

A Graphic Tool for Ontology Viewing Based on Graph Theory

César Stradiotto¹, Everton Pacheco¹, André Bortolon¹, Hugo Hoeschl²
1 WBSA Sistemas Inteligentes SA, Parque Tecnológico Alfa, Centro de
Tecnologia IlhaSoft, SC 401 Km 1 - Módulo 10 - Térreo B - João Paulo -
88030-000 - Florianópolis, SC – Brasil
{cesar,evertonpp,bortolon,hugo}@wbsa.com.br
WWW home page: <http://www.wbsa.com.br>
2 Instituto de Governo Eletrônico, Inteligência Jurídica e Sistemas –
IJURIS, Rua Lauro Linhares, 728 – sala 105 – Trindade - 88036-0002 -
Florianópolis – SC – Brasil
WWW home page: <http://www.ijuris.org>

Abstract. The Knowledge Engineering Suite is an ontology production method, based on relationship networks, for knowledge representation inside specific contexts. The production of these ontologies has three basic steps, since catching the client data, knowledge base creation, and information retrieval and consult interfaces for the final users. During the knowledge base creation process, data verification is required, for nonconformity identification on the produced ontological network. Because it has a tabular interface, the verification step has some limitations about data vision, and consequently the tool usability, making the work for tracking errors or missing data uncomfortable, and susceptible to more errors. To make easier the vision of the created ontologies, in the real shape they are planned, it was implemented a software to viewing these new created ontologies, so the work for data error tracking became more efficient. Such software offers filtering and data selection resources too, to give a way to isolate common groups, when the Knowledge Engineer is looking for nonconformities. This software and its functions are described here.

Introduction

On its classical definition, ontology is a branch of metaphysics that deals with the “being”, including there the natural theory and being types. (HOESCHL [1]). GRUBER *apud* BATEMAN [2,3], borrows the definition from philosophy for the ontologies: a systematic narrative of the existence, and define an ontology as an explicit specification of a conceptualization, having a set of representative terms linked by definitions. Such ontology definition is further detailed by DENNY [4], that observes two main approaches as a common core for virtually all ontologies: 1 - A vocabulary of terms that refer to the things of interest in a given domain; 2 - Some specification of meaning for the terms, [ideally] grounded in some form of logic.

According with DENNY [4], ontologies are arranged inside taxonomies of classes (called *concepts*) and subclasses, having its properties and restrictions. Finally, the union of ontology with a group of instances of its classes constitutes a *knowledge base*.

In DENNY [4,6] there is an excellent reference for ontologies and its corresponding tools, differentiated by application space, storage type, particularities, good and bad points about everyone of them.

1 The Data Structure Model for the Ontology Network

On the ontology model of BUENO [7], Fig. 1, a given ontology can be represented by a directed graph, taking its basic elements: vertices (the **terms**, or ontology concepts) connected by straight lines (called **relations**). An ontology containing terms and relations, is inside a group called sub-theme, which represents a specific subject, and, for last, many sub-themes can be inside a theme, that represents a generic subject. The domain is the last layer, that has inside itself many themes. Such model was constructed with the methodology described in HOESCHL [8].

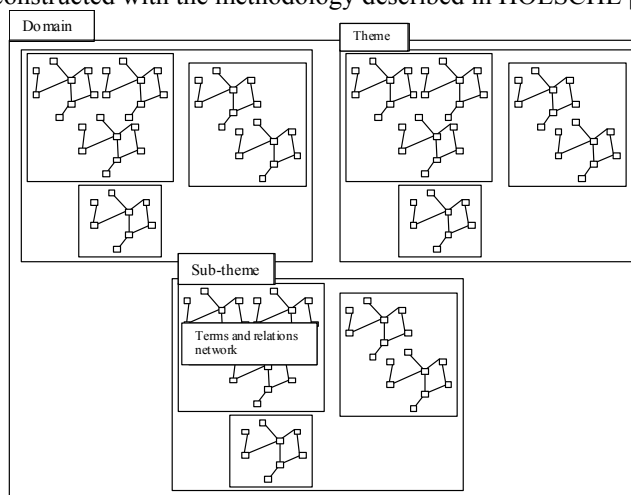


Fig. 1: Ontology structure for the knowledge base read by the software.

