>	As novas idéias são geradas como resultado da forma de visualização?	Não muda mais: Está dentro de uma caixinha, e já está pronto e formatado. É limitado, qualquer idéia que surja pode estragar o que está feito	Apresenta uma interface extremamente desorganizada. A primeira idéia que vem é a da reorganização urgente dos dados visuais	É perceptível que o protótipo está incompleto: Dando uma liberdade de sugerir novas idéias para o responsável pela sua implementação
<b>Modificabilidade</b>	Os itens da visualização podem ser modificados em resposta à dinâmica da discussão?	Não muda mais: Está dentro de uma caixinha, e já está pronto e formatado. É limitado(Aqui conta a incerteza do funcionamento do sistema: Será que ele vai se comportar bem na próxima modificação?)	Alta modificabilidade. Termos, domínios e relações podem ser inseridos na própria interface.	A disposição dos elementos em formato espiral estimula mais o trabalho com seus dados. Dá uma idéia de uma seqüência de relações, devido à disposição espiral. A modificabilidade não foi implementada.
<b>Suporte à Interação de Grupo</b>	A visualização ajuda a facilitar ou estruturar a interação de um grupo de pessoas?	O sistema possui artifícios para identificar os responsáveis por determinada operação de construção dentro das redes de ontologia	Ainda não implementado, mas a base de conhecimento é unificada, mas não atualizada de um usuário para outro	Não implementado. Faz apenas a carga dos dados.

Quadro 19: Observações do entrevistado 2.

## Entrevistado 3:

<b>Título do Critério</b>	<b>Questão Orientadora</b>	<b>Web</b>	<b>Desktop</b>	<b>3D</b>
<b>Impacto Visual</b>	Quão atrativa é a visualização?	Parece uma planilha eletrônica: Muito comum	Tudo é quadrado, parece chão de fábrica	É agradável trabalhar com a impressão da expansão da realidade  É agradável perceber que os termos de um domínio estão dentro de uma envoltória colorida e suave
<b>Clareza</b>	A visualização é facilmente compreendida com um baixo esforço cognitivo?	Não é perceptível que haja vários termos dentro de um domínio: É necessária a rolagem da barra na mesma página, ou navegar entre várias páginas  Também não é perceptível o mesmo termo pertencer a dois ou mais domínios diferentes	Há muita dificuldade em trabalhar com domínios extensos: Os vértices se sobrepõem, dificultando a visão ampla.	Se percebe a intersecção entre domínios e o número de termos  De imediato percebe-se que há poucos ou muitos termos dentro de um domínio
<b>Trabalho Final Percebido</b>	A visualização convida a contribuições e modificações, ou ela lembra um produto finalizado e polido?	Parece o que está melhor acabado: possui várias informações condensadas em uma mesma interface	As informações visuais parecem inacabadas	Falta um nomeador para as envoltórias dos domínios, com o formato semelhante ao formato da envoltória: Transparência, Cor e com uma fonte de tamanho maior
<b>Foco Direcionado</b>	A visualização dirige a atenção para os itens principais da discussão?	Alto foco, a interface de termos e relações mostra relações com um termo referencia somente. A desvantagem aqui é que a edição exige a navegação por vários procedimentos	É melhor que o Web porque é possível fazer a edição na própria interface	É melhor que a desktop porque o usuário pode focar sua visão sobre um determinado vértice sem alterar a posição relativa deste em relação aos outros vértices, basta girar o universo virtual
<b>Facilitação de Idéias</b>	As novas idéias são geradas como resultado da forma de visualização?	Edição difícil  Percepção difícil do conteúdo e domínio	Fácil edição  Visão de um número razoável de termos dentro de um determinado domínio, mas decresce a qualidade para uma quantidade muito grande de termos	Edição mais fácil ainda, devido às cores  Há o artifício de girar o ambiente sem alterar a posição relativa entre termos
<b>Modificabilidade</b>	Os itens da visualização podem ser modificados em resposta à dinâmica da discussão?	Muitas páginas para alterar termos relações e domínios. Ir e voltar, abrir várias janelas	É feita dentro da própria interface	Alta flexibilidade / maleabilidade  Feito dentro da própria interface  Vantagem da profundidade para classificar blocos de dados
<b>Suporte à Interação de Grupo</b>	A visualização ajuda a facilitar ou estruturar a interação de um grupo de pessoas?	Implementado	Não implementado	Não implementado

Quadro 20: Observações do entrevistado 3.

## Entrevistado 4:

<b>Título do Critério</b>	<b>Questão Orientadora</b>	<b>Web</b>	<b>Desktop</b>	<b>3D</b>
<b>Impacto Visual</b>	Quão atrativa é a visualização?	Trata-se de uma planilha formatada para acesso web. Organizada, porém quando em maior volume, de difícil análise.	Uma rede de relacionamentos permite uma melhor visualização dos termos, porém meio confuso quando há muitas relações entre eles.	Visualmente melhor apresentável.
<b>Clareza</b>	A visualização é facilmente compreendida com um baixo esforço cognitivo?	Por ser uma planilha simples, há muita dificuldade em trabalhar a organização e relações de um grande número de termos.	Inicialmente percebe-se uma melhora na visualização dos termos e suas relações, se comparada a planilha, porém, quando há um grande volume de termos, é preciso consultar a planilha para confirmar as relações entre eles.	Não consigo visualizar a relação entre os termos através deste modelo.
<b>Trabalho Final Percebido</b>	A visualização convida a contribuições e modificações, ou ela lembra um produto finalizado e polido?	A interface remete a um relatório, dando a impressão de um trabalho finalizado, pronto para ser replicado ou impresso.	Tenho a impressão de que ela serve de apoio ao analista no processo de construção de ontologias. Não me parece um trabalho final.	Não consigo perceber.
<b>Foco Direcionado</b>	A visualização dirige a atenção para os itens principais da discussão?	Quando tratados individualmente, parece ter maior foco na apresentação das relações.	Com relação a edição, se apresenta uma melhor alternativa, pois é possível visualizar todas as relações em um único ponto.	Precisa ser melhor trabalhado, pois suas reais funções não são facilmente percebidas pelo usuário.
<b>Facilitação de Idéias</b>	As novas idéias são geradas como resultado da forma de visualização?	Fácil edição, pois ele trabalha com caixas de texto.	Fácil edição, pois as relações são mais facilmente visualizadas pelo analista.	Por ser 3D, me parece que o giro em torno do eixo facilita a visualização, porém há dificuldade em identificar as relações entre os termos.
<b>Modificabilidade</b>	Os itens da visualização podem ser modificados em resposta à dinâmica da discussão?	O analista precisar abrir várias páginas para fazer as modificações e depois confirmá-las.	As modificações são feitas na interface, sem a necessidade de abrir muitos campos.	Deve seguir o mesmo conceito da versão desktop.
<b>Suporte à Interação de Grupo</b>	A visualização ajuda a facilitar ou estruturar a interação de um grupo de pessoas?	Implementado.	Em desenvolvimento, porém não implementado.	Não implementado.

Quadro 21: Observações do entrevistado 4.

## Entrevistado 5:

<b>Título do Critério</b>	<b>Questão Orientadora</b>	<b>Web</b>	<b>Desktop</b>	<b>3D</b>
<b>Impacto Visual</b>	Quão atrativa é a visualização?	Limita a visão do domínio (todos os relacionamentos deste)	Permite uma visão mais completa, amigável. Baixa mobilidade	Alta mobilidade, é possível ver os elementos comuns a dois ou mais domínios diferentes. Se espera que tenha mais recursos
<b>Clareza</b>	A visualização é facilmente compreendida com um baixo esforço cognitivo?	Falta ver o conjunto	Maior clareza porque já conhece a rede de ontologia. Serve como uma ferramenta de confirmação dos conteúdos adicionados anteriores (Já tem o conceito do que é a ontologia como rede de relacionamento)	Maior clareza porque já conhece a rede de ontologia. Serve como uma ferramenta de confirmação dos conteúdos adicionados anteriores (Já tem o conceito do que é a ontologia como rede de relacionamento)
<b>Trabalho Final Percebido</b>	A visualização convida a contribuições e modificações, ou ela lembra um produto finalizado e polido?	Finalizada de acordo com o projeto inicial Exigência: Bastava a listagem de termos. Cores não eram necessárias à época. Era difícil ver a incontinência de termos a domínios e desconexão entre termos.	Percebe-se uma maior completeza de dados, comparando à visão inicial. O importante é o contexto. Deve-se ver o conjunto dentro do contexto, e o importante é ser possível ver o que ainda está fora do contexto, ou desconectado.	Percebe-se uma maior completeza de dados, comparando à visão inicial. O importante é o contexto. Deve-se ver o conjunto dentro do contexto, e o importante é ser possível ver o que ainda está fora do contexto, ou desconectado.
<b>Foco Direcionado</b>	A visualização dirige a atenção para os itens principais da discussão?	Há manutenção do foco devido à disposição em forma de lista ordenada dos termos	Há dificuldade na procura de elementos isolados porque são em grande quantidade, e não há uma organização visual à priori, mas é possível ver quem se relaciona com quem, em vários níveis	Há dificuldade na procura de elementos isolados porque são em grande quantidade, e não há uma organização visual à priori, mas é possível ver quem se relaciona com quem, em vários níveis
<b>Facilitação de Idéias</b>	As novas idéias são geradas como resultado da forma de visualização?	Nota-se que o documento é estático, sem visão ampla. Prefere-se trabalhar em Excel, pois a planilha não termina nunca: Sempre é possível acrescentar algo.	Há a segurança de que, apresentado um potencial, uma vontade, uma necessidade, ela será satisfeita.  Traz a idéia da conexão, do mapa, que é possível expandir os elementos que já existem	Há a segurança de que, apresentado um potencial, uma vontade, uma necessidade, ela será satisfeita.  Traz a idéia da conexão, do mapa, que é possível expandir os elementos que já existem
<b>Modificabilidade</b>	Os itens da visualização podem ser modificados em resposta à dinâmica da discussão?	Há de se cumprir um procedimento para alterar incluir ou eliminar algum elemento.	As operações de modificação da rede ocorrem sem a necessidade de navegação entre várias interfaces.	As operações de modificação da rede ocorrem sem a necessidade de navegação entre várias interfaces.
<b>Suporte à Interação de Grupo</b>	A visualização ajuda a facilitar ou estruturar a interação de um grupo de pessoas?	É possível verificar quem é o responsável por determinada operação. Existem interfaces de confirmação pelo administrador do sistema.	Ainda não implementado	Ainda não implementado

Quadro 22: Observações do entrevistado 5.



## Avaliação das entrevistas

<b>Critério</b>	<b>Web</b>	<b>Desktop RR 2D</b>	<b>Desktop e Web RR 3D</b>
<b>Impacto Visual</b>	Muito comum: Parece uma planilha	Confuso, com muitas relações	Parece uma arte final. Dá a dimensão real do domínio, tanto no aspecto espacial quanto de conteúdo
<b>Clareza</b>	Alta clareza devido à ordem Muito simples: Não se sabe se há muitos termos em um domínio	Mostra mais de dois níveis de pertinência (domínios e subdomínios) Sobreposição de vértices	É possível dispor os termos em planos diferentes, para diferenciação de grupos
<b>Trabalho Final Percebido</b>	Visão tabular é vantajosa	Conexões confusas Serve como ferramenta de apoio aos analistas	Percebida a pertinência de termos a domínios Dá uma impressão de que as funcionalidades estão terminadas Falta um nome para as envoltórias que representam os domínios com suas características de cor, transparência e tamanho Percebe-se o quanto a ontologia está completa, e o quanto é extensa Percebe-se o contexto (os termos que estão conectados), e o que falta (os termos não conectados)
<b>Foco Direcionado</b>	Há foco por facilitar procuras no formato tabular	Dificuldade em procurar um termo ou relação. Muitas sobreposições. São necessárias muitas operações de arraste para resolver a sobreposição.	A navegação de giro é usada para reduzir a sobreposição de termos e relações. Com ela são feitas menos operações para reduzir esta sobreposição. Artifício de alteração particularizada de elementos, filtragem, zoom, uso de diferentes planos no espaço 3D
<b>Facilitação de Idéias</b>	Documento estático	Fácil edição Interface desorganizada	Como os elementos da ontologia estão ligados a objetos 3D que representam a realidade com maior fidelidade, as idéias de utilização da informação são geradas de maneira natural e consistente
<b>Modificabilidade</b>	Muitas páginas para alterar: Sem necessidade	Operadores de edição são imediatos	Operadores de edição são imediatos e sensíveis ao contexto (WYSIWYG)
<b>Suporte à Interação de Grupo</b>	Possui artifícios de identificação de responsáveis O sistema só avisa que um elemento já existe na base quando o usuário já terminou de inclusão do elemento	É possível associar ícones de usuários a termos, domínios e relações (identificação de responsáveis, de forma gráfica)	Disponibilidade Web via <i>Applet</i> ou <i>Web Start</i> O sistema avisa sobre a existência de determinado elemento ainda durante a operação de criação deste, e antes da consolidação da operação É possível associar ícones de usuários a termos, domínios, relações e objetos 3D (identificação de responsáveis, de forma gráfica)

Quadro 23: Avaliação das entrevistas.

## 7 DISCUSSÃO

Esta discussão pretende aproximar a literatura revisada com os resultados obtidos, descritos no decorrer desta tese. Para tal, segue-se a ordem dos objetivos específicos: a) Apresentar modelos de representação de conhecimento; b) Apresentar metodologias de representação de conhecimento; c) Apresentar ferramentas de construção de ontologias, e máquinas de busca, suas evoluções/tendências; d) Apresentar bibliotecas de visão de informação, e suas evoluções/tendências; e) Descrever os critérios de usabilidade de software para executar a comparação entre os modelos tabular, bidimensional e tridimensional; f) Definir uma metodologia para o desenvolvimento de um modelo gráfico de visão de ontologias; g) Apresentar um modelo de domínio sobre o qual será desenvolvido este modelo gráfico; h) Criar modelos matemáticos para representar os formatos de visualização de informações, na forma de redes de relacionamento e conjuntos (RRC); i) Criar protótipos para testes de viabilidade de representação gráfica das ontologias em ambiente 2D e 3D; j) Comparar e avaliar os modelos tabular, bidimensional e tridimensional de visão de elementos de ontologias; k) Apresentar uma proposta de construção de ontologias multilíngües e um método de busca multilíngüe sobre as quais serão aplicadas as ontologias construídas.

As tendências da Web Semântica prevêm, de acordo com Berners-Lee *et. al.*, 2001, a composição dos dados da Web através de afirmações e regras, utilizando aplicativos que combinem dados entre si, para a dedução de informações que possam estar implícitas nos grupos de páginas, através de inferências.

Nesse sentido, os autores comentam sobre algumas semelhanças temporais de conceito operacional entre a Web original e o surgimento da Web Semântica. De início, eles enfrentaram problemas de ceticismo quanto à capacidade de comunicação das várias fontes de dados e quanto à possibilidade de encontrar informações em tempo real, o que nos dias de hoje já é considerado normal. Numa perspectiva de futuro, um ponto interessante, que ajuda na construção de futuras soluções para a Web Semântica, é comparar o uso da Web original com a Web Semântica que está por vir.

Na antiga forma da Web, durante a procura e combinação de dados para compilar uma informação final, os usuários executam quatro

processos básicos de construção da informação: 1) Determinar a informação que se deseja obter; 2) Procurar na Web as páginas que contém as informações; 3) Do resultado dessa busca, são descritas as páginas com as informações que mais se adaptam na produção do objetivo final; 4) São compiladas as informações.

Diante disso, o que se pretende é modelar, através de assertivas, o objetivo, as características das páginas a serem consultadas, os critérios de seleção considerados na escolha dessas páginas e, finalmente, sobre como os resultados devem ser combinados para dar a resposta desejada. Outro ponto a ser considerado é a capacidade de alimentar agentes inteligentes com os novos modelos. Assim, os agentes poderão resolver os problemas criados originalmente e, com isso, deduzir novas informações a partir das páginas consultadas e também das páginas que não haviam sido propostas no primeiro modelo da solução.

O que se observa na modelagem por objetos e assertivas é que, na medida em que cresce o número de elementos na descrição do modelo, sua gestão torna-se mais complexa. O motivo é que este modelo, com algumas exceções, é descrito por frases textuais ou por elementos visuais limitados ou com pouca interação com o usuário. Desse modo, uma solução para o aumento do número de componentes de uma ontologia seria a divisão dos componentes em blocos funcionais. Entretanto, isso exigiria uma visão geral dos blocos e de seus componentes internos.

Já o formato *Resource Description Framework* (RDF) foi projetado com o objetivo de descrever elementos da Web na forma de triplas ou sentenças simples, compostas de Sujeito, Predicado e Objeto. Esta representação tem a vantagem de simplificar a codificação e armazenagem, sendo possível sua representação visual através de redes de relacionamento. E no caso do RDF descrever objetos concretos, seus elementos descritivos podem ser associados a objetos do espaço 3D. Outra vantagem do RDF é que o modelo pode ser descrito com o uso de envoltórias. Essas envoltórias facilitam a identificação dos grupos de elementos (representados por outras triplas RDF) que sirvam de valor para propriedades de Sujeitos, ao contrário de ver os grupos acoplados a uma propriedade apenas como um ramo de uma rede de relacionamentos.

Quanto à OWL, ela é dividida em três níveis: Lite, DL (*Description Logics*) e Full, tendo suas características de representatividade, completude e computabilidade bem determinadas.

Inicialmente, esses níveis servem para determinar o escopo da representação lógica de um problema e ter uma métrica prévia do esforço a demandar em sua modelagem.

