Hermes: a Semantic Web-Based News Decision Support System

Jethro Borsje
jethro@jesdesign.nl

Leonard Levering
leonard@levering.eu

Flavius Frasincar
frasincar@few.eur.nl

Erasmus University Rotterdam
PO Box 1738, NL-3000
Rotterdam, the Netherlands

ABSTRACT

The emergence of the Web has made more and more news items available, however only a small subset of these news items are relevant in a decision making process. Therefore decision makers need an information system that is capable of extracting a set of relevant news items automatically. This paper proposes a framework that provides decision makers with the ability to extract a set of news items related to specific concepts of interest. This is accomplished by creating a knowledge base and developing a system that classifies news with respect to the knowledge base.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*User-centered design*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Representation languages*

General Terms

Design, Management

Keywords

Decision Support Systems, Ontology, Semantic Web

1. INTRODUCTION

The amount of information available on the Web is steadily growing. Most of this information has no associated semantics which means that, although it is perfectly human understandable, machines can not reason with it. The Semantic Web [2] project of the World Wide Web Consortium (W3C) aims to provide a framework that solves this issue. For this purpose W3C proposes common formats to structure data, that will not only make the data machine understandable, but also make it interchangeable amongst computers. One of these formats is the Web Ontology Language (OWL) [1], which is an ontology language that can formally describe the meaning of concepts [10]. When data is properly annotated using OWL, a computer should be able to reason with it, making it easier for computers to integrate the data with other information.

A concrete example of a field that has a lot of unstructured information on the Web is the news domain. The Web has become an important platform to spread news around the world. News items appear on the Internet in large quantities, this information overload making it hard to stay up to date. Because of the huge amount of news items available, it is hard to extract only the relevant news items. If all these news items were properly annotated, it would be more easy to extract relevant news items by making use of domain specific ontologies.

In some fields, like the stock market, news items are very important inputs into the decision making process. News messages have a big impact on the prices of stocks and should be carefully monitored by stock market analysts. As the amount of news sources that are available have grown tremendously with the rise of the Web over the last decade, this task is difficult to be done manually. Existing approaches like Google Finance[1], despite being able to filter the news directly related to a certain portofolio (e.g., Google) do not allow users to access other relevant news items which are indirectly related to the portofolio (e.g., competitors of Google like Microsoft). These indirectly related news items play also a role in determining the share price of a company.

In order to support the news items analysts in their daily tasks, and to show a useful application of Semantic Web technologies, we propose Hermes. Hermes is a framework for searching news sources on the Internet, based on Semantic Web technologies. By using these technologies, clear semantics can be assigned to the various news messages, thereby providing a basis for a structured overview of the news. This structured overview enables users to extract relevant news items.

The technologies that we used include the data description languages RDF (Resource Description Framework) [4, 15], OWL (Web Ontology Language) [1], and the corresponding query language SPARQL (SPARQL Protocol And Query Language) [18]. Our approach is based on a conceptual model for storing concepts, news items, and the relations between these concepts and news items in an ontology. In

[1] `http://finance.google.com/finance`

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil

this paper, we present a useful example of the application of Semantic Web technologies to solve a real world problem. We will focus on the NASDAQ stock market domain for the news classification [14].

The paper is organized as follows. In the next section we discuss the work, related to our research. The section thereafter explains the news classification process, while the next section discusses the user interface of our software and how the user can interact with the system. This is followed by a description of the result extraction and visualization process. The last section summarizes our findings, and presents our conclusions.

2. RELATED WORK

This section gives a brief introduction about research in the fields of news classification and ontology visualization.

2.1 News Classification

Recently there has been an increasing number of news items delivered in a semi-structured format, called Really Simple Syndication (RSS) [24]. The RSS format delivers a news item together with its meta-data. The meta-data contains the time of release, a link to the original item, etc. Most press agencies, such as Reuters and AFP, provide RSS feeds that contain news items that cover a specific news category. As these feeds contain meta-data and are category based, the news items contained by them can be called semi-structured news items. These news feeds can be processed with the help of Natural Language Processing frameworks such as GATE [6] (General Architecture for Text Engineering). The GATE framework contains components such as the semantic tagger, the ortho matcher, and the ontology gazetteer, which can be used to analyze the news items. Although these components are provided by the GATE framework, GATE does not provide a complete methodology for news item classification. In the next paragraph we will discuss a complete methodology.