2 Knowledge Engineering Suite

Knowledge Engineering is a definition used to describe the global process of system development, based on knowledge. Typically it evolves a special form of interaction between the system builder, called Knowledge Engineer, and the professional specialist about the worked subject domain. Being by nature a multidisciplinary field, the knowledge engineering comprehends research types, which have a difficult classification, if this research is inside a limited approach (VALENTE [9]).

The Knowledge Engineering process main objective is to catch and incorporate the domain specialist fundamental knowledge, and its prognostics and control systems. This process includes information grouping, turning the knowledge engineer familiar to the subject domain, project analysis and a big effort. Beyond that, the cumulated knowledge may be codified, tested and refined (BUENO [10]).

The knowledge engineer extracts procedures, strategies and practical rules from the specialist professional, commonly used to the problems solutions, and constructs this knowledge inside an intelligent system. The final result is a software project, which solves specific problems like human specialists (BUENO [10]).

Aiming an information retrieval tool inside the enterprise knowledge bases, to allows a fast information recovery by its employees / users, BUENO [7] describes a method for ontology construction, denominated Knowledge Engineering Suite, composed basically of four basic steps (Fig. 2):

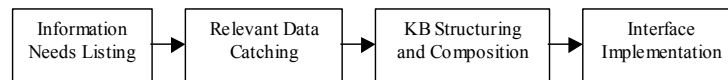


Fig.2: Basic steps to create information retrieval tools.

1. Listing the information needs, inside the client enterprises;
2. Catching the relevant data to the context of the information that may be represented;
3. Structuring and composition of the knowledge bases, which contains that data;
4. Implementation of the final user interfaces, with searching tools, which work over those knowledge bases.

The suite is characterized for grouping a set of professionals, tools and production procedures, which starts from capturing the implicit knowledge inside client enterprises, and results in the creation of information recovery resources to these same enterprises.

This production system is made by three types of professional: The system analyst, which takes care of the technical part of the implementation, the domain specialist, who knows the information needs inside his enterprise, and the knowledge engineer, which works over the knowledge base production over what the retrieval information system will work.

After the expected vocabulary survey is done, its concepts (also called "terms") and relations are included inside databases through collaborative tools, using navigable pages. This data inclusion is done at same time by many knowledge engineers, who evaluate what representative terms and relations have to make part of the edited ontology.

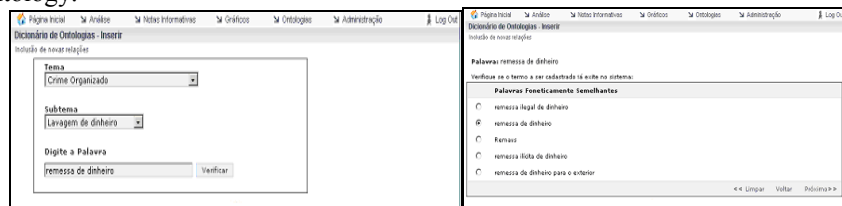


Fig. 3: Two pages of an interface for data inclusion inside an ontology.

During data edition, the navigation is made with steps. Each step is presented as a dynamic page, so the user loses the vision of the process as a whole. In Figure 3 there are two sample pages, used in data edition for only one ontology project.

After data inclusion, it is made its revision through keyword-based search engines, and terms and relations listing. Although being planned as relationship networks, such ontologies are still managed as table data (Fig. 4). The more the data is added on the knowledge base, more difficult, slower and stressing is its management, demanding

greater attention from the knowledge engineer responsible for its production, and increasing the risks for inconsistencies and correction work, what represent a bottleneck over ontology production.