Nesta tese, foram realizados experimentos sobre modelagem de problemas com uso da lógica descritiva disponível na *framework* OWL-DL, utilizando a ferramenta Protege, através da sintaxe SWRL, para a definição de regras precedente/conseqüente, e das máquinas de inferência (Fact++, Hermit, Racer e Pellet), para a geração de conclusões a partir de premissas. Dessa maneira, foi possível entender o funcionamento da sintaxe OWL-DL. Essas experiências permitiram identificar que a sintaxe OWL-DL não possui operadores para a contagem de indivíduos ou classes, no sentido de possibilitar que as máquinas de inferência comparem valores e concluam sobre a veracidade das afirmativas que resultam ou partam da contagem de classes, indivíduos ou propriedades.

Verificou-se que existe uma ferramenta inicial (*TwoUse, 2011*) para a visão de elementos de ontologias construídas na sintaxe OWL-DL, na forma de redes de relacionamento, no espaço 2D, com uso de objetos da representação UML para a representação. Considerou-se, então, a possibilidade de representar a visão de ontologias no espaço 3D e de associar os elementos dessas instâncias de ontologias a objetos no espaço 3D, com o objetivo de: a) Facilitar a comunicação de conceitos e relações entre pessoas; b) Associar as instâncias OWL de ontologias com o formato UNL de representação normalizada de conhecimento.

Dessa forma, têm-se dois canais de uniformização na compreensão da maneira de como uma máquina faz suas inferências e chega às suas próprias conclusões. O primeiro canal é a visão realista, tridimensional de conceitos, pelos participantes do processo de construção de ontologias. O segundo canal de uniformização é a visão de conceitos e relações OWL em vários idiomas através da conversão dos conceitos entre si, com o uso das Palavras Universais e relações da UNL.

Cabe lembrar que os componentes da tecnologia UNL de representação de conhecimento multilíngüe são as Palavras Universais, que representam conceitos que podem ser equivalentes em qualquer idioma, e as relações léxicas (de papel destes conceitos), dentro de uma sentença UNL.

Segundo Cardeñosa *et. al.* (2008), a linguagem da UNL tem um número finito de relações léxicas. Considerando que as ontologias multilíngües podem conter representações gráficas das relações léxicas, procurou-se pesquisar essas formas. Observou-se que, para tal, é necessária a atuação de profissionais de design. Esses profissionais permitirão que haja uma comunicação rápida dos conceitos da ontologia multilíngüe e como elas estão relacionadas lexicalmente.

Sobre as equivalências entre as palavras universais, relações léxicas e atributos da UNL, e as classes e indivíduos, propriedades de objeto, propriedades de dados e restrições da OWL, os autores apontam a necessidade de construir ferramentas que permitam fazer: a) O intercâmbio entre os dois formatos de representação (UNL = OWL); b) A dedução de informações de descrições feitas em vários idiomas, no sentido idioma => UNL => OWL; c) O intercâmbio de objetos e premissas feitos originalmente na sintaxe OWL, em vários idiomas, com o uso das palavras universais e relações da UNL; d) Aplicação de artifícios de inferência para que as máquinas possam transformar um texto redigido em linguagem natural para a sintaxe UNL e vice-versa.

Em qualquer um dos casos acima, ainda necessário a realização de mais pesquisas, na procura de situações que se possam aplicar as equivalências de inferência entre as duas formas de representação.

Por último, foi estudada a tecnologia RC2D (HOESCHL, 2001). Essa tecnologia permite a comparação e busca em um *corpus* de documentos, na qual se aplicam técnicas de classificação de documentos dentro de domínios determinados, a partir do uso de um dicionário de palavras controladas, e de técnicas de Raciocínio Baseado em Casos (RBC), para a comparação da similaridade entre documentos a partir de critérios pré-definidos.

Nota-se que, para Bueno (2005) e Hoeschl (2001), a comparação de similaridade entre documentos é feita a partir de técnicas de RBC, em que, para cada documento, é dado um valor numérico. Essa abordagem possui três vantagens: 1) A simplicidade do tratamento do problema de comparação de similaridade entre dois documentos através de valores no campo dos números reais; 2) A determinação de um índice (comparável) de um texto, através de um processo de sucessivas classificações, chamadas de domínios de conhecimento. Este artifício foi planejado com o objetivo de reduzir a ambigüidade de um mesmo termo, caso este seja usado para comparar dois documentos de contextos diferentes; 3) O envolvimento de profissionais na seleção de palavras que fazem parte de

um vocabulário controlado facilita o processo de comunicação entre os profissionais.

Os trabalhos de Bueno (2005) e Hoeschl (2001), além de darem sustentação à proposta desta tese, oferecem uma forma de fazer com que um conceito torne-se comum entre vários participantes do processo descritivo de documentos; possibilitam também representar um conjunto de conceitos limitados por envoltórias, permitindo uma visão dos domínios do conhecimento.

Nas ferramentas para construção de ontologias e construção colaborativas de ontologias, foram observadas as seguintes tendências: armazenagem de ontologias em servidores remotos; aplicação Web; compartilhamento de recursos para organização, rotulação e referências (BibSonomy); histórico de conceitos e associação de informações aos componentes das ontologias (Collaborative Protègè, Ontowiki, Soboleo); edição de bases de conhecimento de forma colaborativa e dedução de informações (DBin, Protègè); operações de captura e liberação (checkin, checkout) para edição de partes da ontologia (Hozo); uso de Widgets (Ontowiki); edição de conceitos multilíngues (IATE, NeOn); construção de informação hierárquica; integração de módulos de ontologias; identificação de conflitos; sistemas de sugestão e votação sobre partes das ontologias construídas; aplicativos que integram listas de discussão e *chats*; registro de *logs* de operações e interfaces personalizadas.

Para as ferramentas de visão de informação, suas evoluções e tendências, foi apresentada a *framework* ‘redes de relacionamento’ (RR). Ela é usada para construir aplicativos de visão de dados, como: a primeira versão do editor de ontologias e um aplicativo de visão de dados de telefonia e transações financeiras em redes criminais (STRADIOTTO *et. al.* 2004, 2006a, 2006b, 2006c). Também foi descrita a biblioteca de visão de dados Prefuse, aplicável a dados de qualquer natureza (HEER *et. al.*, 2005).

Para o caso da *framework* RR, sua relação com ontologias ocorre por meio da representação dos elementos representativos de ontologias: os elementos atômicos, como classes e indivíduos; e elementos relacionais, como as propriedades, através de vértices e arestas, respectivamente. Para o caso da biblioteca Prefuse, a equivalência também é verdadeira, já que a representação de dados através de RR é uma caso particular disponibilizado pela biblioteca.

A ferramenta de busca Clusty (VALDES-PEREZ *et al.*, 2008a, 2008b; e WATZMAN *et al.*, 2008) apresenta os dados de busca agrupados por temas, em vários níveis. Considerando que um dos objetivos do uso de ontologias é (após uma busca de páginas por palavra-chave) permitir a classificação dessas páginas por assuntos principais e disponibilizar uma lista de páginas encontradas através de uma árvore com os assuntos, acredita-se que a ferramenta Clusty satisfaz esse critério de classificação.

A ferramenta Cluuz, de busca de documentos, apresenta os resultados de três maneiras: 1) Links para as páginas encontradas; 2) Imagens correspondentes às palavras-chave procuradas; 3) Uma interface de redes de relacionamento (também chamada de *tag-cloud*, ou nuvem de rótulos) mostrando as conexões de páginas através de suas palavras-chave em comum.

Combinando as funcionalidades das ferramentas, pode-se obter uma ferramenta de busca que combine visão de temas conectados por palavras-chave, através de uma RR, e agrupamento de vértices representativos de páginas dentro do espaço de redes. Tem-se, então, um resultado desse procedimento, ou seja, uma busca na qual é permitida a circunscrição (ou agrupamento) de resultados comuns através de uma envoltória. Tal espaço de resultados poderia ser representado por meio de uma RR (por palavra-chave comum), em que os vértices com suas páginas correspondentes, com um assunto comum, estariam circulados por uma envoltória. Caso o assunto inicial fosse uma particularização de um assunto mais genérico, o primeiro assunto estaria envolto por uma envoltória que representasse aquele assunto mais genérico, e assim por diante, em sucessivos níveis de agrupamento. Tal representação seria possível através de uma biblioteca que permitisse a visão de dados na forma de RR e conjuntos (RRC). Essa funcionalidade já é disponibilizada por Heer *et al.* (2005), em que um grupo de vértices pode ser envolto por uma casca convexa (*Convex Hull*) construída a partir do algoritmo de Graham (1980), chamado *Graham Scan*.

A *framework* de visão de dados Alphaworks (WATTEMBERG, 2005) sugere três direções a serem tomadas por projetistas de ferramentas de visão de informações: 1) A visão de textos e conversação; 2) A visão de fluxo histórico de dados; 3) A extração de dados de redes sociais. Os dados capturados ou produzidos a partir dessas três formas de visão estão sujeitos a processos de mineração, para descoberta de informações escondidas.

Quanto ao processo de descoberta de informações a partir de textos ou diálogos, informações referentes a acontecimentos ou processos podem ser vistas a partir de ferramentas de visão de argumentação (*Argument Visualization*) aplicadas a listas de discussão ou e-mails. Esta visão de argumentação sobre os diálogos produzidos pode ser feita através de um processo de *parsing* de textos, e posterior aplicação da lógica sobre os argumentos, através de instâncias de ontologias. Cardeñosa *et. al.* (2008) já identificou a possibilidade de uso da UNL, com suas características semelhantes à OWL, para análise léxica e capacidades de dar respostas a questões corriqueiras.

Quanto à visão de histórico de dados, seguindo a previsão de Watterberg (2005), atualmente se observa ferramentas web de métricas de ocorrência de palavras-chave em redes sociais, como o Twitter, que disponibiliza ferramentas grátis para essa finalidade. Caso a solução de interpretação de textos com ontologias for aplicada, combinada com o recurso de métrica, será possível, no futuro, medir a quantidade de ocorrências de uma palavra em uma rede específica e medir a ocorrência de um assunto, independente dos termos utilizados para descrevê-lo.

Quanto ao processo de extração de dados de redes sociais, atualmente se fala em um processo chamado *Croud-Sourcing*, que se caracteriza por dar informações sobre tendências de redes sociais, baseadas em métricas de grafos. Este processo acontece com páginas de postagem de informação, como Twitter e Facebook, que oferece uma plataforma para desenvolvimento de aplicativos sobre as redes sociais construídas dentro dela.

Outra ferramenta de visão de dados é a plataforma M3O, disponibilizada em *Cloud-Computing*, da Vitria Technologies, que permite, de maneira visual, construir um processo de produção baseado em serviços Web, e posteriormente controlar o andamento dos serviços através de *dashboards* (gráficos de barras, gráfico em setores).

A ferramenta Growl possibilita apresentar dados de ontologias no formato OWL, como uma RR, mostrando, de forma completa, a cadeia lógica produzida dentro de uma ontologia, cuja apresentação dos dados é compartimentada em diferentes blocos (tabs), em forma de árvore ou lista, que, por sua vez, apresentam seus limites sobre a visão completa de dados das ontologias construídas.

Assim, a construção de uma ferramenta de visão de ontologias, também como RR, nos espaços 2D e 3D, poderá apresentar avanços na



representação visual de objetos e operadores lógicos das ontologias construídas, em qualquer um dos quatro formatos apresentados: UNL, UML, OWL e o modelo de Bueno (2005), principalmente porque tais objetos e operadores poderão ser representados por cópias digitais de objetos da realidade através de metáforas, auxiliando na comunicação universal de conceitos e operadores. Há também a ferramenta *TwoUse*, construída sobre a plataforma IBM Eclipse, que combina objetos de representação UML para descrever ontologias no formato OWL 2.0.

Já a *Universal Words Dictionary* e os dicionários Online apresentam vantagens no planejamento da ferramenta de construção de ontologias em virtude de disponibilizarem grande quantidade de dados, organizados por sua semântica e por seu léxico, a qualquer tempo e lugar. As principais vantagens práticas são: a) Não há a preocupação sobre armazenagem local de dados, havendo economia de espaço; b) Os bancos são redundantes, sendo possível confirmar a veracidade de significados, ou papéis, de uma página de conceitos, com outra página de conceitos; c) Caso uma das bases não estiver disponível, ela poderá ser substituída por outra base. A desvantagem está no acoplamento das bases a um futuro editor de ontologias, que seria a compreensão (ou a falta dela) da forma como os dados entram (parâmetros de endereço, ou preenchimento de caixas de texto?) ou saem (o formato HTML resultante da página de resposta), com a conseqüente produção de extensões diferenciadas para a comunicação do futuro editor com cada um dos dicionários em particular. Observando as variedades de formatos Web de dicionários por um lado positivo, o acúmulo de formatos de comunicação leva à solução da parametrização e reuso do código, o que permite a comunicação do editor com as páginas nestes formatos.

Para os critérios de usabilidade de software para executar a comparação entre os modelos tabular, bidimensional e tridimensional, utilizou-se o entendimento de Bresciani *et. al.* (2008) que descreveu uma *framework* para avaliação de usabilidade de aplicativos. Esses critérios serviram de base para a comparação de usabilidade entre os três sistemas de construção compartilhada de ontologias, que foram analisados e aplicados nesta pesquisa. Observou-se que cada um dos critérios possui as seguintes propriedades: a) É iniciado a partir de uma pergunta básica; b) Cada critério é uma junção de critérios semelhantes coletados por vários pesquisadores na área de design de software; c) Cada critério pode ter nomes variados, dependendo dos pesquisadores que os utilizam; d) Cada critério possui propriedades que os diferenciam dos demais critérios, mas pode ter dupla interpretação, caracterizando a

existência de dimensões similares; f) Cada critério não tem uma métrica exata, mas valores aproximados (de péssimo a excelente), que podem depender da opinião de cada observador que mede a interface que lhe é apresentada; g) O ganho em um critério pode acarretar na perda em um critério complementar (o que os autores chamam de *trade-off*). Portanto, ao chegar a um consenso sobre a qualidade de uma interface, deve-se tomar o cuidado de não sacrificar demais um critério, em detrimento de outro que tenha sua avaliação excelente. Um equilíbrio deve ser encontrado; h) Devido a uma possível dificuldade de compreensão dos critérios por leigos, os autores observaram a necessidade de apresentar exemplos de bons e maus valores para cada um dos critérios.

Blackwell *et. al.* (2001) enumeram fatores que ajudam a transformar métodos de visualização em métodos de visão de informações (Quadro 10, página 93). Esses fatores podem ser utilizados para justificar valores altos e baixos dos critérios dos Quadros Quadro 8 e Quadro 9 e servem para definir um vocabulário de avaliação de interfaces entre diversos pesquisadores.

Neste estudo, a metodologia de construção de modelos gráficos de editores de ontologias foi composta pelas seguintes etapas: 1) Determinação do espaço de trabalho (2D ou 3D); 2) Determinação do modelo de domínio das ontologias a serem trabalhadas pelo editor; 3) Determinação do vértice: listagem, desenho, manipulação, elementos anexados; 4) Determinação das arestas: listagem, relacionamentos com vértices, posicionamento, manipulação, elementos anexados, relacionamentos com arestas; 5) Determinação das envoltórias: listagem, relacionamentos com vértices, posicionamento, manipulação, relacionamento com outras envoltórias, relacionamento com arestas; 6) Transformação de todos os elementos simultaneamente: translação, rotação, zoom; 7) Seleção de Grupos: criação do processo de seleção, marcação e desmarcação dos elementos selecionados; 8) Determinação de parâmetros geométricos da rede de relacionamentos: padronização dos parâmetros, criação de meios de alteração dos parâmetros pelo usuário e respectivas interfaces; 9) Criação de métodos de importação e exportação de dados da rede de ontologias para sistemas de arquivos ou para bases de dados; 10) Remodelagem do domínio para a inclusão de objetos 3D nas instâncias de ontologias; 11) Pesquisa de formatos de representação de objetos 3D; 12) Criação de métodos de importação de múltiplos objetos 3D completos para dentro de um ambiente virtual do editor; 13) Programação de transformações sobre o conjunto objetos 3D + rede de ontologias: translação, rotação e zoom; 14) Programação da

manipulação particularizada de cada objeto 3D; 15) Programação do processo de seleção particularizada de cada objeto 3D; 16) Teste com várias instâncias do editor, em máquinas diferentes, dentro de uma LAN; 17) Programação do processo de comunicação e atualização de dados dos editores entre si e com a base de conhecimento; 18) Criação de métodos de exportação de objetos 3D para um sistema de arquivos e/ou base de dados; 19) Disponibilização das instâncias do editor em páginas Web; 20) Uso experimental de várias instâncias do protótipo em uma LAN.

Os modelos de domínio da Engenharia da Mente serviram de base para o modelo proposto do editor 3D. Os primeiros modelos de domínio da Engenharia da Mente foram feitos para apenas um idioma e apresentavam os domínios de ontologias em dois níveis: um domínio ou subdomínio pode conter vários termos, assim como um termo pode estar contido em vários (sub)domínios. Dentro de um (sub)domínio, os termos contidos podem ser sinônimos entre si ou manter relações de metonímia (parte) ou hiperonímia (classe). Quanto às palavras universais, estas podem ser equivalentes de um termo.

No segundo modelo de domínio, os domínios também foram representados por palavras universais, e a pertinência de domínios a um idioma. Dessa forma, as várias instâncias de ontologias puderam ser construídas em idiomas diferentes, e conectadas a uma única ontologia normalizada, representada somente por palavras universais. Os domínios foram apresentados em dois níveis, no máximo.