The SemNews [11] application aims to extract meaningful information out of news items, which appear on the Internet in news feeds. The news items are analyzed by OntoSem [12], SemNews's underlying natural language processing engine, and are classified with respect to their meaning. After this, they are stored together with their semantics in an ontology using OWL. By making use of this method, news items that have been published all over the Web can be stored in a structured way, to make them more accessible for the user. However, SemNews lacks an easy and powerful way to search through the news. This makes the extraction of relevant news items a cumbersome job.

myPlanet [13] is another ontology-based news classification system. In addition to news classification, it also personalizes the application by retrieving only the news items that interest a certain user. It does so by allowing the user select the desired concepts from the ontology and relies on heuristics to extend the result set. Nevertheless, the classification approach is not fully automated and the user cannot exploit the graph nature of the ontology in order to define its preferences.

2.2 Visualization

In the past few years visualization of ontologies has become a popular research field. A lot of the text-based ontology editors are unable to cope with the large amounts of data which usually resides in ontologies. These editors are very often unable to present the data in a way that allows for easy navigation and understanding of the information with respect to its context [5]. Because the data in an ontology can easily consist of thousands of entities, it is more appropriate to use visual tools rather than textual tools to present an ontology. This enables users to effectively navigate the ontology, understand the complex data structure, and interpret the relationships between entities.

Protégé [16], for example, is an ontology editor that supports hierarchical views based on the sub- and superclass relationships in OWL, using a tree layout. This is cumbersome if the user wants to understand the graph-like structure which is inherent to the use of RDF and OWL data [21]. By making use of plugins it is possible to obtain different views of the ontology. An example of one of these plugins is OntoSphere [3], which displays ontologies in a three-dimensional space. Another plugin is Jambalaya [20], which uses a hierarchical multi-perspective visualization technique to show the ontology. OntoViz [19] shows the data as a simple graph, which can be customized to the needs of the user.

IsaViz [17] is a visual environment for browsing and authoring RDF models. It is capable of showing both RDF and OWL data in a graphical way, thereby making use of a graph layout algorithm which determines the location of the elements. Despite the good quality of the drawn graph IsaViz is unable to load large ontologies, as the visualization algorithm is too slow, the drawing time of the graphs growing exponentially. Furthermore the graphs produced by IsaViz look very fuzzy when large datasets are used.

GViz [22] is a general-purpose visual environment which facilitates browsing and editing of graph data. It provides users with the ability to create customized visualization scenarios which are able to suit their specific needs. An RDF data format plugin for GViz is proposed, which creates the possibility to use GViz to generate customized visualizations of RDF data [8]. As GViz provides the user with many options and advanced visualizations scenarios, it might not be usable for relatively inexperienced users, whom may not understand the meaning of the complex visualizations.

3. CLASSIFYING NEWS

In this section of the paper we describe the process of classifying news within our Semantic Web decision support system. News items are provided by news feeds and should be classified at such a pace that stock traders have read the relevant news items earlier than their competitors. At the current moment, based on the rate of news items emerging on the selected RSS feeds, the system starts classifying news items every 5 minutes. After classification, the news items should be personalized to be tailored to the needs of the stock trader. Therefore the performance of the classification process of news items is very important. This does not allow us to use the most sophisticated (and time-consuming) ontology populating techniques, because the main aim is to provide reliable classification in a timely manner. By reliable classification we mean that news items should always be assigned to the most relevant concepts in the ontology.

3.1 The Knowledge Base

In the knowledge base we store all information about the news domain. We chose to store the information in an ontology, not only because the Semantic Web is based on the idea of sharing information in ontologies, but also because it supports easy extensibility of the information and it is well supported in multiple software environments. Ontologies are also easily interchangeable, this enables others to use our knowledge base, or conversely if other people have knowledge bases of another domain, we can benefit from their knowledge to classify news and to populate the news ontology.