Tema	Sub tema
Crime Organizado	Lavagem de dinheiro

Devolução de verbas públicas	agente de dinheiro público - Sinônimo
devolução de verbas públicas	devolução de verbas públicas - Sinônimo
través de dinheiro público	través de dinheiro público - Sinônimo
sonstigeção de verbas públicas	sonstigeção de verbas públicas - Sinônimo
través de verbas públicas	través de verbas públicas - Sinônimo
agente de recurso público	agente de recurso público - Sinônimo
Operação Oufanudo	Operação Oufanudo - Conexão
Operação Fuga de Dólar	Operação Fuga de Dólar - Conexão
lavagem de dinheiro	Isso é parte de
repressão à corrupção	repressão à corrupção - Sinônimo
repressão à corrupção	repressão à corrupção - Sinônimo
lavagem de dinheiro	Isso é parte de
repressão à lavagem de dinheiro	repressão à lavagem de dinheiro - Sinônimo
repressão à lavagem de dinheiro	repressão à lavagem de dinheiro - Sinônimo
combate à reciclagem de dinheiro	combate à reciclagem de dinheiro - Sinônimo
repressão à reciclagem de dinheiro	repressão à reciclagem de dinheiro - Sinônimo
repressão à reciclagem de dinheiro	repressão à reciclagem de dinheiro - Sinônimo
recuperação de ativos ilícitos	recuperação de ativos ilícitos - Conexão
recuperação de patrimônio de origem ilícita	recuperação de patrimônio de origem ilícita - Conexão
lavagem de dinheiro	Isso é parte de

Fig. 4: Sample of a table formatted generated report about ontology terms e relations.

3 The Proposed Solution

Disposing from an already existent framework, for network viewing (STRADIOTTO [11]), and observing the lack of production continuity, and the limitation given by table format data vision, it was suggested an integrated graphic solution, which allowed the knowledge engineer to verify its production results in a network format, inside a tool that integrated all navigation and searching resources inside a single environment, and which reused all functionalities from that framework.

3.1 Relevant Ontology Properties

The creation of a software that allowed ontology data vision while it was being created, considered the following properties:

1. Ontologies are relationship networks, and have their simplest format as a directed graph made by its vertex (its concepts) connected by edges (the relations between vertex). Its graphic representation can include geometrical and design elements like vertex length, line width, colors and term symbol.
2. Ontologies are constantly being updated, with more and more included terms and relations, by the fact that it represents a vocabulary domain, which is on continuous changing, and incorporate inside it new used terms used in knowledge bases inside client enterprises. So it's necessary that all these data have to be seen as part of a whole network, inside an integrated environment, also including searching and data treatment tools.
3. Terms and relations have their internal data, which refers, respectively the concepts, and the relations between them.

3.2 Tool Requirements for Graphical and Integrated Data Presentation

Observed the Knowledge Engineering Suite needs, and the ontology properties seen above, the proposed integrated environment, which is created to solve the verification step, have to satisfy the following objectives:

1. Showing nodes and corresponding connections like an interlinked network with vertex and edges, with its properties represented as graphical identifiers, like formats, colors and drawing styles;
2. Loading ontology data according to user selected criteria;
3. Positioning clearly these data in the screen, manually or automatically, in cases where the user requires it;
4. Allowing whole or partial network vision, immediately, according to knowledge engineer chosen criteria;
5. Allowing user seeing internal information from the elements inside the graph;
6. Offering the user searching and data refinement resources;
7. Allowing user persisting network transformations, for ulterior consult;
8. Showing to the user an integrated environment with all verification tools needed to manage the new produced ontologies.

Satisfying these objectives, data control and verification problems can be solved, and knowledge engineer gain continuous production and efficiency on ontology production.

3.3 Project Considerations and Limits

The software implementation for network like ontology viewing had as premises:

1. The existent framework classes for network like data viewing were reused;
2. The software loads data from only one type of database format, with only one type of knowledge base structure;
3. This data loading is done considering the following initial pre-selection criteria: chosen theme and sub-theme, initial term and connected ones, initial and final date and time for data inclusion;
4. Terms are represented by icons or points, together with internal texts or not;
5. Relations are represented by edges, colored accordingly its type, with style accordingly to the storage situation on the knowledge base;
6. Data refinement can be done by the same criteria on item 7.3, including type of relation to be shown;
7. Due to the existence of another types of knowledge bases, the software will have generalized some items, which will compose a derived framework, made to ontology network management;
8. The new application is Multiple Document Interface type, which can show the load data through many integrated, but independent windows;
9. The application does not persist the edited network ontology, but only the final picture of the graphic result, serving only for visual purposes.

3.4 The Final Result and Working Model

The resulting software, its load and filter process and some of its main features are described as follows. In figure 6, left side, is shown the main screen, having four data group samples.