O modelo de domínio proposto nesta tese apresenta a variante de pertinência de um domínio a múltiplos domínios, e, reciprocamente, a continência de vários domínios a um domínio. Um domínio pode conter outro domínio em múltiplos níveis, e não somente dois como no modelo de Bueno (2008). No modelo proposto, existe a ontologia normalizada, composta por palavras universais como elementos conceituais atômicos e por domínios determinados por palavras universais. Assim, uma instância normalizada de uma ontologia pode ser conectada a uma instância equivalente da mesma ontologia em diferentes idiomas.

Quanto ao papel dos objetos tridimensionais na ontologia, um objeto 3D pode estar associado a um conceito normalizado, para casos em que haja comum acordo entre os usuários sobre o conceito que representará aquele objeto, dentro do espaço virtual, ou estar associado a um termo, dentro de um domínio, em um idioma determinado. Um objeto 3D mais genérico pode estar relacionado a outros objetos mais

específicos, como uma classe, ou ainda um objeto pode estar relacionado aos seus objetos partes.

A construção de modelos matemáticos para a visão das ontologias foi efetivada com a geração de idéias de representação gráfica de tipos de relacionamento, com a geração dos elementos básicos de representação visual de informação (Quadro 11, página 101), e depois com a combinação das formas de representação (Tabela 3, página 103). A descrição dos modelos estruturais foi detalhada, inicialmente, com a apresentação de um diagrama de relações entre classes que pertencem à *framework* usada para a construção do editor 2D de ontologias (Figura 36, página 107), e depois um quadro indicando a função de cada uma das classes (Quadro 12, página 108).

A necessidade de construção de modelos matemáticos surgiu a partir da visualização dos resultados do trabalho de Bueno (2005). Posteriormente, com o aprendizado das estruturas de bases de dados e da linguagem SQL, foi observada a possibilidade da representação gráfica das entidades em larga escala. Tendo o modelo de domínio na base, a estrutura correta no aplicativo e o modelo matemático correspondente a essa estrutura, foram representados todos os elementos necessários da ontologia, tanto no espaço bidimensional quanto no espaço tridimensional.

A geração de ideias para a construção do modelo estrutural partiu de três visões: a) Do espaço vazio; b) Da visão de um termo como vértice; c) Da visão da conexão de dois termos com uma relação semântica, como uma aresta (bi) direcionada, conectando dois vértices; d) Da visão de um domínio como uma envoltória.

A ordem de complexidade desses elementos teve a seguinte visão: do mais simples, ou o espaço vazio que não contém nada; depois o vértice, por ser representado por um ponto, ou um elemento indivisível; depois a aresta, representada por uma linha reta, com conexão em dois pontos, em seu modo mais simples; posteriormente a envoltória, representada em sua forma mais básica como um círculo circunscrito a todos os pontos que contivesse.

A partir desses elementos básicos (vazio, vértice, aresta e envoltória), foi imaginada a possibilidade da multiplicidade dos elementos: primeiro, relacionando-os consigo mesmos; depois, relacionando-os com os elementos de maior complexidade.

Quanto ao posicionamento relativo dos elementos, o vértice, embora sua forma básica pareça um ponto, ou um elemento indivisível, considerou-se que, para qualquer vértice, ele tenha uma geometria extensa, não sendo mais um ponto, e também com o fato de que uma aresta estar conectada ao vértice. A ideia de local de conexão entre vértice e aresta abriu um espaço para a interpretação do vértice como um elemento geométrico qualquer, e não somente um ponto; neste caso, a de que necessitasse de um parâmetro de posição (no espaço bidimensional ou tridimensional) onde estivesse sendo conectado por uma ou várias arestas.

Sobre as arestas, a atenção recaiu sobre suas características geométricas, que poderiam ser parametrizadas, já que a aresta representa uma ponte entre dois pontos, com uma origem e um destino. Em muitos casos de representação gráfica, elas são definidas normalmente por cores, espessuras e tracejados; então, questionou-se: porque não lhes diferenciar através da sua própria trajetória, entre os pontos de origem e destino? Por isso, foi adicionado mais uma propriedade geométrica à aresta, que auxilia a diferenciá-la das demais que estarão presentes na representação visual de uma ontologia, ou seja: sua trajetória.

Ainda aproveitando o raciocínio sobre a possibilidade de posicionamento de uma aresta sobre um vértice, pode-se estender este raciocínio para o posicionamento de uma aresta sobre o ponto de uma outra aresta. Já que a aresta possui uma extensão, e já que ela pode ter uma geometria definida, foi imaginada a seguinte situação: como um vértice com dimensões não nulas, uma aresta também poderia ser apontada por outra aresta. Essas duas possibilidades, sobre a geometria da aresta e sobre uma aresta ser apontada por outra, levanta uma questão: Se uma aresta 'A' aponta para uma aresta 'B', para que ponto da aresta 'B' a aresta 'A' está apontando? A questão foi respondida quando se determinou a posição do ponto de conexão de 'A' em 'B', no espaço definido pela geometria da aresta 'B'. Para que este posicionamento fosse possível, partiu-se para pesquisas de definição de pontos dentro da trajetória linear por partes, de uma aresta, a partir de seu comprimento de arco.

As arestas podem ser diferenciadas por parâmetros geométricos, como: trajetória ou visualização (cor, tracejado ou espessura). Vale lembrar que as arestas podem ter extremidades com geometrias determinadas, que podem ser determinadas por esses parâmetros. Uma extremidade é uma geometria que ocupa um espaço, que está definido

entre o ponto apontado pela aresta, e o último (ou primeiro) ponto que a compõe. Considerados os dois pontos de referência, a política de posicionamento dos pontos das extremidades funciona de maneira idêntica ao posicionamento dos pontos de uma aresta.

Sobre as envoltórias, considerando seu desenho circular ao redor dos elementos pontuais contidos, pode-se escolher um ponto de origem qualquer, dentro do círculo que a representa, e considerar a envoltória como uma sequência de segmentos de reta que inicia neste ponto, percorre a circunferência, até chegar ao mesmo ponto. Dessa forma, é possível fazer uma aresta apontar para a envoltória, determinando o seu ponto de conexão a partir do ponto de origem escolhido. Nesse caso de conexão entre envoltória e aresta, há um inconveniente: considerando que exista uma envoltória 'E' com posição livre, uma aresta 'A', que parte de uma origem fixa 'O', e aponta para a envoltória 'E' em um ponto escolhido de sua circunferência 'p'. Dado que a posição de 'O' é fixa, haverá posições de 'E' em que a aresta 'A' cruza a envoltória até chegar a 'p', o que acrescenta um cruzamento de linhas na visão dos elementos. Diante disso, a estratégia de conectar uma aresta 'A' a uma envoltória 'E', através de um ponto em sua circunferência, torna-se um inconveniente na visão desses dados. Melhor, então, seria programar o sistema de visão de envoltórias e arestas (das ontologias), para que, dada uma aresta 'A' com ponto de origem fixa 'O', apontando para uma envoltória 'E' com centro 'c', o sistema identificasse a linha reta entre 'O' e 'c', formando uma reta de cruzamento diametral à circunferência de 'E', calculasse os dois pontos de cruzamento entre a reta de 'A' e a circunferência de 'E', e assumisse como ponto final da aresta, o ponto que estivesse mais próximo de 'O'. Sendo assim, a aresta 'A' e a envoltória 'E' nunca se cruzariam, tornando mais limpa a visão dos dados.

Sobre a possibilidade da multiplicidade de elementos, tanto no editor 2D quanto no editor 3D, teve-se a preocupação de prever possíveis necessidades de visão de dados, decorrentes do crescimento do número de elementos das instâncias de ontologias. Nos casos dos dois editores, previu-se haver mais de uma relação semântica entre os dois termos, dependendo do domínio a que os termos (ou a que as relações entre eles) pertencessem. Se caso existir, será possível ao engenheiro de ontologias representá-lo visualmente.

Para ser possível que entre dois vértices existam várias arestas, todas com formato e sentido diferentes (não direcionadas, orientadas da

origem ao destino, do destino à origem, e bidirecionais) e independente, foi necessário incluir um contêiner de arestas (uma classe chamada de SuperAresta), que por sua vez estivesse conectada aos dois vértices que estariam apontados por todas essas arestas.

A situação de multiplicidade de elementos também ocorre com as envoltórias, na qual uma instância de uma envoltória pode conter várias instâncias de envoltória, ou estar contida dentro de outras instâncias de envoltórias.

Para o caso das envoltórias contendo outras envoltórias, foi planejada a implementação da envoltória contendo uma lista de instâncias também de envoltórias. Considerando a programação dos elementos com orientação a objetos, diferente do caso das arestas, em que um contêiner de uma classe SuperAresta contém instâncias de classes Aresta, não herdadas da classe SuperAresta, o caso da classe Envoltória representa uma recursividade: ela contém uma lista de instâncias da própria classe Envoltória. Tal situação significa que, diferente da situação das arestas, pode ocorrer casos em que as envoltórias se contenham de maneira cíclica: uma envoltória 'E0' (que na ontologia representa um domínio 'D0') pode conter outra envoltória 'E1' (que na ontologia representa um domínio 'D1'), que por sua vez pode conter novamente 'E0' ('D0'). Casos de ciclo de continência também podem ocorrer em vários níveis. Exemplo: em uma envoltória inicial 'E0', ela pode conter 'E1', que contém 'E2', que contém 'E3', que novamente contém 'E0' (uma continência cíclica de quarto nível). Sabe-se que essa situação pode ocorrer, o que causaria um ciclo infinito no processamento; um artifício para bloquear ou identificar a situação não foi planejado ainda.

Sobre as combinações de conexão entre elementos, partiu-se da separação dos componentes de uma rede de relacionamentos (e envoltórias). Baseado nisso, foi construída a tabela de possibilidades de conexões entre os elementos. O processo de separação dos componentes por ordem de complexidade (análise) e de sua reunificação em várias combinações (síntese) ajudou a prever situações naturais.

Um fenômeno interessante observado foi que, para algumas combinações de elementos, o número de possibilidades de combinações é maior do que um. Por exemplo, ao tomar a linha L1 e a coluna C3, da Tabela 3 (página 103), que representa a combinação da visão de uma aresta com o espaço vazio, pode-se representar uma aresta discretizada ou uma aresta contínua (uma curva de Bézier ou uma curva gerada por

uma função paramétrica), e não somente uma aresta reta. Desse modo, têm-se possibilidades diferentes de combinações de uma envoltória com o espaço vazio, ou seja, várias envoltórias conectadas entre si por suas bordas, ou ainda, uma envoltória isolada com dois pontos de sua borda conectados.

Outro fenômeno interessante encontrado foi que, como os quatro elementos básicos estão dispostos tanto na linha superior inicial quanto na coluna esquerda inicial da tabela, a matriz de combinações torna-se simétrica em relação à sua diagonal descendente, ou seja, a combinação resultante, a princípio, de  $L_i \times C_j$  é equivalente a  $L_j \times C_i$ . Por simplificação, poder-se-ia ignorar as células abaixo dessa diagonal, para que as células de informação não se repetissem, mas, ao se aproveitar as células que por princípio seriam abstraídas, têm-se mais espaço para incluir mais combinações dos elementos básicos  $L_i \times C_j$  ou  $L_j \times C_i$ .

A combinação de conexões entre elementos ajudou na criação de novas possibilidades, e na pré-seleção de situações que foram programadas no sistema de visualização de dados de ontologias. O método de análise/síntese ajudou a fazer previsões de possíveis situações que serão usadas no futuro, para a representação de instâncias de ontologias, na forma de redes de relacionamento, e ajudou a fazer previsões para a representação das redes de informação com objetos tridimensionais. Algumas combinações foram implementadas no sistema de visão de ontologias, como a possibilidade de conectar vértices, arestas e envoltórias, a outros vértices, arestas e envoltórias, através de arestas, e ainda a possibilidade de incluir vértices, arestas e envoltórias, dentro de várias envoltórias simultaneamente, e também incluir envoltórias dentro de envoltórias em vários níveis.

Apesar de essas possibilidades estarem implementadas no código fonte do editor 3D de ontologias, estando a futura framework preparada para representar os tipos de relacionamento no espaço tridimensional, nota-se a falta de disponibilização das formas de informação de maneira persistida, seja em sistemas de arquivo em uma base de dados, com as tabelas e restrições adequadas. Mas, mesmo que essas possibilidades não estejam disponíveis no formato de persistência, a implementação delas na interface 3D mostrou que sua aplicação é possível, tanto para os casos de informações das instâncias de ontologias que precisam ser mostradas visualmente (conexão de múltiplas arestas entre vértices, múltiplos níveis de envoltórias e múltiplas envoltórias contendo um



elemento) quanto para os casos que ainda não ocorreram (conexão entre quaisquer tipos de elemento, com o uso de múltiplas arestas).

Para a visão de relações de ocorrência conjunta vistas como arestas conectando um vértice a uma envoltória com vários vértices, a estratégia inicial foi deixar a visão de uma rede de relacionamentos limpa, reduzindo o número de cruzamento entre arestas. No modelo de Bueno (2005), na rede de relacionamentos das ontologias produzidas, há casos em que um mesmo vértice termo está conectado a vários outros vértices termo (como um hub), por relações que indicam ocorrência conjunta (relação de dois termos conexos) de termos dentro dos documentos semanticamente analisados. Se retiradas as relações entre o termo hub e os termos conectados, e colocados os termos conectados dentro de uma envoltória, e conectar o termo hub à envoltória por uma aresta, visualmente se manterá o significado de que todos os termos pertencentes à envoltória são conexos ao termo hub, apontado pela aresta. Dessa forma, para representar os termos conexos a um termo hub, trocam-se 'N' arestas, por somente uma aresta, mais uma envoltória ao redor dos termos conexos ao termo hub, reduzindo 'N' vezes o número de arestas, e deixando a rede de relacionamentos da ontologia mais clara para o engenheiro de ontologias. Este recurso ainda não foi implementado para o caso de relações de conexão (ocorrência conjunta em textos) entre termos, mas já há suporte de programação pronto para ser usado, no futuro, para a implementação desta forma de representação no espaço 3D.

Além do ganho de clareza na visão dos relacionamentos desse tipo, há ganhos de performance da máquina que os apresenta visualmente, já que, ao invés de recalcular 'N' arestas, a máquina precisará recalcular apenas uma envoltória (que para o modelo 2D possui oito segmentos de reta, e para o modelo 3D possui seis superfícies).

Sobre o ganho de performance nas máquinas, de acordo com o tipo de visualização do conjunto de relacionamentos, Nagy et. al. (2010) consideraram o fato de que, até o ano de 2007, as máquinas desktop comuns não possuíam processamento suficiente para garantir uma visualização contínua e em tempo real para visualização de dados em um espaço 3D. Atualmente, as máquinas já possuem tal processamento, e o resultado visual em tempo real pode ser auxiliado pelas próprias formas de apresentação de informação, como é o caso da substituição das formas de representação de elementos conexos; como explicado

mais à frente, é o caso da substituição das formas de representação de sinônimos.

Apesar do ganho de clareza para a forma proposta de representação de relação semântica entre um termo hub e vários termos, essa nova forma não deve substituir por completo a forma antiga, pois ela serve como um elemento de destaque de vértices que tenham um diferencial dos demais vértices conectados ao hub, que estejam dentro da envoltória.

Na opção de visão das relações de sinonímia como envoltórias, seguindo o mesmo raciocínio do uso de envoltórias como substituição das arestas, para o caso de arestas representando relações de sinônimos entre vértices, tem-se um ganho ainda maior.

De início, com a representação visual de dois termos sinônimos, conectados por uma aresta não-direcionada e caso se substitua a aresta por uma envoltória, então não se tem ganho algum. Caso haja três termos, ainda estaremos perdendo processamento.

Agora, considerando um conjunto de vinte ou trinta termos, todos sinônimos entre si, tem-se um número muito maior de arestas. Considerando que, dados  $N$  termos conectados entre si na forma de uma clique, tem-se  $(N \times (N - 1))/2$  arestas conectando todos os  $N$  vértices dois a dois. Para o caso de vinte vértices:  $(20 \times 19) / 2 = 190$  arestas. Para trinta vértices sinônimos entre si:  $(30 \times 29) / 2 = 435$  arestas, e assim por diante. Para o caso de grandes números de vértices, foi mais conveniente substituir todas as relações por uma envoltória, ao redor dos vértices sinônimos entre si. A envoltória possui a vantagem de possuir menos elementos a ver (e no caso da máquina, a processar), do que as  $N \times (N - 1) / 2$  arestas a mostrar, para o caso destes  $N$  vértices sinônimos.

Mesmo assim, como opção, deve-se deixar a forma de representação das relações de sinonímia como arestas, pelos mesmos motivos de particularização que foram dados aos vértices conectados por relações de ocorrência conjunta em documentos (vértices conexos).

Sobre a persistência de relações de sinonímia e de ocorrência conjunta, apesar da possibilidade de representação de relações de ocorrência conjunta de termos e de relações de sinonímia entre termos, com o uso de envoltórias ao redor de termos agrupados, na interface visual do editor 3D, na base de dados correspondente, a disposição dos dados referentes as relações é seqüencial: os termos são representados dentro de uma tabela de termos, em que cada termo corresponde a um

registro da tabela, com o seu respectivo índice numérico. Por sua vez, as relações são armazenadas em uma tabela de relações, em que cada relação corresponde a um registro com o seu número de identificação, o termo de origem da relação e o termo de destino desta relação.

Partindo desse modelo de domínio, foram verificados dois problemas de consulta de relações, a partir de um termo, na base de dados: 1) Se o usuário quiser todos os termos conexos a um termo hub conhecido; 2) Se quiser todos os termos sinônimos de um termo de referência conhecido (e também sinônimos entre si).