To keep the information exchange as simple as possible there are no special restrictions on the knowledge base, as long as it is stored in OWL format. For every concept in the ontology, a synonym property is defined to supply the classification process with lexical representations of a concept. Moreover a WordNet sense property is defined. This property stores URIs (Uniform Resource Identifier) of WordNet [7] senses in order to retrieve more synonyms from WordNet. These URIs refer to an RDF resource of the WordNet OWL/RDF representation. We have chosen to start working with the OWL/RDF WordNet representation that is currently under development by the W3C [23]. WordNet also provides hyponyms for synsets that can be useful in some cases in the classification process. In fact by linking concepts to WordNet senses the domain specific knowledge base has become an extension of the general WordNet lexical ontology.

To demonstrate our approach we have developed a domain specific ontology. We chose to model a knowledge base that can classify news in the NASDAQ domain. More specifically, we selected a subset of twenty companies. Our aim is to collect and classify all news items that are relevant with respect to these companies.

In the framework there is another important ontology besides the knowledge base ontology: the news ontology. The news ontology bridges the gap between the concepts of the knowledge base and the news items itself. It stores the news items together with all relevant information like the associated hyperlink, release time, and source. The classification process links news items to the concepts in the knowledge base and vice versa, this linkage is also stored in the domain ontology. By using the relations between the concepts in the knowledge base and the news items we can provide news personalization. Using these two ontologies, the knowledge base ontology and the news ontology, we are now able to start the classification process and populate the news ontology with news items which are linked to concepts in the knowledge base ontology.

3.2 The Classification Process

The classification process basically consists of two steps: searching for the name or synonyms of the concept as defined in the ontology, and looking up words that denote the concept with the help of WordNet. We will discuss these steps of the process in this section. The classification process is knowledge base centric, which means that a concept is loaded from the knowledge base, then this concept is directly matched against every news item. This is opposed to a news item centric approach, where for every news item every concept in the ontology would be matched. The ontology centric approach was chosen to have better performance in combination with retrieving synonyms and hyponyms from WordNet, which has to be done every time a concept is loaded from the knowledge base. An overview of this process can be found in Figure 1. The concepts and their associates

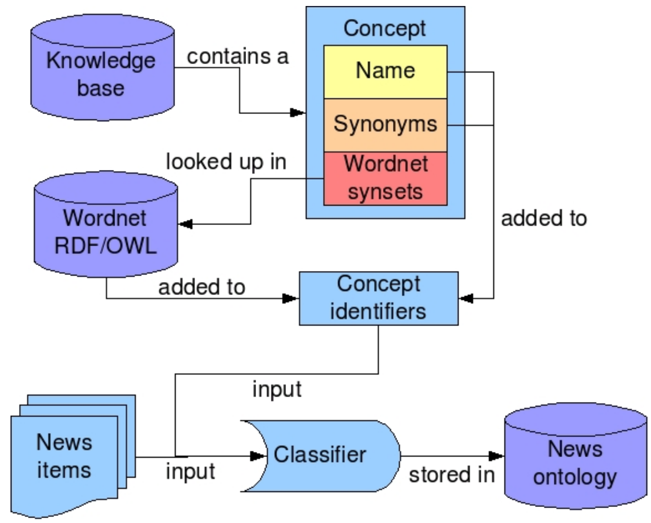

Figure 1: An overview of the classification process.

WordNet synsets are manually added to the knowledge base by the knowledge engineer.

3.2.1 *Concept matching*

Concepts are represented as classes or individuals in the ontology. In each iteration one of the concepts is selected by the classification process. For this concept its name and synonyms are retrieved from the ontology, and its presence in the news items is checked. A relation between a news item and a concept in the knowledge base can be denoted by one or more lexical representations of a concept. For example, "New York" and "Big Apple" can denote the same relation between a news item and a concept (the concept being New York City). These lexical representations are stored as hits, to be able to examine the reason of existence of the relation in the future by users or by domain experts.