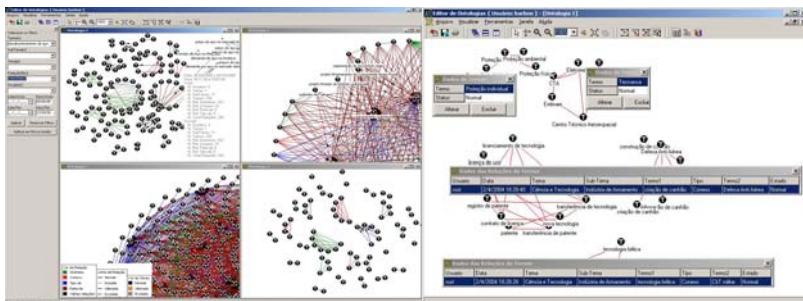


Fig. 5: Ontology viewer main screen, and at its right side many windows showing internal terms and relations properties.

The software has its data working made with 3 steps: 1 – Loading, 2 – Edition and filtering, 3 – Picture persistence

The available tools are the following:

Selective data loading - There are five non-exclusive criteria to chose, before loading knowledge base data: Theme, Sub-theme, Term and neighbors, Specific Relation, and Time Interval.

Internal data viewing (Fig. 5, right side) - Terms and relations internal properties can be seen through specific data windows.

Data edition (Fig. 6, left side) – According with its needs, user can dispose data manually, with manual operations over data groups, or over all loaded network.

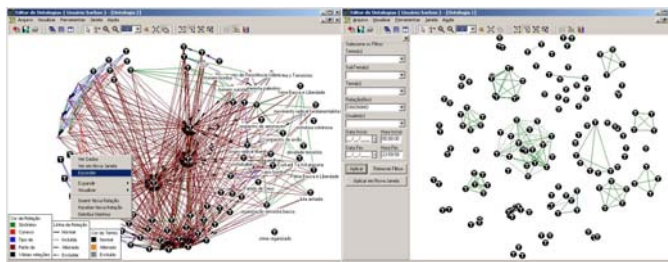


Fig. 6: Menu with operations over groups or isolated data. At its right side, a filter applied only over relations

Data drawing options - Edges show internal data referring to its represented relation, and to its state inside the knowledge base, so as the vertex with its internal parameters. The knowledge engineer can chose the way these data are shown on screen, changing parameters like color or style on edges or color on vertex.

Filtering (Fig. 6, right side): In the cases of ontology networks with a large amount of data, the filtering tool can help the user to visually select data by its internal properties, like edition time, by whole texts or wildcards like ? and * (MS-DOS) for vertex, or its corresponding theme or sub-themes, and relations by its type.

Network union (Fig. 7): if the user needs to verify if some terms are contained inside two different themes or sub-themes, he can use the network union tool. It

corresponds to the fusion of two different sets of data. Identical data are not duplicated after the union operation.

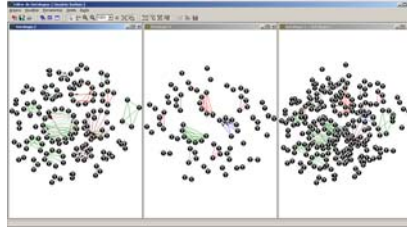


Fig. 7: Windows showing two initial networks, and the result after network union operation.

Persistence of visual disposition of data: For situations where it is needed report generation or ontology production generation, data visual disposing on planar space can be stored on bitmap or jpeg style pictures.

4 Complementary research approaches

Beyond accomplish the needs of data verification, the application was a research point about graph drawing and wildcard-based text search algorithms, and a research point for library modularity for code reuse.

4.1 Graph Drawing Algorithms for Ontology Network

It was proposed the research of two simple algorithms of vertex distribution on two-dimensional space. These were based on a circular distribution, due to its relatively easy implementation and fast drawing, with different principles:

N-Part Radial Disposal

This distribution makes a division of the total angle (2π radians) in N parts, chosen by the project manager. Figure 8, left side, has a 10 step angular distribution. The distribution circle starts from the smallest radius (zero, on this case), and it's increased every time the angular counter completes N steps (in this case $N = 10$). Inside the network are three disconnected vertex, so they are drawn apart of the major vertex group.

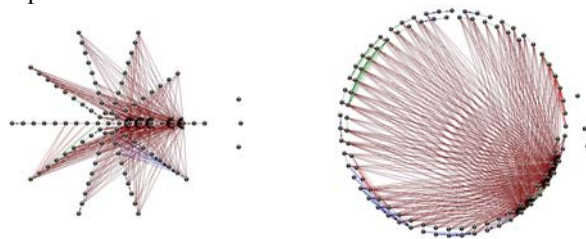


Fig. 8: 10-Part Radial Disposal, right side: Radial Disposal by CAL.