Caso se conheça o termo hub e se deseja seus termos conexos, basta requisitar uma *query* de relações de ocorrência conjunta, nas quais os termos de origem ou de destino sejam o termo hub. Para o caso de se conhecer um termo de referência e se deseja seus termos sinônimos, têm-se dois princípios: a) Uma vez que um usuário cria uma relação de sinonímia de um termo A com termo B, e sabendo que B já é sinônimo de C, D, ..., então o sistema infere que A também é sinônimo de C, D, ..., e, além de conectar o termo A com os demais termos sinônimos de B, formando a clique, ele preenche esta informação também na base de dados, criando os registros correspondentes às relações de sinônimo entre A e C, A e D, e assim por diante. B) Quando o usuário cria uma relação de sinonímia de um termo A com termo B, e sabendo que B já é sinônimo de C, D, ..., então o sistema infere que A também é sinônimo de C, D, ..., e apenas conecta o termo A com os demais termos sinônimos de B, formando a clique na interface visual, sem preencher as relações correspondentes na base de dados.

Nos dois casos, para que um segundo usuário recupere todos os termos sinônimos a partir de um termo inicial, o processo necessita de um encadeamento de pesquisas na base de dados: inicialmente, pede-se uma *query* de todas as relações de sinonímia que tenham o termo de referência como origem ou como destino. Posteriormente, tem-se como resultado o termo de referência como um termo hub, conectado com os demais termos resultantes como sinônimos. Entretanto, após essa pesquisa, ainda não se tem a conexão de sinonímia entre os termos entre si, exceto com o termo de referência, que é o termo hub.

Para que seja possível capturar as relações de sinônimos de todos os termos ligados ao *hub*, executa-se, para cada um dos termos iniciais ligados ao hub, uma nova *query*, procurando por todas as relações de sinonímia que possuem os termos de origem ou destino, capturando novamente alguns dos termos pertencentes ao conjunto da primeira

*query*, mais alguns novos termos, conectados aos termos resultantes da primeira *query*. Como resultado visual, são conectados os termos da primeira pesquisa com os termos da segunda pesquisa; para completar o grupo de sinônimos, executa-se novamente a *query* sobre todos os termos atualmente conectados, até que os termos resultantes das *queries* não sejam mais diferentes dos termos resultantes.

O processo descrito acima é trabalhoso para a máquina, o que, a princípio, pode-se concluir que a maneira computacional mais barata de armazenar as relações de sinônimos, quando criadas na interface de construção de ontologias, seja armazenar somente a relação de sinônimo criada pelo usuário entre dois termos atuais, sem guardar as relações de sinônimo inferidas pela máquina.

Quanto ao desenho das extremidades de arestas, o processo de cálculo dos pontos é idêntico ao cálculo dos pontos da aresta entre vértices, sendo uma lista de coordenadas no espaço (tanto para o caso 2D quanto para o caso 3D). Para o caso das coordenadas 2D, o eixo principal dos pontos da extremidade é o eixo X, horizontal. Para o caso das coordenadas 3D, o eixo principal dos pontos da extremidade é o eixo Z, de profundidade. O uso desses eixos garante que a seta sempre siga a direção e sentido definido pelos pontos extremos da aresta.

Apesar de o modelo contemplar os modelos de extremidades das relações do Diagrama de Classes da Linguagem Unificada de Modelagem, há uma liberdade maior de construção dos formatos dessas extremidades, já que os parâmetros geométricos são passíveis de customização. Com essa liberdade, um outro problema aparece: como cada usuário pode determinar a forma geométrica das suas formas de representação dos elementos das instâncias das ontologias, pode ocorrer uma perda da comunicação sobre a forma que representa cada relação semântica. Assim, preferiu-se utilizar um conjunto simbólico previamente conhecido e largamente utilizado.

Um dos motivos pelos quais foi feito um esforço para parametrizar o formato geométrico das extremidades foi o fato de que as relações semânticas entre os termos de uma ontologia são em grande quantidade, e por isso muitas delas podem assemelhar-se entre si, necessitando de formas de diferenciação visual. De um princípio, como a quantidade de relações semânticas trabalhadas na prática, a partir do modelo de Bueno (2005), são poucas, julgou-se a quantidade de símbolos do Diagrama de Classes da UML suficiente. Mas o modelo

matemático já está preparado para suportar mais representações visuais de extremidades associadas a novas relações semânticas.

Para os modelos de navegação e manuseio de objetos no espaço 2D e 3D, um sistema de navegação para um ambiente é indispensável, já que são poucos os casos em que é possível ver todos os elementos de informação de maneira imediata. É necessário, então, que o sistema de visualização e manuseio ofereça recursos que permita ao usuário caminhar entre os dados das ontologias, em separado, ou entre blocos de dados, e que também permita capturar e movimentar dados separados ou seus blocos.

Nessas condições, a experiência de implementação da navegação para os espaços 2D e 3D foi diferente. Para o caso da translação, enquanto que para o espaço 2D havia somente a opção de alterar a posição absoluta dos elementos (vértices, arestas e envoltórias), para o espaço 3D, há duas opções: ao mover o espaço com um dispositivo de navegação (normalmente o *mouse*, para um posicionamento contínuo, extenso e imediato, e mais raramente o teclado, para um posicionamento discretizado, curto e dedicado), ou altera-se a posição absoluta dos elementos das ontologias, ou a posição da câmera do observador.

Quanto ao caso da rotação, no espaço 2D, este foi implementado somente na primeira versão do editor de ontologias, e no sistema de gestão de informações de investigação Nexus (STRADIOTTO *et. al.*, 2004), quando os dois sistemas foram construídos com base na mesma *framework*. A rotação no espaço 2D depende de um centro geométrico de rotação (que para a *framework* era o centro da janela que continha a rede dos elementos da ontologia), e seu ângulo de giro é uma função linear do deslocamento do dispositivo de navegação (o *mouse*), no eixo vertical ou horizontal. No espaço 3D, para a rotação, têm-se duas alternativas: ou são girados todos os elementos ao redor de um centro, ou a câmera. A solução usada foi o giro da câmera ao redor de um ponto no espaço (originalmente no centro de coordenadas geométricas), e foi implementada a partir de um exemplo de código Java encontrado em uma página Web repositório de códigos de exemplo.

Para o caso de aproximação e afastamento (Zoom), no espaço 2D, nos dois editores 2D iniciais, foi implementada uma modificação apenas da posição relativa dos elementos entre si: os elementos apenas ficavam mais próximos, ou mais afastados, mas não alteravam, proporcionalmente, seu tamanho. Esse fenômeno fazia com que, caso o usuário aproximasse um dos elementos, os demais elementos ficassem

afastados deste, de maneira correta. Mas caso o usuário afastasse os elementos, eles ficavam cada vez mais agrupados, e quando o usuário voltasse a aproximá-los, muitos permaneciam posicionados no mesmo ponto, se sobrepondo.

Ainda para o caso de aplicação de Zoom no espaço 2D, é interessante utilizar, ao invés da posição espacial com coordenadas inteiras, coordenadas de ponto flutuante, para que sempre se mantenha uma diferença na posição entre dois elementos geométricos, mesmo para valores altos de afastamento.

O processo de Zoom utilizado nos dois primeiros editores de ontologias é diferente do processo utilizado por Heer *et. al.* (2007), na biblioteca Prefuse de visão de dados, em que o Zoom foi aplicado a todos os elementos que compõem uma rede de relacionamentos. Desse modo, quando se aproximou uma RR, verificou-se que, além da aproximação relativa dos elementos, houve o aumento do tamanho dos elementos representados, e vice-versa.

Para o caso do editor no espaço 3D, foi garantido na programação que o efeito de Zoom só pode ser aplicado de uma forma, com a movimentação da câmera. O processo de Zoom por transformação linear de escala nos seus elementos pontuais não foi aplicado. Um dos motivos para a não aplicação, foi o desconhecimento do efeito desta aplicação de escala nos elementos que já sofreram outro tipo de transformação linear, como uma rotação ou translação. Neste caso, sabe-se que a ordem em que são aplicados os transformadores altera o resultado da visão dos elementos.

Para o manuseio de elementos, foi usado o *mouse*. Para o caso dos editores 2D, baseados na mesma biblioteca (Borland C++ Builder), o princípio utilizado foi a captura da posição relativa do *mouse* sobre o objeto, e o deslocamento deste objeto com a movimentação do *mouse*, sempre deslocado da posição relativa inicial. A API utilizada possuía recursos para que cada objeto ‘sentisse’ a posição relativa do mouse.

Nas duas primeiras versões foi possível fazer a seleção de grupos de vértices e a movimentação deste grupo, baseando-se no princípio de que, movendo um vértice, move-se todo o grupo. Os vértices selecionados são acumulados em uma lista. Caso o usuário clique sobre um dos vértices e o movimento, o sistema identifica que este vértice está na lista, e movimenta de um mesmo deslocamento todos os demais

vértices da lista. Na versão 3D, a seleção de grupos ainda não foi implementada.

Na segunda versão do editor de ontologias, considerando que o usuário poderia ver um domínio como envoltória, procurou-se copiar o modelo de agregação de dados mostrado em Heer *et. al.* (2007), em que, ao se clicar em um ponto que pertencesse a uma envoltória, e o cursor do *mouse* fosse movimentado, todos os vértices, arestas e envoltórias que pertencessem a esta envoltória também seriam movimentados no mesmo deslocamento.

Quanto ao editor 3D, o processo de seleção de elementos parte do princípio de que o sistema obtém, do usuário, uma região retangular da tela, correspondente à região de seleção. Para o sistema (A API Java3D), o retângulo possui profundidade infinita. Como resposta, o sistema devolve para o usuário uma lista de todos os objetos que possuem pontos dentro dessa região, em sua ordem de profundidade: Os elementos mais próximos da câmera estão ordenados em primeiro lugar.

Apesar de a resposta da API garantir que o elemento mais próximo sempre seja selecionado, há casos em que o primeiro elemento acessado pelo *mouse* pode não ser o elemento de interesse do usuário. É o caso de envoltórias e vértices: caso o usuário clique em um ponto onde coincidem uma envoltória e um vértice atrás dela, a prioridade será dada para o vértice, já que, caso o usuário desejasse focar somente na envoltória, teria de clicar sobre ela em um ponto onde não coincidissem nenhum outro elemento. O processo é semelhante para uma envoltória A contendo outra envoltória B: caso o usuário clique sobre um ponto que esteja sobre A e B, sendo que A contém B, então a envoltória com foco será B. O usuário pode perceber qual envoltória está dentro de outra pelo fato de que a extensão das faces de A é ligeiramente maior que a extensão das faces de B.

Observando o menu de tarefas do editor Web de ontologias da Suíte de Engenharia do Conhecimento, nota-se que os submenus estão desconectados dos termos, relações e domínios objetos de sua ação. O menu principal está localizado à esquerda da interface principal, enquanto a interface de entrada de parâmetros do elemento trabalho está à direita, ocupando o maior espaço.

Foi notado – e também tal fato foi confirmado pelos usuários do sistema – que este desenho dos menus prejudica a produtividade no processo de construção de ontologias, já que ao acessar um dos menus, o

usuário não vê a ontologia em seu atual estado de trabalho, seja como um todo, seja apenas pela aproximação de um grupo de dados, através de Zoom.

Outra desvantagem dessa interface é que, ao iniciar o processo descrito em um dos sub-menus (como alterar um domínio, ou incluir um termo), na interface à direita, o usuário deve ainda procurar o termo ou domínio desejado, já que o usuário não pode ver todos os termos, domínios e relações ao mesmo tempo, de forma imediata, para que escolha facilmente o elemento sobre o qual deseja trabalhar.

Considerando essas limitações, procurou-se abordar os elementos particularizados de uma forma em que o usuário pudesse ver a instância de ontologias que estivesse editando, de maneira imediata, e que pudesse escolher o elemento que desejaria alterar, apenas usando o cursor do *mouse*, sem a necessidade de etapas adicionais no processo de edição, tal como acontece com o editor Web de ontologias (esta limitação também foi criticada por um dos entrevistados sobre a comparação dos editores).

A solução encontrada foi a implementação de menus contextualizados, disparados pelo clique de um botão do *mouse*, em que o usuário escolhe o elemento de trabalho, clicando sobre ele, e dependendo da classe do elemento, o menu apresenta as possíveis atividades a serem executadas.

Caso de o usuário clique em um espaço vazio, há opções para criar um novo vértice, ou um novo domínio (tanto no editor 2D quanto no 3D). Caso o usuário clique em um vértice (uma palavra universal ou um termo), o sistema oferece opções de menu para alterar seu conteúdo, eliminá-lo, criar uma relação (ou conectar uma relação em processo de criação), ou incluí-lo ou removê-lo de um domínio. Caso o usuário clique sobre um domínio, o sistema oferece opções para eliminá-lo, ou para eliminá-lo junto com seu conteúdo, ou alterar seus parâmetros de identificação, ou para incluí-lo ou removê-lo de dentro de um domínio pai. No caso do editor 3D, ainda não há uma opção contextualizada de menu para os objetos 3D dentro do ambiente virtual do editor. Tais funcionalidades devem ser incluídas no futuro.

Durante os trabalhos experimentais com os editores, vários testes de viabilidade foram executados com o objetivo de criar protótipos para testes de viabilidade de representação gráfica das ontologias em ambiente 2D e 3D. A finalidade das experimentações era verificar as



funcionalidades julgadas necessárias para a visão dos elementos das instâncias de ontologias, nas suas várias formas, desde as mais simples (visão de termos e palavras universais como vértices), até as mais complexas (inserir múltiplas arestas parametrizadas entre vértices, incluir termos, arestas e envoltórias dentro de envoltórias, e sincronizar os elementos geométricos e as relações entre eles com a base de dados).

A etapa seguinte foi a análise do impacto nos elementos vistos em grande quantidade. Para os casos de vértices e arestas, a apresentação de elementos em quantidades moderadas já exigia um processamento computacional da máquina, causando demoras nos resultados visuais. Mesmo assim, deve-se considerar que a capacidade das máquinas vem melhorando consideravelmente, o que resolverá o problema de processamento visual de dados em grandes quantidades. Outro ponto positivo a respeito do tempo de processamento de atividades é o crescimento do uso de tecnologias compartilhadas para o processamento de dados (de bases de conhecimento ou de visão de informações), a insurgente tecnologia de *Cloud Computing*, em que uma tarefa é dividida em várias sub-tarefas, e sub-tarefas são distribuídas para várias máquinas.

A construção de protótipos também serviu para isolar uma funcionalidade, sem a necessidade de passar por todo um processo de preparo de espaço até que a situação na qual seria aplicada a funcionalidade ocorresse, havendo, assim, ganho de tempo. Comparada com o processo de implementação de uma nova funcionalidade diretamente dentro do aplicativo, a prototipagem foi a maneira mais fácil de conseguir entender como a nova solução funcionava, e quais os pontos da nova solução deveriam ser alterados para que ela fosse encaixada dentro do sistema final.

O teste comparativo de usabilidade com os três modelos de editores de ontologias (tabular, 2D e 3D) foi efetivado na forma de entrevistas, mostradas no capítulo 5. Foram observados elementos comuns nas respostas dos entrevistados:

1) Quanto ao editor Web, seu modelo é o mais antigo dos três, conseqüentemente, com recursos visuais mais limitados. Os dados de ontologias são apresentados de duas formas: listas indentadas (domínios e subdomínios) e tabelas (relações semânticas e termos).

2) Visão de tabelas: os usuários entrevistados responderam que tal interface é muito comum. Para o caso de dados em formato tabular,

foi difícil ver a totalidade dos dados de interesse (uma tabela quase sempre tem suas dimensões maiores que a tela; se os caracteres forem reduzidos de tal forma que a tabela se reduza à área da tela, perde-se a acuidade visual para compreender o que está escrito ou desenhado). O critério de Clareza teve por parte dos entrevistados um valor baixo. Apesar disso, para casos em que a tabela não precisa ser vista completamente, mas somente a ordem de seus elementos, a ordem foi facilmente percebida (valor alto no critério de clareza).

Para o critério de Trabalho Final Percebido, os entrevistados chegaram a um acordo sobre a vantagem da visão tabular, por possuir ordem e separabilidade (ou classificação) dos dados apresentados das ontologias de trabalho. Sendo assim, foi fácil identificar quanto falta para o término da construção de uma ontologia. Tal ordem e classificação são melhoradas ainda mais com o uso de cores de fundo (ou cores de caracteres) para as células da tabela.

A representação tabular também facilita o Foco Direcionado, mesmo se a tabela exceder as dimensões da tela. Todos os entrevistados responderam que, devido à ordem apresentada na visão tabular, a procura por elementos foi facilitada, por ser sequencial, não havendo a necessidade de mudanças de atividade durante a construção de uma ontologia com o uso destas tabelas.

3) Visão em árvores: para o caso de árvores expansíveis, também chamada de Lista Indentada, o problema está em não ser possível ver níveis inferiores de nós. Caso a árvore esteja com alguns nós contraídos, e ela caiba dentro da região visível da tela, não se sabe o conteúdo dos nós inferiores que estão dentro dos nós contraídos. Ao contrário, caso a árvore esteja toda expandida, sua extensão pode ser tão grande que esteja fora da região visível da tela (novamente um valor baixo no critério de clareza), exigindo inclusive uma navegação extensa para que o usuário possa ver seus elementos.

Quanto ao Foco Direcionado, a visão em árvores atua como um detalhador de informações, já que, quando o engenheiro de ontologias abre os nós mais externos, referentes aos domínios, fica cada vez mais se aprofundando na descoberta dos nós mais internos (subdomínios ou termos).