3.2.2 *Improving the classification result with WordNet*

We improve our classification process with the help of WordNet. For every concept the process retrieves synonyms and hyponyms from WordNet. Hyponyms are only retrieved when the concept is a class in the ontology that does not have any subclasses and instances, or if the concept is an individual. The intuition for this approach is that when a class does not have any subclasses or instancees in the ontology, the domain expert, who defined the ontology, is not interested in a more detailed classification of the concept. However, if more specific forms of the concept are found it is still useful to classify them as the more general one. If WordNet can provide hyponyms for an individual the same intuition as for classes without children (subclasses or individuals) holds true.

The classification process loads the WordNet senses from the ontology and sends them to the WordNet library, which in turn provides the classification process with lexical representations for this WordNet senses. These lexical representations are then checked for presence in the news items. If an item is present the same process is applied as for the lexical representations stored in the ontology.

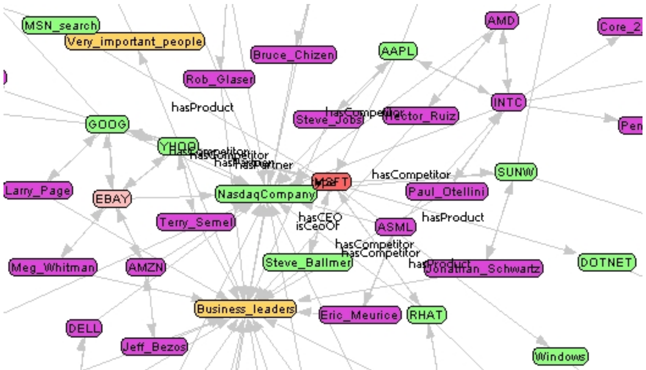

Figure 2: The conceptual graph.

3.2.3 Evaluation

When all concepts from the knowledge base are passed and all their respective lexical representations are looked up, the process starts an evaluation round. In this evaluation round the relations of every news item are assessed. If enough relations are found for a news item to become useful, it is added to the news ontology. Currently our heuristic is that every news item should have at least three relations to concepts in the knowledge base in order to be useful in the decision making process. Having less relations means that a news item is not relevant for the domain of the knowledge base and therefore is not of any interest to the end-user.

4. ONTOLOGY VISUALIZATION

In order to present the knowledge base to the user, we depicted it as a directed graph. We decided to use a graph instead of a tree, because this results in a clear picture of the relations between concepts, even when the knowledge base gets very large. A tree only shows the sub- and superclass relations between nodes, while a graph also shows the other relations between the nodes. The user can use this graph to specify the concepts in which he is interested. Figure 2 shows the conceptual graph.

Each node in our graph represents a concept in the knowledge base, having two properties: a lexical representation of the concept in the ontology and a Uniform Resource Identifier (URI). The lexical representation is displayed on the node, while the URI is used to uniquely identify that particular node. Nodes can have multiple incoming and outgoing edges.

Each edge in our graph represents a relation between two concepts in the knowledge base, is directed, and has a label associated with it. This label represents the type of relation between two nodes. An example of this is the edge between the nodes `MSFT` (which stands for Microsoft) and `Steve Ballmer` with the label `hasCEO`. This edge shows that Microsoft has a CEO named Steve Ballmer; it is an outgoing edge for Microsoft and an incoming edge for Steve Ballmer.

In our graph we use a node coloring scheme to emphasize the node's states and types, which is depicted in Table 1. Looking at Figure 2, this coloring scheme becomes apparent.

4.1 Selecting Concepts

When the user wants to start the information retrieval process, he is presented with an interface such as shown in Figure 3, i.e., a graph which depicts all the concepts in the knowledge base. By using this graph, the user is able to select concepts about which he wants to see news, we call these concepts the "concepts of interest".

Table 1: Node coloring scheme.

Color	State
Yellow	Nodes of type OWL individual
Purple	Nodes of type OWL class
Red	Selected nodes
Green	Nodes directly related to the selected node
Pink	Nodes in the keyword search result