Radial Disposal by Constant Arc Length

In this model, the vertex are been accumulated over a circumference in such a way that a given vertex stays at a constant radial distance from its immediate neighbor.

This distance is measured by the arc length that remains between the two following vertex. As the vertex are positioned, there is an angle increment I that varies accordingly with the actual radius, from circumference which is actually being covered. The initial circumference has a maximum radius, and is being reduced as the vertex reach 2π radians (Fig. 8, right side).

4.2 Textual Search through Wildcards

In any ontology network, similar terms inside its knowledge base occurs frequently, be it because the terms have identical stems, suffix, or because two terms have identical textual sequences. Aiming to make easy the search for groups of terms with textual similarities, the software has a filtering tool that works with wildcards: the symbol “*”, that means any text with any length, or the symbol “?”, meaning a single position inside the searched text, similarly to the file searching command **DEL** used in the Microsoft Disk Operating System (MS-DOS).

Figure 9 shows an initial data group, with two applied wildcard filters: “**terr***”, for the result case shown at the left-bottom, and “***o?a***” for the right-bottom result case. Obviously, more complex cases can be tried.

On the left-bottom case (string “**terr***”), the filter shows only the terms which starts with the string “**terr**” and finishes with any string: **terrorismo químico-biológico**, **terrorismo islâmico**, **terrorismo**, and **terrorismo radical**. The immediate neighbors of these selected terms are also shown.

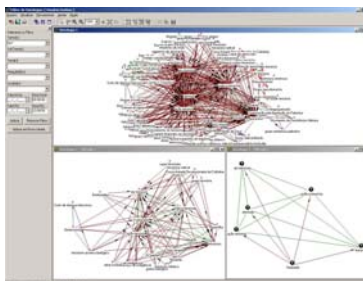


Fig. 9: Wildcard filtering: Made from the greatest data group. Left: filtering by the string “***o?a***”. Right: filtering by the string “**terr***”.

On the right-bottom case, only terms which starts with any string followed by the letter “**o**”, with any character inside (situation represented by “?”), followed by the letter “**a**”, and has any string following it, is selected. For coincidence, only the term with the string “**Hezbollah**” was chosen, and also its neighbors.

5 Network Viewing Libraries and Similar Work on Ontology Viewing

At Internet there are some available *frameworks* or libraries for network viewing: GRAPHLET [12], BROWN [13], LINK [14], JGRAPHT [15], JGEF [16], JUNG, [17], or graphical interface application for ontology edition, like KAON [18], KAON2 [19], MR3 [20], ONTOTRACK [21] and PROTÉGÉ [22].

6 Conclusions

Comparing the suggested solution with the obtained results, there is a notable evolution about the ontology revision speed. With the application, the following objectives were reached:

1. The software shows the edited ontologies as a two-dimensional network;
2. The data load, network viewing, manipulation, and data search visual modules are all integrated inside the same environment;
3. The data-filtering module is based on multiple criteria: specific term, sub-themes and themes with wildcards, relation types, and time interval;
4. There are features for vertex selection, and viewing formats management for vertex and edges. So, interesting groups of data to knowledge engineers can be detached from another groups, turning easier the verification step on ontology production;
5. Users can manage colors and drawing styles of the described relations inside the knowledge base, and also vertex colors, accordingly its situation inside this same base;
6. The visual results can be persisted in raster pictures (jpeg or bitmap formats) for ulterior consultancy and reports.
7. The framework described on STRADIOTTO [11] had its classes reused, avoiding code repetition;
8. Terms and relations internal data can be identified almost immediately, due to information shown through colors and shaping on graphic mode;

Finally, knowledge engineers had a considerable time gaining due better usability with the new graphical interface, during ontology verification and report.