4) Sobre o grupo de elementos e sobre os procedimentos do editor: no critério de Facilitação de Idéias, o editor Web de Ontologias obteve críticas pelo fato de ser estático, não sendo possível a

alteração/atualização de elementos (de tabelas ou árvores) na medida em que as necessidades se apresentavam. Sendo um documento estático, caso se pense em outras alternativas que não estejam representadas como tabelas, árvores, ou caixas de texto, o editor não permitiria a concretização das idéias de representação, frustrando o usuário. Essa limitação caracterizou os conceitos de VISCOSIDADE (o usuário tem dificuldade em fazer pequenas mudanças na estrutura de informação) e BAIXA POSSIBILIDADE DE IMPROVISACÃO (o usuário pode expressar poucas partes da estrutura de informação, ainda não definidas na ferramenta), conforme previu Blackwell *et. al.* (2001).

No critério de modificabilidade, o editor Web recebeu críticas quando à quantidade desnecessária de etapas que possui para alterar cada um de seus elementos, e também sobre a indisponibilidade de menus sensíveis ao contexto (perda no critério de Foco), em que o usuário poderia acessar opções específicas somente do elemento acessado. Como exemplo, para alterar um termo de uma instância de ontologia, neste editor, são necessárias nove etapas: 1) Pedir a alteração do termo; 2) Selecionar o domínio do termo; 3) Selecionar o termo de uma lista; 4) Inserir a nova escrita do termo; 5) Pedir a alteração via botão; 6) A máquina pede um aviso de confirmação; 7) O usuário confirma compressão em botão; 8) A máquina mostra termos semelhantes; 9) Aviso de fim da operação de alteração.

Dentro de um menu sensível ao contexto, basta apenas: 1) Uma requisição de alteração através de botão de mouse; 2) Uma seleção de operação de alteração; 3) Uma apresentação de uma caixa de escrita do novo termo; 4) Um botão de confirmação da nova escrita do termo.

Quanto ao critério de Suporte à Interação de Grupo, o editor Web apresentou a desvantagem de mostrar que dado elemento já existe na base de dados da ontologia quando o usuário já fez uma série de operações para a inserção deste elemento. Para os casos dos editores de ontologias 2D e 3D, através de artifícios de auto-complemento, e identificação de conceitos e domínios relacionados, é possível ao usuário identificar que um elemento desejado já existe na base, antes de continuar as operações de inserção.

Observando as funcionalidades de suporte à interação em grupo, os três editores de ontologias oferecem quase todas as funcionalidades, citadas por Noy *et. al.* (2008), como pertencentes à Web Semântica: uma semântica, uso de ontologias (a ferramenta Web é um editor de ontologias) com seu respectivo esquema de base de dados (que é o

modelo de domínio), e a taxonomia com definições explícitas (as instâncias de ontologias construídas possuem relações de classes/subclasse/instância). As três ferramentas ofereceram apenas uma funcionalidade pertencente à Web 2.0: o compartilhamento dos objetos construídos, alterados ou destruídos, das instâncias de ontologias.

5) Quanto aos editores 2D e 3D, a apresentação dos termos como nós de uma rede de relacionamentos, de suas relações semânticas como arestas, e de seus domínios como envoltórias, permitiu a apresentação da rede completa dentro da tela. Para poucos nós, poucas arestas e poucas envoltórias – em casos em que há poucos cruzamentos de arestas e envoltórias e poucas sobreposições entre os nós – a rede tem seus elementos bem destacados, e a visão das continências e dos relacionamentos facilitou a dedução de informações (alta clareza), mais do que a visão de elementos de ontologias dentro de tabelas ou árvores.

Uma vantagem adicional à visão de ontologias como redes de relacionamento foi a possibilidade de mudar os elementos gráficos que representam os termos e relações, como: cor, espessura, tracejado e trajetória de uma aresta e envoltórias, e também cor, espessura, tracejado e trajetória das extremidades de arestas. Nagy *et. al.* (2010) argumentam que o controle de variáveis visuais para facilitar a procura, conexão e compreensão dos mapeamentos entre termos, a escala da rede e a amostragem parcial da rede são uma ajuda considerável na visão dos elementos de uma ontologia e na descoberta de componentes ocultos, que é fornecida quando se permite que o usuário veja classes de elementos de seu interesse com propriedades particularizadas, tais como seus componentes geométricos.

Além da determinação dos elementos gráficos de arestas e vértices, existe a vantagem de mostrar elementos anexados às arestas (ver Figura 51, página 133), vértices e envoltórias da rede de relacionamentos referente à ontologia trabalhada, aumentando ainda mais a liberdade na representação de informações sobre a ontologia.

Para casos em que a rede de relacionamentos possua muitos elementos sobrepostos, a apresentação da rede ao usuário mostra muitos pontos de cruzamento de arestas e sobreposição de vértices, tornando a interface confusa, sendo necessário que o usuário disponha de artifícios de reorganização da rede, ou de filtragem de dados, para deixar na tela os dados relevantes. Os entrevistados reclamaram da dificuldade de procura de um dado termo ou palavra universal dentro de uma rede muito populosa (Baixo Foco Direcionado). Essa dificuldade é

comentada por Katifori *et. al.* (2007) que sugere que as visões de ontologias devem ser auxiliadas por artifícios de *query* e filtros, e também por Kobsa (2004) que a dificuldade da visão 3D é dada pela falta de ferramentas de procura, de destaque visual, filtragem e navegação.

Uma vantagem – quanto à organização dos dados – do editor 3D sobre o editor 2D e tabular, é a possibilidade de dispor um grupo de vértices em planos em várias camadas, ou em várias superfícies (cilíndrica, cônica, ou esférica) para identificar grupos rapidamente. Equivalentemente, no editor 2D, tais formas poderiam ser a retangular, triangular ou circular.

Os engenheiros entrevistados comentaram sobre a vantagem de apresentar os domínios em vários níveis (editores 2D e 3D), aumentando em muito a capacidade de representar as informações na profundidade necessária de assuntos. Quanto ao Trabalho Final Percebido, todos notaram a necessidade da ferramenta como apoio na visão das instâncias de ontologias construídas, já que a visão em redes permite ver a instância de uma ontologia como um todo, dentro do mesmo espaço.

No quesito de Facilitação de Idéias, no editor 2D, os entrevistados comentaram sobre a interface desorganizada, em que não havia uma ordem perceptível pelo usuário no agrupamento dos termos e palavras universais. O problema da desorganização dos dados pode ser resolvido com a programação de módulos que permitam ao usuário trabalhar rapidamente com a seleção de grupos de elementos, e o posicionamento desses grupos, ou ainda com módulos que façam a reorganização dos elementos de maneira autônoma, como é descrito nos trabalhos de Heer *et. al.* (2007), com a *framework* Prefuse, de visualização de dados, construída na linguagem Java, e também como é descrito por Stradiotto *et. al.* (2004, 2006), com a *framework* RR, para a construção de aplicativos de apresentação de dados na forma de redes de relacionamento.

Quanto ao quesito de Modificabilidade, os entrevistados comentaram positivamente sobre a velocidade com que os elementos da instância de ontologia construída podiam ser alterados, criados ou eliminados, tanto no editor 2D quanto no 3D. Para tal, estavam disponíveis aos usuários os menus sensíveis ao contexto, ajudando também na manutenção do foco de trabalho (mais um quesito avaliado positivamente).

6) Sobre o editor 3D, quanto ao critério de impacto visual, os entrevistados comentaram sobre a interface do editor de ontologias em três dimensões parecerem uma arte final. Um dos prováveis motivos para essa percepção é o fato de que o conjunto de objetos, relações semânticas e domínios, mais os objetos 3D mostrados na tela fornecerem uma impressão muito convincente de que o resultado pertence à realidade que está fora da tela. Outra vantagem levantada foi a capacidade de o usuário perceber a dimensão real do domínio, tanto no aspecto espacial quanto de conteúdo: Em várias situações é possível perceber relações de pertinência devido à visão de muitos objetos 3D envolvidos por uma construção, por exemplo, ou é possível ver relações de classe ou instancia conectando um modelo mais simplificado a vários outros modelos derivados, mas mais complexos.

Um fenômeno que ajuda na percepção dos elementos da ontologia como reais foi a projeção da luz sobre eles. A variação do ângulo de uma superfície provoca mudanças nas tonalidades das cores, formando *dégradés* que se assemelham aos elementos da realidade. Um segundo fenômeno que talvez tenha ajudado na percepção das redes de ontologias como arte final foi a multiplicidade de retângulos que representam os termos e palavras universais de uma ontologia, parecendo um conjunto de seres vivos que pertencem a uma mesma classe, e que andam em bando, como acontece na vida real. O terceiro fenômeno foi a visão dos elementos conectados com relações semânticas, como árvores: aplicados os algoritmos matemáticos corretos, a disposição de vértices e objetos conectados no espaço 3D pode resultar em formas de extrema beleza. Tais observações são confirmadas com a afirmação de Bosca *et. al.* (2005), que sugere que a apresentação 3D oferece maior riqueza de informações.

Quanto ao critério de clareza do editor 3D, os entrevistados comentaram sobre a desvantagem de vários elementos permanecerem posicionados no mesmo ponto, projetado no espaço 2D da tela, necessitando de várias operações de arraste de vértices para obter uma disposição mais clara. Neste caso, o espaço 3D também apresenta problemas de sobreposição de elementos (a ordem Z, citada em aplicativos gráficos). O caso de alinhamento de elementos na profundidade aconteceu quando os elementos ficam posicionados ao longo do eixo ótico do observador. Essa dificuldade confirma as afirmações de Plaisant *et. al.* (2002), de que a visão 3D aumenta a complexidade da interação, e também de Cockburn e McKenzie (2002),

que mostraram que a navegação em um espaço 3D pode ser difícil para um usuário novato.

Uma solução rápida para a descoberta de que mais de um elemento existe atrás do objeto mostrado em uma dada posição foi um leve giro do ambiente. Os elementos que estão atrás do objeto, por terem seu alinhamento saindo do eixo ótico do observador, se revelarão. Uma segunda solução para o problema de sobreposição de elementos foi utilizar recursos de transparência, de tal forma que, caso haja um elemento atrás de outro, em um ponto de interesse do usuário, o objeto de trás será revelado. Esse artifício foi utilizado na apresentação das envoltórias, pois se sabe que, em um espaço 3D, uma envoltória que envolve um objeto por todos os lados, tem alguma parte de sua superfície constituinte sobrepondo os objetos que circunda.

Quanto ao critério do Trabalho Final Percebido, houve um consenso somente entre dois entrevistados no que se refere à visão da conexão entre termos e palavras universais. Para um entrevistado, não foi possível perceber um trabalho final. Outro entrevistado respondeu sobre a falta de informações sobre as envoltórias, faltando rótulos (foi percebido o objetivo final do trabalho e uma carência até chegar no objetivo final). Um quarto entrevistado comentou sobre o alto nível de completude da forma de representação das ontologias sendo o espaço 3D o mais próximo do desejado. A percepção do quarto entrevistado confirma a hipótese de Katifori *et. al.* (2007), de que a visão humana baseia-se em projeções tridimensionais do mundo real, assim as visões mais próximas da visão 3D seriam as mais efetivas.

Sobre esse ponto, observou-se uma confusão na compreensão do critério de Trabalho Final Percebido: o Trabalho Final Percebido refere-se à construção de uma instância de uma ontologia até o seu final, enquanto está sendo mostrada no editor experimental, ou à construção do editor 3D de ontologias multilíngües e de suas funcionalidades até o seu final? No caso do entrevistado que respondeu sobre a falta de rótulos, esta é uma funcionalidade do editor, que não está completo. No caso dos entrevistados que comentaram sobre a percepção da conexão entre termos, e sobre a sua pertinência a um dado domínio, mesmo sem conexões, esta é uma característica da ontologia que é mostrada. Em futuros testes, esta questão deve ser dividida em duas partes: 1) Trabalho Final Percebido sobre o Editor, em que se percebe o quanto o editor possui funcionalidades para facilitar a construção de ontologias; 2) Trabalho Final Percebido da Instância de Ontologia Construída, em que

se percebe o quanto uma ontologia possui seus elementos conectados de maneira coerente, domínios construídos e objetos relacionados aos termos corretos, nos vários idiomas.

Sobre o critério de Foco Direcionado, os entrevistados fizeram comentários de dificuldades de procura de um elemento pontual, como um termo ou palavra universal. Praticamente, a dificuldade de procura de um elemento, tanto no espaço 2D quanto no espaço 3D, foi ocasionada pelo fato de que, em uma escala na qual se vêem todas as entidades na tela, estas são tão reduzidas que não é possível ver as informações que as diferenciam. Em compensação, em uma escala grande a ponto de se enxergar os detalhes que diferenciam os elementos, não é possível ver os detalhes dos elementos que estão fora da tela; conseqüentemente, não é possível encontrar, de maneira imediata, as informações dos elementos externos para se ter pistas durante a procura do elemento de interesse.

Sobre o quesito de Facilitação de Idéias, alguns entrevistados comentaram sobre a possibilidade de girar a rede como um todo, conseguindo descobrir elementos sobrepostos. Neste ponto, seria possível que os entrevistados compreendessem que facilitar uma idéia seria o mesmo que “ter uma idéia sobre algo”, ou melhor, perceber como algo está estruturado, ou como funciona? Um dos entrevistados comentou sobre a segurança de que, uma vez existente uma vontade, um potencial, há uma garantia de que este potencial será satisfeito (uma percepção de capacidade e realização: do que é possível fazer). Este mesmo entrevistado comentou sobre o fato de o editor 3D trazer uma idéia da conexão, uma idéia do mapa (novamente o comentário sobre uma percepção, sobre “ter a idéia” de como algo está estruturado), e também sobre a possibilidade de expandir os elementos que já existem (novamente uma percepção de capacidade: do que é possível fazer).

Um dos entrevistados comentou sobre a utilidade das cores para facilitar a edição. Pode-se considerar que a facilidade de edição dos elementos mostrados pelo editor acelerou o cumprimento das idéias do usuário, pois aquilo que o usuário imaginou, para fazer parte de sua ontologia construída, tornou-se rapidamente a realidade dentro do espaço. Esta característica de tornar realidade o que o usuário imaginou pode ser comparada à propriedade de plasticidade, em que as estruturas gráficas plásticas podem ser adaptadas a necessidades locais (ver a propriedade Plástico-Robusto, de Star e Griesemer (1989), no Quadro 10, página 93).



Um dos entrevistados comentou sobre a incompletude do protótipo de editor 3D, e que, devido a isso, se sentia livre para sugerir novas idéias de funcionalidades para a ferramenta. Aqui se pode observar que o critério de Facilitação de Idéias foi um adicional ao critério de Trabalho Final Percebido: Quanto mais o trabalho pareceu terminado, mais idéias podiam ser aplicadas nele.

Um dos entrevistados fez comparações das entidades presentes no espaço 3D de trabalho do editor com elementos da realidade: quanto aos retângulos que representavam termos e palavras universais, vistos de longe, parecia uma revoada de pássaros. Se esses termos e palavras universais estivessem conectados, pareciam florestas ou postes de transmissão de energia elétrica (o entrevistado estava, naquele momento, trabalhando em um sistema de gestão de processos em uma empresa de energia elétrica). Quanto à disposição espiral dos vértices, este usuário fez metáforas com ciclones, com a cadeia de DNA (o entrevistado é bacharel em Biologia), e com luzes de discoteca.

Com esta seqüência de comparações, verificou-se que a combinação de elementos de informação dentro de um espaço tridimensional, combinando as várias formas de movimentação disponíveis, e também com as variantes de luz e cores, pode levar o usuário a criar metáforas de elementos da realidade, o que ajuda no processo de *brainstorming* para a geração de ideias para soluções em outras áreas. Essa geração de ideias está conivente com a afirmação de Bueno (2005), que se utiliza da etapa de visualização das ontologias, durante seu processo de construção, para a criação de ideias na geração de novas soluções.

Quanto ao critério de Modificabilidade, ficou claro, durante o uso do editor 3D, em sua fase experimental, que as alterações na rede de ontologias puderam ser feitas sem a necessidade de várias etapas: bastava o acesso a um elemento, o disparo de um menu flutuante contextualizado e a escolha da operação necessária neste menu para executar a alteração do elemento escolhido. Esta facilidade também foi identificada no editor de ontologias em 2D. Outros pontos comentados por um dos entrevistados, quanto à modificabilidade, foi a ajuda da terceira dimensão para classificar grupos de dados: elementos da ontologia que possuíam propriedades interessantes para o usuário poderiam ser colocados em primeiro plano, mais próximos do usuário. Elementos que possuíam propriedades menos relevantes eram

posicionados mais distantes do usuário. Os dados eram alterados dentro da própria interface, o que aumenta a modificabilidade das ontologias.

Algumas respostas, para o item de Modificabilidade, foram colocadas como Não Implementado. Este resultado se deve ao fato de que, na época das entrevistas comparativas, havia somente artifícios para incluir vértices (termos, palavras universais) e relações, e artifícios para alterar propriedades de vértices (termos e palavras universais).

Quanto ao critério de Suporte à Interação de Grupos, durante as entrevistas, o editor não possuía tal funcionalidade, sendo implementada parcialmente após as entrevistas. Mesmo assim, na tabela final, para os editores 2D e 3D, é comentada a possibilidade de implementação de algumas funcionalidades de auto-complemento (para garantir que uma operação de inclusão não seja desperdiçada) e identificação de responsáveis por determinadas operações, através da própria interface gráfica. A última versão permitia a visão da aplicação dentro de uma página Web, em que testes de visão foram feitos com navegadores Web de três marcas diferentes. Para que a visão da aplicação Web fosse possível, foi necessária a abertura manual de permissões nas máquinas que acessavam a página Web na qual a aplicação Java estava disponível. Um último experimento de atualização da inclusão de um termo na ontologia em duas máquinas, de maneira simultânea, foi feito com sucesso, mas sem continuação.