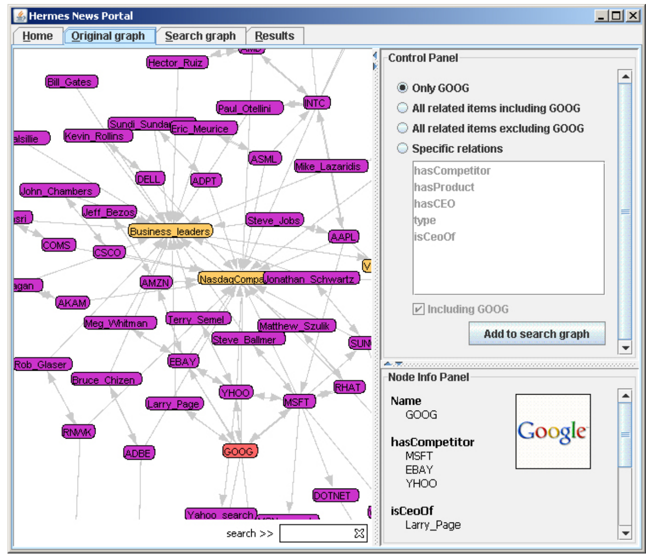

Figure 3: An overview of the application showing the knowledge base.

By selecting a concept in the graph the control panel in the application is activated, thereby enabling the user to add the selected concept (and optionally its related concepts) to the concepts of interest. These concepts of interest are the concepts to which the news items must be related. The control panel provides the user with the ability to not only add the selected node to the concepts of interest, but also nodes directly related to it. A user might for example want to add the node `GOOG` (which stands for Google) and its competitors to the concepts of interest.

4.2 Time Constraints

Time constraints are useful, because the degree to which news items are relevant varies across time. In some cases only recent information is relevant, for example when trying to determine what recent news influenced the price of a stock. In other cases a historical overview might be more appropriate, for example in the case where analysts want to analyze news from the past quarter to be able to say something about an upcoming quarterly result announcement. Our framework allows the user to impose constraints on the time period in which the extracted news items emerged. There can be constraints on either the time or the date of the news items.

Table 2: An example of a default time constraint query

```
PREFIX hermes: <http://hermes-news.org/news.owl#>
SELECT ?title
WHERE {
   ?news hermes:title ?title .
   ?news hermes:time ?date .
   ?news hermes:relation ?relation .
   ?relation hermes:relatedTo hermes:Google .
   FILTER
   (
      ?date > "2007-03-01T00:00:00.000+00:01" &&
      ?date < "2007-05-31T00:00:00.000+00:01"
   )
}
```

Table 3: Custom time functions

Function	Output
currentDate()	xsd:date
currentTime()	xsd:time
now()	xsd:dateTime
dateTime-add(A, B)	xsd:dateTime
dateTime-subtract(A, B)	xsd:dateTime

5. RESULT EXTRACTION AND VISUALIZATION

In order to retrieve the news items that are related to the concepts of interest specified by the user, we need a query language which is capable of extracting data from an ontology. We make use of SPARQL [18], which is a query language capable of extracting data from ontologies, based on path expressions and filters. However, SPARQL does not support advanced time functions, so we extended SPARQL with the more advanced time functions we need in order to apply time and date constraints to the news items.

5.0.1 SPARQL Time Extension

Users are mostly interested in recent news, so time constraints are an important part of the result extraction procedure. By making use of a `FILTER` tag, SPARQL is able to restrict the result set based on some conditions, which in our case are usually time constraints. An example of a time constraint search query is:

"Which are the news items about Google from the past three months?"

If this question is asked on the 31st of May 2007, then the SPARQL equivalent of this question is depicted in Table 2.

Although using a filter with a hard coded date can help in this case, it is more convenient to use predefined functions like `currentDate()` and `currentTime()`. It might also be useful to perform arithmetic with times and dates to create more expressive time and date filters. Table 3 shows the custom time functions we have added to SPARQL.

Using these custom time functions makes it easier to express the previously discussed search query in SPARQL. Because some time functions provided by our extension are relative, it is no longer needed to hard code the time constraint in the appropriate `xsd:dateTime` format. Instead it is possible to create an expression by combining the custom time functions, which enable more powerful time constraints directly into SPARQL. In Table 4, the same query as in Table 2 is depicted, only now the time constraint in the `FILTER` is constructed using our custom functions. The first argument (A) for the `dateTime-add` and `dateTime-substract` functions is of type `xsd:dateTime`. The second argument (B) of these functions is of type `xsd:duration`.