Although this advances, there are still some items to be approached on a near future:

1. The system doesn't do textual reports, but only pictures of the actual visual distribution of the ontological data;
2. The software works over only one database, inside only one data structure, needing further adaptations for cases where new database formats and structures are used;
3. It would be interesting allowing the software to write directly on the managed knowledge bases, being a complete knowledge management system. Actually it can only read data;
4. About data protection, the knowledge bases are protected with ciphered data. This type of protection demands creation or use of translation libraries, which are not implemented inside the actual system;

Bibliography

1. HOESCHL, H. C. ; BUENO, Tânia Cristina D' Agostini ; SANTOS, Cristina Souza ; MATTOS, Eduardo da Silva ; BORTOLON, Andre . The study of ontologies in the development of intelligent systems for the management of juridical knowledge. 34^a Jornadas Argentinas de Informática e Investigación Operativa, 2005, Rosario, 2005.
2. BATEMAN, J. John Bateman's Ontology Portal. <http://www.fb10.uni-bremen.de/anglistik/langpro/webpace/jb/info-pages/ontology/ontology-root.htm> - Accessed on december 2005.

3. GRUBER, T. . A translation approach to portable ontology specifications. Knowledge Acquisition 5:199-220; 1993
4. DENNY M. Ontology Building: A Survey of Editing Tools. November, 2002. XML.com <http://www.xml.com/pub/a/2002/11/06/ontologies.html>
5. USCHOLD, M. GRUNINGER, M. Ontologies and Semantics for Seamless Connectivity. SIGMOD Record, Vol. 33, No. 4, December 2004.
6. DENNY M. Ontology editor survey results. November, 2002. XML.com http://www.xml.com/2002/11/06/Ontology_Editor_Survey.html
7. BUENO T. C. D et al. Knowledge Engineering Suite: a Tool to Create Ontologies for Automatic Knowledge Representation in Knowledge-based Systems. in: The 2nd International Workshop on Natural Language Understanding And Cognitive Science (NLUCS-2005) in ICEIS - 7th international conference. Proceedings of 7th International Conference On Enterprise Information Systems. 2005.
8. HOESCHL, Hugo. C. BUENO, Tania. C. D., BARCIA, Ricardo. M., BORTOLON, Andre., MATTOS, Eduardo Da Silva. Olimpo: Contextual structured search you improve the representation council of UN security with information extraction methods In: ICAIL 2001 Proceedings. New York: ACM SIGART, 2001, p.217 – 218.
9. VALENTE, A. Legal Knowledge Engineering: A modeling approach. IOS Press: Amsterdam, 1995. p8.
10. BUENO, T. C. D. Engenharia das Mentas. (Tese de doutorado). Engenharia de Produção. Universidade Federal de Santa Catarina. 8 de setembro de 2004.
11. STRADIOTTO, C. R. K.; BORTOLON, A.; HOESCHL, H. C.; MARAFON, M. J.. Ferramenta de Desenvolvimento de Software para Representação Visual de Redes de Relacionamento. In: CONGRESSO NACIONAL DE TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO, 2004. <http://www.sucesu2005.com.br/palestras2004/24.html> . Acessado em Janeiro de 2006
12. GraphLet, A toolkit for graph editors and graph algorithms. <http://www.infosun.fmi.uni-passau.de/Graphlet/>. Acessado em dezembro de 2005.
13. Brown, J., Data Structure Project. <http://www.helsinki.fi/~jbrown/tira/overview.html>. Acessado em janeiro de 2006.
14. LINK -- Programming and Visualization Environment for Hypergraphs. <http://www.cs.sunysb.edu/~algorith/implement/link/implement.shtml>. December 2005
15. JGraphT. <http://jgrapht.sourceforge.net/> Acessado em janeiro de 2006
16. JGEF. Java Graph Editing Framework. <http://gef.tigris.org/> Accessed on december 2005
17. JUNG. Java Universal Network/Graph Framework. <http://jung.sourceforge.net/> Accessed on January 2005
18. KAON. In: <http://kaon.semanticweb.org> Accessed on January 2006.
19. KAON2. In: <http://kaon2.semanticweb.org/> Accessed on January 2006.
20. MR³. Meta-Model Management based on RDFs Revision Reflection. In: <http://panda.cs.inf.shizuoka.ac.jp/mmm/mr3/> Accessed on January 2006.
21. ONTOTRACK. In: <http://www.informatik.uni-ulm.de/ki/ontotrack/> Acesso em 17 de janeiro de 2006.
22. PROTÉGÉ. In: <http://protege.stanford.edu/index.html> Accessed on January 2006.