Como compensação para a falta de compartilhamento, um artifício possível de ser usado para que um usuário saiba quais os novos dados foram inseridos por outros usuários, foi a importação de dados com uma determinada frequência, pelo usuário interessado. Esta solução (incompleta) também foi sugerida para a atualização de dados no editor de ontologias em 2D. Essa solução é apenas um paliativo, já que ela não é em tempo real, e não resolve a atualização de processos de eliminação de elementos da base de dados, vistas pelas interfaces de vários participantes do processo compartilhado de construção de ontologias.

Quanto ao problema de falta de consenso sobre os conceitos dos elementos de uma ontologia, e sobre o nível máximo de encadeamento de domínios do conhecimento, este trabalho de construção de uma ontologia foi realizado entre os anos de 2005 e 2006, entre 2008 e 2010. Nesse período foram observados problemas na conceituação dos elementos das ontologias, entre os engenheiros do conhecimento.

Quanto à relação semântica de meronímia (parte), para facilitar a comunicação entre os engenheiros responsáveis pela construção das ontologias, tais relações eram descritas como A é parte de B, se B correspondia a um todo que continha A como uma de suas partes. De forma inversa, a relação B é parte disso de A, significa que B é o todo de A, ou que B contém A. A descrição deste segundo tipo de relação possui falha nas duas interpretações: a) Supondo que a descrição das duas relações semânticas seja de inversão (A é parte de B é inversa de A é parte disso de B), semanticamente, não há diferença entre a descrição A é parte de B, e a descrição A é parte disso de B, já que, com exceção do pronome demonstrativo isso, que significa que B já é previamente conhecido em um parágrafo, parte de e parte de + isso são equivalentes, quanto à informação de que o primeiro é a parte e o segundo é o todo. Portanto, as duas descrições descrevem a mesma coisa, mas na prática são usadas para descrever situações inversas. b) Supondo que a descrição das duas relações seja de equivalência (A é parte de B é equivalente a A é parte disso de B), o uso de uma das duas descrições não se justifica, já que, computacionalmente, para descrever a mesma relação de meronímia basta uma delas, e a outra descrição da relação pode ser descartada. Um motivo para o descarte da relação A é parte disso de B é a sua grafia incoerente, já que, se A é parte disso, contendo o pronome demonstrativo indicando que isso já é previamente conhecido, então não é necessário informar também de quem A é parte, que na descrição A é parte disso de B, seria B.

Como sugestão de solução para o problema de descrição da relação de meronímia usada, se A é parte, e B é o todo que contém a parte A, como uma de suas partes, é preferível usar as expressões A está contido em B, e B contém A, nomenclatura utilizada, inclusive, na descrição de relações entre conjuntos, em disciplinas básicas de matemática, com sua representação gráfica correspondente: Uma envoltória circunscrevendo outra.

O problema de descrição dessa relação ocorreu durante o processo de comunicação entre os profissionais responsáveis pela construção de instâncias de ontologias, em que, por uma necessidade de respostas rápidas à demanda de um cliente, decidiu-se pelo uso corriqueiro das duas descrições, como inversas, da relação de meronímia.

Quanto à relação semântica de hiperonímia (classe), a descrição é bem usada: se A é classe de B, então significa que B possui as mesmas

propriedades de A, mais as suas próprias, e sendo abstrações as duas. O que não se pode definir aqui, com esta descrição somente, é se B pode ser uma instância de uma classe A, ou se B pode ser também uma classe, mais específica, que contém as mesmas propriedades de A, mais as suas próprias, sendo assim uma subclasse de A. Portanto, salvo os casos em que não seja necessário, é preferível sempre usar a descrição do mais específico para o mais genérico: B é subclasse de A, indicando que A e B tem as mesmas propriedades, e B contém as suas propriedades adicionais, e as duas são abstrações, ou B é instância de A, indicando que B é uma realização de A, sendo A abstrata e B concreta. Para os dois casos, vale a afirmativa de que A é classe de B.

Foram observados alguns resultados quanto à sintaxe das palavras universais produzidas. Novamente, devido ao problema de atendimento a uma rápida demanda, notou-se que não houve o cuidado adequado na construção de instância de palavras universais, durante a criação das instâncias das ontologias multilíngües. Uma palavra universal tem sua sintaxe básica definida por duas partes: uma palavra, da língua inglesa, mais freqüentemente usada, que serve como raiz do conceito a representar (chamada de *headword*), e contendo os elementos de domínio que restringem esta palavra raiz, indicando os domínios do conhecimento a qual esta palavra pertence, e também os papéis usados para esta palavra, dentro de uma sentença (as restrições). Foi percebido que muitas das palavras universais produzidas eram traduções diretas de termos para a língua inglesa, utilizadas como equivalentes aos termos dentro de determinados domínios, mas sem as correspondentes restrições de uma palavra universal, que seriam equivalentes aos domínios e subdomínios da instância de ontologia em seu idioma original.

Sobre o nível máximo de encadeamento de domínios do conhecimento, foi observado que uma das limitações mais restritivas, no antigo modelo do editor de ontologias, era a representação dos domínios do conhecimento em apenas dois níveis: um termo qualquer era guardado em um subdomínio, e este subdomínio era guardado em um domínio. Havia possibilidades de que um domínio contivesse vários subdomínios, e que um subdomínio contivesse vários termos, mas não havia qualquer possibilidade de que um subdomínio estivesse contido em mais de um domínio, ou que um termo estivesse contido em mais de um subdomínio, simultaneamente, caracterizando o resultado do conhecimento como um rizoma, ao invés de uma árvore.

Essas limitações ocorreram porque o modelo de domínio continha uma tabela de domínios, em que cada registro de domínio apontava para o seu domínio pai, somente, permitindo uma multiplicidade de múltiplos domínios filho para somente um domínio pai. Esse modelo, apesar de não permitir múltiplos domínios pais para um subdomínio, ou para um termo, permitiu que um domínio contivesse múltiplos termos ou múltiplos domínios filhos. Quanto ao número máximo de níveis de encadeamento de domínios, o modelo permitiu um número infinito, mas, na prática, foram observados no máximo três níveis. O motivo do número pequeno de níveis talvez seja porque o número pequeno dos níveis evitasse a ocorrência de contradições em seus encadeamentos, facilitando a gestão dos domínios por um grande número de pessoas.

Posteriormente, com a mudança de modelo de domínio, duas tabelas referentes a domínios foram criadas: uma tabela com os nomes dos domínios pertencentes à instância de ontologia, e outra tabela indicando pares de pertinência de domínios. Como exemplo: tem-se uma ontologia com quatro domínios (A, B, C, D), para indicar que A pertence a B e C ao mesmo tempo, e também B e C pertencem a D, preenche-se a segunda tabela com as tuplas ((A, B), (A, C), (B, D), (C, D)), em que as tuplas (A, B) e (A, C) indicam que o domínio A é subdomínio de B e C simultaneamente, e as tuplas (B, D) e (C, D) indicam que os domínios do conhecimento B e C pertencem ao domínio D. Para contar a capacidade de níveis de encadeamento entre domínios, a nova solução é idêntica à anterior. A mudança se dá no fato de que este novo modelo permite representar um domínio ou termo como contido em mais de um domínio simultaneamente, o que pode acontecer em problemas reais de representação de conhecimento, caracterizando a representação em rizoma, e não mais em árvore.

Embora o editor ofereça a possibilidade de inclusão de modelos 3D, associados a conceitos, durante a construção do protótipo e durante os testes de importação e navegação sobre os modelos, foi notada uma limitação: quando se quer a disponibilidade de um modelo 3D, na própria Web, ele não está tão disponível em documentos, quanto à informação em texto, ou quanto à informação em formato de imagem ou vídeo.

Por conveniência de programação, o único formato no qual o editor estava apto a reconhecer foi o da *W3C Virtual Reality Modeling Language* (VRML), já que existem bibliotecas completas o suficiente para a carga de entidades geométricas e texturas, que são suficientes

para a carga de modelos estáticos bem representativos. Devido a esta limitação, houve uma dificuldade considerável em encontrar exemplos na Web que pudessem ser usados para a associação de conceitos a modelos prontos. Durante a procura, foram identificadas as seguintes limitações na captura de modelos tridimensionais para a representação e comunicação de conceitos:

1) Apresentado um conceito que precisa ser representado por um modelo 3D, a disponibilidade do modelo na Web é ainda pequena: mesmo para conceitos mais simples, que não estão dentro de várias camadas de domínios e subdomínios, tem-se dificuldade de encontrar modelos 3D para representar este conceito, com uma fidelidade relativamente baixa. São mais fáceis de encontrar modelos de entidades que estejam no conjunto de objetos de uso comum, e que tenham geometria simples, como: modelos de roupas de profissões, calçados, veículos, móveis, alguns tipos de construções e armas (relativas aos jogos mais famosos). Foi observado que os objetos produzidos são muito ligados ao estilo, e não tem a intenção de repassar o conceito, ou repassar informações sobre seus componentes para fins de comunicação. Sabe-se que modelos 3D, por terem uma produção difícil, mesmo com ferramentas e profissionais especializados, são criados como contribuições de livre arbítrio de seus criadores, portanto, se sua produção não é estimulada com alguma forma de troca, seu número é baixo.

2) Os modelos são disponibilizados em poucos formatos: apesar da existência de mais de cinquenta formatos de gráficos modelados no espaço 3D, formatos mais comuns ainda são dos aplicativos mais populares, como 3D Studio, AutoCAD ou VRML, ou modelos para jogos de computadores da Microsoft Direct X e Counter Strike. Dessa forma, é interessante programar aplicativos que possam ler arquivos nesses formatos. Com a evolução da linguagem Java, e seu reuso por vários grupos de pesquisa ao redor do mundo, é grande a chance de encontrar uma biblioteca que faça a leitura e escrita desses formatos, pronta a ser reusada. Uma segunda alternativa que está surgindo recentemente são serviços web de conversão de formato de arquivos: o usuário fornece o endereço do arquivo que deseja converter, e o serviço retorna um endereço de onde o usuário possa fazer o *download* do modelo convertido no formato desejado, para a sua máquina. Uma terceira alternativa de conversão de formatos de um modelo 3D é o uso de um aplicativo especializado para modelagem. Alguns aplicativos são

de livre distribuição, podendo ser baixado de sua página web correspondente, instalado, e utilizado para a conversão.

3) Muitas páginas Web com modelos 3D são pagas: são muito comuns páginas web que mostram uma foto demonstrativa em pequena escala (*thumbnails*) de seus modelos 3D, mas quando o link das imagens é acessado, ela leva o usuário a uma página de sugestão de ingresso do usuário como cliente do serviço de disponibilização de modelos 3D, fornecendo campos de entrada de nome e número de cartão de crédito. Neste processo, depende do usuário final do editor de ontologias comprar o modelo pronto, quando o julgar adequado para representar um conceito multilíngüe dentro da ontologia construída. Muitos modelos simples são vendidos a preços acessíveis, podendo ser um caminho fácil para a colaboração do engenheiro do conhecimento na construção de uma instância de ontologia.

Uma segunda solução para a falta de modelos foi a criação de modelos por conta própria, o que serviu para perceber as seguintes dificuldades:

1) Modelos tridimensionais são demorados de construir, em qualquer editor: quanto maior o nível de detalhe, mais tempo leva para que um modelo esteja pronto para ser considerado representativo de um conceito. Um modelo com baixíssima qualidade de geometria (com poucas transformações, partindo de uma forma básica como um cilindro, paralelepípedo, cone ou esfera) e baixíssima qualidade de textura (os detalhes são desenhados a mão, com somente uma ou duas aplicações de processamento de *pixels*, como *blurring*, distorção por refração em vidros irregulares, ou aplicação de granularização) leva aproximadamente duas horas para ser produzido. E isso considerando que o modelo pode ser produzido para atender somente à representação de um único conceito. Para a construção de um modelo, deve-se também considerar o nível de perícia que um profissional tem sobre uma ferramenta, demandando certo tempo para o seu aprendizado, caso precisar.

2) Nem sempre o designer consegue modelar o que precisa: para a construção de um modelo tridimensional não há um padrão. O resultado desejado depende muito da aplicação dos deformadores corretos sobre o objeto inteiro, ou sobre partes dele, isso sem contar sobre como deve estar o objeto representado quando se aplica um deformador: se somente por pontos, superfícies planas ou blocos de superfícies planas. Em grande parte dos casos, o resultado é diferente do

esperado, devendo o profissional guardar o resultado parcial em uma biblioteca de modelos, procurando reusá-lo no futuro, caso necessite de um modelo que seja semelhante ao resultado parcial conseguido. Dessa forma, o designer deve partir de modelos já prontos, aproximados do modelo 3D desejado para representar o conceito. O resultado depende muito da repetição das tentativas, até chegar a um resultado satisfatório.

3) Um modelo 3D deve ser planejado: os modelos 3D que representam conceitos podem também ajudar a representar as relações (de parte, instância ou classe). Deve-se fazer um planejamento de que partes este modelo conterá (o modelo de um prédio deve contar também os móveis dos apartamentos?), quais as etapas necessárias para a construção do modelo 3D (caso as etapas sejam conhecidas por um designer experiente), e, depois de pronto, como ele será ‘desmontado’ para que leigos, que vejam a instância de ontologia com conceitos relacionados àquele modelo, possam ver também como ele se relaciona com suas partes, instância e classes.

Por exemplo: uma flor, em sua conceituação básica, possui uma raiz, um caule, com folhas ao redor, e a coroa com suas pétalas, nesta seqüência. O designer deve planejar o modelo da raiz, do caule, das folhas, e o modelo do centro da coroa, das pétalas, de forma que, durante a construção, ele possa partir de um elemento básico, como um cilindro, e aplicar o deformador de extrusão na sua extremidade inferior, para iniciar o molde da raiz, posteriormente aplicar a extrusão na sua extremidade superior, e para o caso das pétalas e folhas, iniciar com uma superfície plana, recortada em duas curvas. Sendo assim, ele poderia montar o modelo 3D completo da flor, e deixar disponível os modelos de suas partes, para que engenheiros do conhecimento, ou para que os próprios usuários leigos possam associar a espécie representada da flor, modelada completamente, com suas partes, já desmontadas e disponibilizadas pelo designer que a construiu, tal como em uma visão explodida de um projeto. Os tipos de relacionamento, do exemplo da flor completa com as suas partes, são de meronímia. Tal exemplo pode ser extremamente óbvio para efeitos didáticos nesta tese, mas o método pode ser expandido para casos em que a visão das partes de um elemento não são tão óbvias visualmente, como a representação das partes de elementos microscópicos, primeiro disponibilizados em sua versão completa, ou para casos de representação de fenômenos físicos e químicos que são conhecidos através de metáforas, como os fenômenos de transmissão de calor, com uso de cores, ou como as ondas eletromagnéticas, com o uso de linhas ou vetores de campo.



Por último, foi observada a tendência de compartilhamento de modelos 3D graças à iniciativa da empresa Google, que disponibiliza publicamente seu modelador 3D Sketchup, que, do momento de redação desta tese, está em sua versão 8. A empresa disponibiliza um portal com vários modelos 3D no formato COLLADA, um formato texto XML, que é alimentado pelos próprios usuários do aplicativo. Sendo assim, se tem acesso a uma quantidade enorme de modelos 3D, e a oportunidade de associar tais modelos – ou de suas partes – a conceitos das futuras ontologias construídas pelo editor 3D de ontologias. Vale lembrar que o aplicativo Google Sketchup permite a rotulagem de objetos 3D, rotulagem de suas partes, ou a rotulagem de grupos de objetos.

Na proposta de construção de ontologias multilíngües e de um método de busca multilíngüe sobre o qual foram aplicadas as ontologias construídas, de início, poder-se-ia questionar a utilidade deste tipo de buscador: do que adiantaria o retorno de documentos em um idioma diferente do idioma da busca? Esta pergunta pode ser respondida de três formas:

1) Este buscador seria útil para procurar documentos que são úteis em seu idioma original, como é o caso das leis de um determinado país, dentro de uma negociação diplomática. Obviamente, este buscador deve ser usado por pessoas que conheçam tanto o idioma do documento de busca quanto o idioma dos documentos retornados.

2) Este tipo de buscador serve como ferramenta de ensino de outro idioma, em que um usuário interessado entraria com os termos no seu idioma, e a ferramenta lhe indicará os termos dos documentos retornados, equivalentes aos termos de entrada.

3) Finalmente, um buscador multilíngüe serviria como um caminho intermediário na construção de um buscador de linguagem cruzada (*Cross-lingual searcher*), já que a quantidade de variáveis trabalhadas em um buscador que retorna documentos em um idioma diferente do idioma de procura teria o espaço de variáveis reduzido, pois não haveria a preocupação, a priori, com todo o processo de conversão entre o idioma do documento de entrada e do documento de retorno. Assim, uma vez que o processo de construção de ontologias estiver com seus métodos bem estabelecidos e o processo de busca multilíngüe com sua qualidade de resposta razoável, parte-se para a construção de buscadores de linguagem cruzada, em uma etapa seguinte.

Apesar de o buscador funcionar com termos equivalentes em idiomas diferentes, e com sinônimos no mesmo idioma, ainda não se sabe como será seu comportamento quanto a termos equivalentes, mas que tem diferentes relações semânticas. Casos frequentes são expressões idiomáticas, como, em português: *rápido*, *rápido!!*, em que seu equivalente em inglês é *Hurry Up!!* (Rápido para cima).

Sobre o uso da tecnologia UNL para a redução no número de dicionários de equivalência entre idiomas: considerou-se que um buscador multilíngüe pode ser construído de duas formas: 1) Com a tradução direta de um idioma 'Ik' para outro 'Il' (e vice-versa), através de um dicionário de equivalência 'Dkl'. Assim, caso se tenha N idiomas a serem traduzidos entre si, têm-se  $N \times (N - 1)$  dicionários a compilar; 2) Com a tradução de um idioma 'Ik' para outro 'Il' (e vice-versa), através de um dicionário normalizado da UNL U. Assim, têm-se N dicionários de conversão, do idioma Ik, para o dicionário normalizado U.

Quanto ao processo proposto de construção de ontologias multilíngües, este método é derivado do método proposto em Bueno (2005), em que, na execução de um projeto envolvendo diferentes instituições, e conseqüentemente diferentes culturas organizacionais, se considera necessária uma etapa de uniformização de vocabulário, para que todos os envolvidos no projeto entendam com um mesmo significado o mesmo conjunto de termos e relações semânticas.

Para que tal aconteça, faz-se necessário o levantamento de um dicionário de termos coletados do vocabulário cotidiano de dentro das empresas envolvidas, a partir de seus documentos, ou a partir de documentos desejados pelos participantes de cada uma das instituições (*corpus* de documentos).

O levantamento de termos e relações semânticas pode ser feito por algumas ferramentas automáticas para este fim (*Wordsmith*, citado nesta tese), ou através do uso de bibliotecas, como as bibliotecas de expressões regulares (*RegEx*) do Java.

A etapa de uniformização de conceitos é feita com a conversão dos termos coletados para o formato das UW's da UNL. Tal processo de conversão utilizará uma combinação de dicionários on-line, WordNet, do dicionário de palavras universais, e de ontologias para o processo de desambiguação.

Após a construção dos termos, será feita a determinação dos domínios e subdomínios, em vários níveis, para facilitar a especialização do processo de procura de documentos, reduzindo, com este artifício, o problema de ambigüidade dos termos produzidos nos vários idiomas.

Posteriormente será feito o processo de construção de relações semânticas, com a sua dedução a partir da comparação da redação atual de uma sentença com redações de sentenças anteriores.

Ainda não se sabe qual o tamanho do problema para resolver a equivalência, entre termos de dois idiomas diferentes, mas que tem diferentes relações semânticas. Para resolver este problema, talvez se recorra à solução de guardar listas de expressões idiomáticas, com suas respectivas expressões UNL equivalentes, para manter a solução de N dicionários de conversão Idioma/UW para N idiomas.

O processo de construção de ontologias também considerou a associação de conceitos e relações a objetos digitais semelhantes à realidade. Acredita-se que, com esta equivalência direta, resolvem-se muitos dos problemas de interpretação de conceitos referentes a objetos físicos, ou locais, reduzindo barreiras linguísticas. É muito comum encontrar ontologias em formato texto, organizadas em domínios e subdomínios, mas tais ontologias seriam muito mais compreensíveis se cada um de seus termos fosse associado a um objeto tridimensional, e os termos e suas relações semânticas seriam ainda mais compreensíveis se for possível associar os objetos do espaço virtual entre si, como relações de parte, equivalência ou classe, o que é muito comum na visão explodida de projetos de peças mecânicas. O mesmo raciocínio vale para a associação de conceitos de locais, e seus respectivos objetos localizados, como um rio, tipos de plantas, animais, ou construções, em um mapa 3D. Este fenômeno já está ocorrendo em aplicativos como o Google Earth e Bing Maps.

Apesar da facilidade de comunicação oferecida com a equivalência de termos a objetos 2D ou 3D, deve-se considerar o custo que os objetos 2D ou 3D exigem: a memória, os diferentes formatos na sua construção, e o tempo demandado na construção. Modelos tridimensionais limpos e estáticos ocupam uma quantidade considerável de memória. Mais memória é demandada ao se incluir texturas e diferentes luzes nos modelos, e mais memória ainda é ocupada ao se dar movimento aos modelos. Finalmente, toma-se um tempo e dedicação exclusiva de um profissional para a produção de tais modelos, o que exige artifícios para a criação de bibliotecas de modelos por sua

similaridade geométrica, ou pela similaridade no uso de transformadores matemáticos para produzi-los.

Um último ponto a considerar, é o fato de que cada EC terá de testar, iterativamente, a eficácia da equivalência entre termos, domínios e relações entre diferentes idiomas, através do uso da ferramenta de busca, como uma forma de diagnóstico. Sabe-se que todas as etapas da construção e testes de ontologias são consideravelmente enfadonhas, o que exige do futuro sistema de construção um conjunto de ferramentas e técnicas de visualização e procura que evidenciem ao máximo os itens procurados pelos EC's, durante as construções. Para estes casos, serão necessárias novas pesquisas referentes à interação Homem-Máquina, considerando a possibilidade de aumentar os canais de comunicação entre o usuário e o sistema, como reconhecimento de palavras faladas, síntese digital de palavras, dispositivos para interface de Realidade Virtual, como luvas digitais, HMD, e a possibilidade de agentes inteligentes, com uso de algoritmos de IA, inclusive aproveitando o conhecimento dedutivo armazenado nas ontologias construídas.

## 8 CONSIDERAÇÕES FINAIS

Os modelos de domínio propostos para este editor são estruturas rígidas, pois derivam de um modelo de base de dados: o modelo, uma vez definido, não pode mais ser adaptado a novas necessidades, ou sua adaptação, em tempo de uso, é muito arriscada ou cara. Isso requer uma previsão de cenários completa e estruturada. Sugere-se criar um modelo de domínio baseado em estruturas flexíveis de armazenagem e consulta, como XQuery (Vitria M3O).

Ainda não se sabe o quanto a representação tridimensional de objetos auxilia na compreensão de elementos de ontologias, pois faltam testes de uso da ferramenta para a construção de instâncias reais de ontologias, mas se conhece que a disposição de um grupo de objetos, ou de locais, no espaço 3D, conectados entre si, por relações de classe, parte/pertinência ou sinonímia, pode informar mais a respeito da instância de ontologia, do que simplesmente conceitos escritos em um idioma diferente do idioma de um Engenheiro de Conhecimento.

O processo de construção de instâncias de ontologias, auxiliadas com objetos 3D, é um processo caro, já que a construção de um modelo demanda tempo e profissionais especializados. A vantagem oferecida é a facilidade de comunicação entre pessoas de diferentes idiomas, e a classificação de objetos de design com os conceitos relacionados: um uso invertido da ferramenta. Uma segunda alternativa para a construção de objetos 3D seria o processo de scanning, mas devido ao fato de ferramentas de scanning 3D serem tecnologias de ponta, seu custo é alto, e são ferramentas ainda de uso complexo, que exigem treinamento especializado, aumentando o custo ainda mais.

Com a repetida construção de ferramentas de visão de informação na forma de relacionamentos e conjuntos, combinadas com a inclusão de objetos no espaço 3D, já se possui uma expertise na construção deste tipo de ferramenta, que pode ser repassada a outros grupos de trabalho.

A construção do editor ainda é uma etapa inicial, em que, a partir de seu uso na construção de instâncias de ontologias, serão criadas novas ferramentas anexadas para atender as possíveis necessidades de comunicação, tanto do sistema principal com sistemas remotos quanto dos Engenheiros do Conhecimento entre si. Embora o editor esteja funcionando parcialmente, ele já mostra que é possível associar

elementos geométricos com rótulos, e é possível disponibilizar a ferramenta na Web.

A ferramenta possui uma estrutura computacional preparada para representar várias formas de conexões entre elementos (Envoltória/Envoltória, Envoltória/Aresta, Envoltória/Vértice, Aresta/Vértice, Aresta/Aresta, Aresta/Envoltória), junto com a representação de objetos 3D, permitindo uma vasta expressividade de domínios do conhecimento, relacionamentos lógicos, em múltiplos idiomas, com seus equivalentes em ambiente tridimensional.

Diante disso, concluiu-se o seguinte:

a) Sobre a Programação:

Após os vários protótipos experimentais usados na construção do aplicativo, foi possível programar os seguintes componentes visuais:

A representação visual de conjuntos através de envoltórias;

A representação visual de múltiplas arestas entre dois vértices;

A representação visual de informações anexadas em vértices e arestas.

A versão apresentada do aplicativo de edição de redes de ontologias possui classes que podem ser reusadas e expandidas para a criação de outros aplicativos.

Embora se tenha observado que a framework utilizada ofereça liberdade para construção de aplicativos, verificou-se que a performance de aplicativos de visão de redes de relacionamento e envoltórias (a partir do modelo de Graham (1972)) da toolkit Prefuse, descrita por Heer et. al. (2005), é muito superior.

b) Sobre a Usabilidade:

Observou-se que a quantidade de critérios de usabilidade é grande. Alguns critérios têm uma descrição complexa (Ex.: os critérios de modificabilidade, trabalho final percebido e facilitação de idéias). Esta dificuldade reside no fato dos critérios possuírem conceitos similares que se sobrepõem, ou se confundem, sendo necessário um

planejamento da aplicação desses critérios, à medida que a comparação dos editores for feita. Entretanto, existe a possibilidade que um mesmo critério seja desmontado em vários subcritérios, sendo assim possível criar uma métrica de comparação de usabilidade entre sistemas.

Nesse sentido, recomenda-se que, após a pesquisa dos critérios, um núcleo de pessoas com habilidades para implantação de métodos avalie a usabilidade no sistema de construção de ontologias.

c) Facilidades esperadas para o método de produção de ontologias:

Intui-se que este modelo de editor de ontologias 3D pode oferecer as seguintes facilidades:

Aplicação digital da metodologia de representação do conhecimento com ontologias, descrita por Hoeschl et. al. (2007a) e Bueno (2008a).

Permite uma visão mais clara da rede de ontologias criada, na qual o usuário pode identificar quase que imediatamente as relações corretas, em excesso, faltantes, ou relações trocadas

Os usuários do método/sistema podem comunicar os resultados das ontologias criadas, entre si, de forma digital/compartilhada em redes, acelerando a verificação das ontologias.

## 9 SUGESTÕES PARA TRABALHOS FUTUROS

A partir do comparativo de funcionalidades de outras ferramentas de construção de ontologias, ou de rotulagem de informações, observou-se que o projeto da ferramenta oferecida no decorrer da tese ainda tem muitas funcionalidades a serem projetadas, principalmente nos módulos de comunicação entre sistemas, comunicação entre usuários, compartilhamento de instâncias de ontologias, inferências, reconhecimento de formatos, rotulagem e interface com o usuário. Devido ao fato de a ferramenta ter sido construída em Java, que é uma linguagem extensível, e de disponibilidade pública na Web, pensa-se na adoção de uma política de construção de extensões independentes, os chamados plug-ins, política esta já usada em muitos aplicativos, tais como Protege e ImageJ (este último, construído em Java, mas para tratamento de imagens).

Apesar de o editor construído, em sua forma protótipo, oferecer recursos para associar termos multilíngües, objetos e palavras universais da UNL, ainda não foi estudado artifícios para que, após a construção de uma instância de ontologia, a máquina infira informações. Apenas a possibilidade de escrita do texto, em um idioma qualquer, ou na sintaxe de palavras universais, foi oferecida. Também há a necessidade de representar os símbolos lógicos, comumente usados em instâncias de ontologias OWL, e já presentes em ferramentas que associam OWL com UML, com os elementos disponíveis na estrutura computacional do editor. Um rápido exemplo a respeito desta representação gráfica de símbolos lógicos seria representar o operador lógico de implicância como uma aresta direcionada, conectando duas envoltórias, e dentro das envoltórias, as premissas do precedente, e os elementos do conseqüente. A operação lógica “e” seria representada como uma envoltória ao redor das premissas necessárias, apontando para o conseqüente, e o operador “ou” seria representado pelos seus elementos fora de envoltórias, mas todos apontando para o conseqüente.

O editor construído pode representar objetos concretos no espaço 3D, mas ainda não se sabe como representar elementos abstratos, como leis, regras e procedimentos. É muito provável que, com o tempo, haja modelos gráficos de representação de tais elementos, já que, pelo crescimento de informações na Web, e pelo alto custo do processo de comunicação entre os diferentes elementos abstratos, será feita uma procura por soluções de padronização. Por enquanto, há a solução de



criar formas de representação de conceitos abstratos por profissionais especializados no ramo (designers), mas ainda assim, há a necessidade de acordos sobre a forma gráfica de representação de um conceito abstrato, o que é bem diferente de representar um conceito concreto a partir de sua comparação com um objeto observado na realidade.

No futuro, deseja-se permitir a comunicação do sistema com os vários dicionários e serviços de tradução existentes na Web. Muitos serviços são grátis, pois estão a título experimental. Serviços que disponibilizam páginas para acesso por pessoas recebem parâmetros pela barra de endereços do navegador Web, e são passíveis de processos de “desmonte” (*parsing*) de páginas web, para a comunicação com o editor de ontologias.

A associação de objetos 3D a conceitos também pode ser útil para sistemas inteligentes de reconhecimento de imagens, já que, quando uma ontologia com vários objetos 3D, com seus conceitos correspondentes, estiver disponibilizada, um sistema poderá fazer o processo de reconhecimento de um objeto a partir das imagens geradas por vários ângulos de visão deste objeto, e da associação das imagens ao conceito do objeto. Vale lembrar que, se o objeto 3D já existir em um ambiente virtual, é mais fácil gerar imagens dos vários ângulos de visão do objeto virtual, do que produzir as visões de objetos reais. Tais informações de visão de objetos 3D e associação a conceitos poderão servir para tecnologias intensivas em conhecimento armazenado neste espaço, como a robótica, realidade virtual, realidade aumentada, holografia, e internet de coisas.

Ainda falta completar sua interface para que toda a estrutura de conexão entre vértices/arestas/conjuntos/objetos 3D esteja disponibilizada para o EC, e ainda falta pesquisar a utilização dos tipos de objetos/conexões para representar os domínios, relacionamentos e termos das descrições, ou os elementos da realidade.

Quanto ao processo de avaliação de usabilidade, foram enumerados vários critérios para avaliar a conveniência de uso de ferramentas para a construção de ontologias. No futuro, poderão ser criados algoritmos para a avaliação automática dos critérios, durante o próprio uso da ferramenta pelos EC's.

A visão de mapas 3D já é uma realidade. Prova disso são as aplicações *Google Earth*, *Google Sketch-Up* e *Bing 3D Maps*. Há a intenção de associar tanto o processo de rotulagem multilíngüe de

objetos 3D quanto os objetos aos mapas 3D, ou objetos 3D, disponibilizados por essas ferramentas. De maneira mais direta, fazer o editor proposto “conversar” com as três ferramentas.

É necessário completar a ferramenta de construção de ontologias, para que os métodos de construção de instâncias de ontologias dos modelos de Bueno e Hoeschl entrem em produção, servindo para busca de documentos através do cálculo de valores de contínuos de similaridade entre documentos e através da determinação de domínios sucessivos, cujo princípio de busca se baseia nos métodos de RBC.

Finalmente, os temas relacionados a seguir, embora intensamente pesquisados, necessitam que se ampliem estudos específicos para auxiliar no processo de automação do método proposto de construção de ontologias: *Ontology Matching*; *Machine Translation*; *Word / Phrase Alignment*; *Projetos de sintaxe de scripts*; *Construção de Sistemas de Clusters de Computadores ou Grids*; *Métodos de parsing de textos em documentos disponíveis na Internet (Spiders / Crawlers)*; *Métodos de mineração de textos*; *Aplicação de ontologias para métrica de similaridade em Sistemas de Raciocínio Baseado em Casos*; *Uso das representações da Universal Modeling Language para a conversão de modelos de ontologias*. Tais tarefas devem ser iniciadas com o auxílio de participantes de futuros times de desenvolvimento nesta área.

## REFERÊNCIAS

ALANI, H. TGVizTab: An Ontology Visualisation Extension for Protégé. In Proceedings of Knowledge Capture (K-Cap'03), Workshop on Visualization Information in Knowledge Engineering , Sanibel Island, Florida, USA. 2003

BEDIN, S. P. M. Metodologia para validação de ontologias: o caso ORBIS\_MC. 2007. 113f. Dissertação (Mestrado em Ciência da Informação) – Programa de Pós-Graduação em Ciência da Informação, Universidade Federal de Santa Catarina, Florianópolis, 2007.

BEDIN, S. M.; OLIVEIRA, T. P. S.; SILVA, E. R. G.; BUENO, T. C. D.; TODESCO, J. L. Utilizando agentes inteligentes em um sistema de monitoramento de informações sobre gases do efeito estufa. In: 37 JAIIO - Simposio sobre la Sociedad de la Información, 2008, Santa Fé. 37 JAIIO. Buenos Aires: Sadio, 2008.

BERNERS-LEE, T. Semantic Web Road Map. Disponível em <<http://www.w3.org/DesignIssues/Semantic.html>> Outubro de 2008.

BERNERS-LEE, T., HENDLER J., LASSILA, O. The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. Scientific American Magazine. Disponível em <<http://www.sciam.com/article.cfm?id=the-semantic-web>> Outubro de 2008.

BLACKWELL, A.F., BRITTON, C., COX, A. GREEN, T.R.G., GURR, C.A., KADODA, G.F., KUTAR, M., LOOMES, M., NEHANIV, C.L., PETRE, M., ROAST, C., ROES, C., WONG, A.; YOUNG, R.M. Cognitive Dimensions of Notations: Design tools for cognitive technology. In M. Beynon, C.L. Nehaniv, and K. Dautenhahn (Eds.) Cognitive Technology 2001 (LNAI 2117). Springer-Verlag, pp. 325-341.

BOSCA, A., BONINO D.: OntoSphere3D: A Multidimensional Visualization Tool for Ontologies. DEXA Workshops 2006

BOSCA, A., BOMINO, D., AND PELLEGRINO, P.. OntoSphere: more than a 3D ontology visualization tool. In Proceedings of SWAP, the 2nd Italian Semantic Web Workshop, Trento, Italy, December 14–16, CEUR. Workshop Proceedings, ISSN 1613-0073, online <http://ceur-ws.org/Vol-166/70.pdf>. 2005

BRESCIANI, S., BLACKWELL, A., EPPLER, M. Choosing visualizations for collaborative work and meetings: A guide to usability dimensions. Darwin College Research Report – DCRR-007 Darwin College. Cambridge University. United Kingdom CB3 9EU. Disponível em <[www.dar.cam.ac.uk/dcrr](http://www.dar.cam.ac.uk/dcrr)> Darwin College Research Report – DCRR-007 - ISSN 1749-9194. Fevereiro de 2008

BRESCIANI, S., EPPLER, M. J. Usability of Diagrams for Group Knowledge Work: Toward an Analytic Description. I-KNOW 07, Graz, Austria, pp.416-423. Disponível em <<http://www.knowledge-communication.org/publications.html>> Maio de 2009.

BUENO, T. C. D. Criação de um Modelo de Domínio em UML para a construção de um Editor de Ontologias Multilíngüe. Florianópolis, 2008. Trabalho de Conclusão de Curso - Curso de Pós-Graduação em Engenharia de Software com UML. Faculdade de Tecnologia do SENAI, Florianópolis, 2008.

BUENO, T. C. D. ONTOJURIS. Edição Multi-Idiomas de Ontologias. Disponível em <<http://www.i3g.org.br/ontojuris/>> Maio de 2009.

BUSCHBAUM, A. R. V. Lógica Geral. Disponível em <[http://www.exe.inf.ufsc.br/~arthur/index.php?page=materiale\\_didatico&lang=pt](http://www.exe.inf.ufsc.br/~arthur/index.php?page=materiale_didatico&lang=pt)> Dezembro de 2009

CARACCILO C., SINI, M. KEIZER, J. (2007) Requirements for the Treatment of Multilinguality in Ontologies within FAO. OWLED 258. Disponível em <<http://www.neon-project.org/nw/Publications>> Maio de 2010

CARDEÑOSA, J., GELBUKH, A., TOVAR, E. - Universal Networking Language: Advances in Theory and Applications - Instituto Politécnico Nacional - Centro de Investigación en Computación. México 2005. ISSN 1665-9899. ISBN 970-36-0226-6. Criação de um Modelo de Domínio em UML para a construção de um Editor de Ontologias Multilíngüe <http://www.cicling.org/2005/UNL-book/> Novembro de 2009

CARDEÑOSA, J.; GALLARDO, C.; IRAOLA, L.; DE LA VILLA, M. A. A new knowledge representation model to support multilingual ontologies. A case study. IKE 2008 – The International Conference on Information and Knowledge Engineering. Nevada USA. Julho 2008.

CARDEÑOSA, J., GALLARDO, C., TONI, A. Multilingual Cross Language Information Retrieval. A new approach. CSIT 2009 - October,

2009, Yerevan, Armenia – Disponível em <<http://csit.am/2009/proceedings.php>> Janeiro de 2010

CKC CHALLENGE 2007 - Workshop on Social and Collaborative Construction of Structured Knowledge at 16th International World Wide Web Conference (WWW2007) - Banff, Canada, May 8, 2007. Disponível em <<http://km.aifb.kit.edu/ws/ckc2007/>> Maio de 2010

CLUSTY. Disponível em <<http://clusty.com/>> Novembro de 2008.

CLUUZ Disponível em <<http://www.cluuz.com/>> Novembro de 2008.

CLUUZ SEARCH BASICS Disponível em <<http://www.cluuz.com/SearchBasics.aspx>> Novembro de 2008.

COCKBURN, A.; MCKENZIE, D.. Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments. In Proceedings of ACM Computer-Human Interaction Conference on Human Factors in Computing Systems. ACM Press. 2002

DMITRIEVA, J., VERBEEK F.J. Information Visualization from Ontology.

Proc. 5th international conference on formal ontology in information systems. FOIS 2008

DMITRIEVA, J., VERBEEK, F.J. Node-Link and Containment Methods in Ontology Visualization. Proceedings OWLED 2009, Washington DC, USA. 2009. Disponível em <http://www.webont.org/owled/2009/>.

DMITRIEVA, J., VERBEEK, F.J. Multi-view Ontology Visualization. Proceedings 11th International Protege Conference. Disponível em <http://protege.stanford.edu/conference/2009/abstracts/S9P3Dmitrieva.pdf> f. 2009

GLOCK, A.; RAUTENBERG, S.; GUEMBAROVSKI, R. H.; HOESCHL, H. C.; TODESCO, J. L.; GAUTHIER, F. A. O. O uso de ontologias como ferramenta no auxílio à recuperação de informações em e-gov. IV CONeGOV, 2007, Palmas. v. 4. p. 184-193.

GOLEMATI, M., HALATSIS, C., VASSILAKIS, C., AND KATIFORI, A.. A context-based adaptive visualization environment. In Proceedings of the 10th Information Visualization Conference, IV06, London. 2006

GRAHAM, R. L. An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set. Information Processing Letters Volume 1 Número 4: 132-133 (1972)

GROWL WEBSITE University of Vermont. Gund Institute for Ecological Economics. Ecoinformatics Collaboratory. Disponível em <<http://www.uvm.edu/~skrivov/growl/>> Maio de 2010

HEER J., CARD, S. K., LANDAY, J. A. Prefuse: A Toolkit for Interactive Information Visualization. ACM Human Factors in Computing Systems (CHI), 421-430, 2005

HICKS, M., O'MALLEY, C., NICHOLS, S.; ANDERSON, B. Comparison of 2D and 3D representations for visualizing telecommunication usage. Behav. Inform. Tech., Vol. 22, No. 3, 185–2003

HOESCHL, H. C.; SECCO, D.; BUENO, T. C. D. ; OENNING, G. SERPI - System for Registration of the Industrial Property Production. IADIS International Conference WWW/Internet 2005, 2005, Lisbon, 2005.

HOESCHL, H. C.; BUENO, T. C. D.; NICOLINI, A. T. ; BEDIN, S. M. SAEI management system an application of knowledge-based system with ontologies for state crisis management. IV CONeGOV, 2007, Palmas. v. 4. p. 88-94.

HOESCHL, H. C. ONTOWEB: ONTOWEB: O NOVO FURACÃO DA INTERNET. JavaFree.org Disponível em <<http://javafree.uol.com.br/viewtopic.jbb?t=853570>> Maio de 2009.

HOESCHL, H. C. ONTOWEB®: FERRAMENTA INFORMACIONAL DE GOVERNO ELETRÔNICO Disponível em <<http://www.ontoweb.com.br/analise/SobreOntoWeb.htm>> Maio de 2009.

HORRIDGE, M. - OWLViz – A visualisation plugin for the Protégé OWL Plugin. Available in <<http://www.co-ode.org/downloads/owlviz/>> Accessed on August, 2010

HORRIDGE, M., KNUBLAUCH, H., RECTOR, A., STEVENS, R., WROE, C. A Practical Guide To Building OWL Ontologies Using The Protégé-OWL Plugin and CO-ODE Tools - Edition 1.0 - August 2004 - Protégé OWL Tutorial The University of Manchester Disponível em

<<http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/>> Novembro de 2009

HUNT, L. A PREVIEW OF HTML 5. Disponível em <<http://www.alistapart.com/articles/previewofhtml5>>. Outubro de 2008.

IATE ABOUT (Inter-Active Terminology for Europe) Disponível em <[http://iate.europa.eu/iatediff/about\\_IATE.html](http://iate.europa.eu/iatediff/about_IATE.html)> Novembro de 2008.

IATE (Inter-Active Terminology for Europe) Brochure. Versão em Português. Disponível em <<http://iate.europa.eu/iatediff/brochure/index.html>> Novembro de 2008.

ISKOLD, A. Overview of Clustering and Clusty Search Engine. Read Write Web. Disponível em <<http://www.readwriteweb.com/>> Setembro de 2008.

JAMBALAYA <<http://www.thechiselgroup.org/jambalaya>> Accessed on August, 2010

JAVA3D. Disponível em <https://java3d.dev.java.net/> Acessado em Agosto de 2010

JIM WISSNER and NOVA SPIVACK, Twine, USA - April 2009 - Semantic Web Use Cases and Case Studies - Case Study: Twine. Disponível em <<http://www.w3.org/2001/sw/sweo/public/UseCases/Twine/>> Maio de 2010

KATIFORI, A., HALATSIS, C., LEPOURAS, G., VASSILAKIS, C., GIANOPOULOU, E. Ontology visualization methods a survey. ACM Comput. Surv., 39(4):10+, 2007.

KOBSA, A.. User experiments with tree visualization systems. In IEEE Symposium on Information Visualization (INFOVIS'04). 9–16. 2004

KRIVOV, S., WILLIAMS, R., VILLA, F. GrOWL: A tool for visualization and editing of OWL ontologies. Web Semantics: Science, Services and Agents on the World Wide Web. Volume 5, Issue 2, June 2007, Pages 54-57. Software Engineering and the Semantic Web

LIU & ÖZSU. Encyclopedia of Database Systems, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag, 2009. Disponível em <<http://tomgruber.org/writing/ontology-definition-2007.htm>> Setembro de 2011

MACMANUS, R. Twine: The First Mainstream Semantic Web App? Read Write Web. Disponível em <<http://www.readwriteweb.com/>> Setembro de 2008.

MONTIEL-PONSODA, E., AGUADO DE CEA, G., GÓMEZ-PÉREZ, A., PETERS, W. Modelling Multilinguality in Ontologies. 2008. Disponível em <<http://www.neon-project.org/nw/Publications>> Maio de 2010

NAGY, M., VARGAS-VERA, M. Reasoning Introspection and Visualisation Framework for Ontology Mapping on the Semantic Web, Studia Universitatis Babes-Bolyai Series Informatica, Volume LV, Number 2, (2010)

NICOLINI, A. T.; BUENO, T. C. D.; BEDIN, S. M.; OLIVEIRA, T. P. S.; OENNING, G. O sistema inteligente de monitoramento de informações para a redução de gases de efeito estufa do setor madeireiro do planalto norte catarinense. In: IV CONeGOV, 2007, Palmas. v. 1. p. 215-222.

NOY, N. F., CHUGH, A., ALANI, H. The CKC Challenge: Exploring Tools for Collaborative Knowledge Construction (2007). Technical Report SMI-2007-1260, Stanford Medical Informatics, Stanford University. IEEE Intelligent Systems. Vol 23 No 1. January / February 2008. Disponível em <<http://eprints.ecs.soton.ac.uk/14793/>> Maio de 2010

OENNING, G.; NASCIMENTO, E. B.; HOESCHL, H. C.; SERPI - System For The Registration Of The Intellectual Property Production For Technological Innovation Nucleus. In: 35° JAIIO, 2006, Mendoza.

ONTVIS, 2010. Multi-view ontology visualization. Disponível em <http://www.liacs.nl/~jdmitrie/ontVis/OntologyVisualization.html>

NOY, N. F.; MCGUINNESS, D. L.; Ontology Development 101: A Guide to Creating Your First Ontology. Stanford University, Stanford, CA, 94305. Disponível em <[http://protege.stanford.edu/publications/ontology\\_development/ontology\\_101-noy-mcguinness.html](http://protege.stanford.edu/publications/ontology_development/ontology_101-noy-mcguinness.html)> Maio de 2010

O'REILLY, T. Web 2.0 Compact Definition: Trying Again. Disponível em <<http://radar.oreilly.com/archives/2006/12/web-20-compact.html>> Outubro de 2008.



OWL Web Ontology Language - Use Cases and Requirements - W3C Recommendation 10 February 2004 - Disponível em <<http://www.w3.org/TR/webont-req/>> Maio de 2010

OWL Web Ontology Language Overview - W3C Recommendation 10 February 2004. Disponível em <<http://www.w3.org/TR/owl-features/>> Maio de 2010

PENDER, TOM. UML, A Bíblia; Tradução de Daniel Vieira. – Rio de Janeiro: Elsevier, 2004 – 2ª Reimpressão. Tradução de: UML Bible. ISBN 85-352-1408-9

PLAISANT, C., GROSJEAN, J.;BEDERSON, B. B. 2002. SpaceTree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In Proceedings of IEEE Symposium on Information Visualization, Boston, 57–64.

PREFUSE Information Visualization Toolkit. Disponível em <<http://prefuse.org/>> Novembro de 2008

PROTÈGÈ <<http://protege.stanford.edu/>> Accessed on August 2010

RACERPRO Disponível em <<http://www.racer-systems.com/>> Dezembro de 2009

RAUTENBERG, S. ; GAUTHIER, F. O. ; LOTTIN, P. ; DUARTE, C. E. J. ; TODESCO, J. L. . ontoKEM: uma ferramenta para construção e documentação de ontologias. In: I Seminário de pesquisa em ontologia no Brasil, 2008, Niterói.

RDF PRIMER Disponível em <<http://www.w3.org/TR/rdf-primer/>> Maio de 2010

SMALLMAN, H. S., ST. JOHN, M., OONK, H. M.; COWEN, M. B.. Information availability in 2D and 3D displays, IEEE Comput. Graph. Appl., vol. 21, no. 5, pp. 51–57, Sept/Oct. 2001

SPRYLOGICS Disponível em <<http://www.sprylogics.com/>> Novembro de 2008.

STRADIOTTO, C. R. K., BORTOLON, A., HOESCHL. H. C.; Ferramenta de Desenvolvimento de Software para Representação Visual de Redes de Relacionamento. SUCESU 2004. Disponível em <[www.sucesu2004.org.br](http://www.sucesu2004.org.br)> Janeiro de 2008

STRADIOTTO, C. R. K.; PACHECO, E.; BORTOLON, A.; HOESCHL, H. C. A Graphic Tool for Ontology Viewing Based on

Graph Theory. Professional Practice in Artificial Intelligence, IFIP 19<sup>th</sup> WCC. v. 218, p. 141-150, Santiago. Chile. 2006

STRADIOTTO, C. R. K.; PACHECO, E.; BORTOLON, A.; HOESCHL, H. C. The RR Project. A Framework for Relationship Network Viewing and Management. Professional Practice in Artificial Intelligence, IFIP 19<sup>th</sup> WCC. v. 218, p. 121-130, 2006

STRADIOTTO, C. R. K.; PACHECO, E.; BORTOLON, A.; HOESCHL, H. C. Projeto Redes de Relacionamento. In: Conferencia Latinoamericana de informática, 2006, Santiago. p. 51-52.

STRADIOTTO, C, R. K., BUENO, T.C.D., HOESCHL, H. C. Ontology Graphical Editor for Multilingual Document Search System. FQAS 2009: 453-464

TUDORACHE, T., NOY, N. F., TU, S., MUSEN, M. A. Supporting Collaborative Ontology Development in Protege - Seventh International Semantic Web Conference, Karlsruhe, Germany, Springer. 2008 - Disponível em

<[http://bmir.stanford.edu/file\\_asset/index.php/1374/BMIR-2008-1331.pdf](http://bmir.stanford.edu/file_asset/index.php/1374/BMIR-2008-1331.pdf)> Maio de 2010

TWINE. Disponível em <<http://www.twine.com/>> Novembro de 2008.

TWINE: THE TECHNOLOGY. Disponível em <<http://www.twine.com/technology>> Novembro de 2008.

TWOUSE. SEMANTIC WEB + MODEL DRIVEN DEVELOPMENT. Disponível em <<http://code.google.com/p/twouse/>> Maio de 2011

UNIVERSAL NETWORKING DIGITAL LANGUAGE FOUNDATION Disponível em <[www.unl.org/](http://www.unl.org/)> Setembro de 2008.

UNIVERSAL NETWORKING LANGUAGE. Specifications. Version 3, Edition 2. Disponível em <<http://www.unl.org/unlsys/unl/UNL%20Specifications.htm>> Janeiro 2009.

UNIVERSAL WORDS DICTIONARY. Disponível em <<http://www.unl.fi.upm.es:8099/unlweb/>> Janeiro de 2010.

VALDES-PEREZ, R. INTRODUCING CLUSTERING 2.0 Disponível em <<http://searchdoneright.com/2008/01/introducing-clustering-2.0/>> Novembro de 2008.

VALDES-PEREZ, R. HOW TO EVALUATE A CLUSTERING SEARCH ENGINE. Disponível em <<http://searchdoneright.com/2007/03/how-to-evaluate-a-clustering-search-engine/>> Novembro de 2008.

WATTENBERG, M., Introduction to Visualization. From data to pictures to insight. Visual Communication Lab. IBM. 16 May 2005 Disponível em <<http://www.alphaworks.ibm.com/contentnr/introvisualization>> Novembro de 2008.

WATZMAN, A. or KURINSKY, K. Yahoo! Internet Life Names Vivisimo Search Engine Developed at Carnegie Mellon The Best New Search Service on the Web Disponível em <[http://www.cmu.edu/PR/releases01/010613\\_vivisimo.html](http://www.cmu.edu/PR/releases01/010613_vivisimo.html)> Novembro de 2008.

WISS, U., CARR, D.; JOHNSON, H.. Evaluating three-dimensional visualization designs: A case study of three designs. In Proceedings of the Second International Conference on Information Visualisation (IV'98). p. 137. 1998

WORDNET ABOUT Disponível em <<http://wordnet.princeton.edu/>> Princeton University. Janeiro de 2010.