Table 4: An example of a time-constraint query, using custom functions

```
PREFIX hermes: <http://hermes-news.org/news.owl#>
SELECT ?title
WHERE {
   ?news hermes:title ?title .
   ?news hermes:time ?date .
   ?news hermes:relation ?relation .
   ?relation hermes:relatedTo hermes:Google .
   FILTER
   (
      ?date > hermes:dateTime-substract(hermes:now(), P0Y3M) &&
      ?date < hermes:now()
   )
}
```

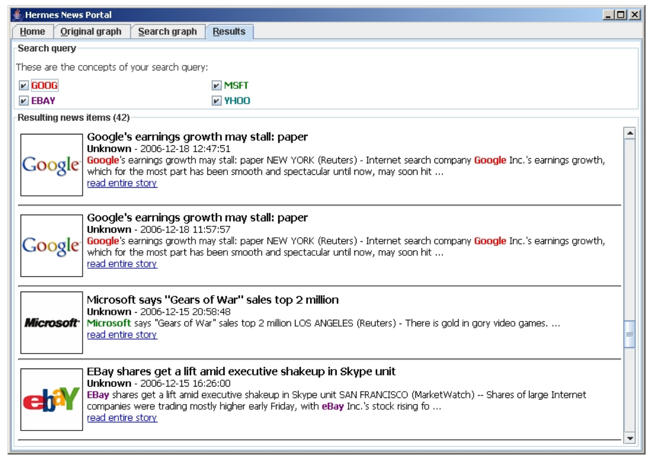

Figure 4: A visualization of various news items.

5.0.2 News Item Extraction

The ultimate goal of our software is to provide the user with the news items that are related to the concepts of interest. Once the user has selected all the concepts which are of interest to him, the relevant news items are to be extracted. This is done by the Result Extractor which generates a SPARQL query based on the concepts the user is searching for, and executes this query.

The result is shown to the user as a summary of all the news items which are related to the selected concepts. An example of this summary can be found in Figure 4. For each news item the title, the source, and the date are shown together with the first few lines of the item. However many news agencies do not encode the source in their RSS feeds.

6. CONCLUSION

The Hermes news portal is a decision support system for domains that are highly dependent on news from several sources. Furthermore it is bringing the benefits of semantic relationships to a decision support system. As part of the framework we introduced ontology-based news classification, which we extended with synonyms and hypernyms that are retrieved from WordNet. Ontology-based personalization allows the decision maker to make use of relations between concepts when searching for evidence supporting or rejecting a decision, based on a pool of recent and older news items. As users are able to limit the search space by time constraints, they are able to retrieve recent information from a recent news item, or instead retrieve a historic overview, about the evolution of the domain over time.

By using Semantic Web techniques the different sources of news items can easily be integrated, and the relations between concepts in a knowledge base and news items can be represented. The ontology facilitates semantic-based information retrieval for decision making purposes. The emergence of the Semantic Web provides new opportunities for decision support systems. We have demonstrated its usefulness in the stock market domain where information in news items heavily supports decision making.

In the future we plan to extend our framework with semi-automatic ontology learning components in order to keep the knowledge base up-to-date. Furthermore a front-end improvement would be to provide context graphs for news items. Such a graph depicts the news item in the context of the knowledge base by visualizing all news item relations to concepts in this knowledge base. We also plan to evaluate the tool with real users to further improve our approach.

7. ACKNOWLEDGMENTS

The authors are supported by the EU funded IST-STREP Project FP6-26896: *Time-determined ontology-based information system for realtime stock market analysis* (TOWL). More information is available on the official website[2] of the TOWL project.

8. REFERENCES

[1] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. Owl web ontology language reference. W3C Recommendation 10 February 2004, 2004. `http://www.w3.org/TR/owl-ref/`.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[3] A. Bosca, D. Bonino, and P. Pellegrino. Ontosphere: More than a 3d ontology visualization tool. In *The 2nd Italian Semantic Web Workshop (SWAP 2005)*, 2005.

[4] D. Brickley and R. V. Guha. Rdf vocabulary description language 1.0: Rdf schema. W3C Recommendation 10 February 2004, 2004. `http://www.w3.org/TR/rdf-schema/`.

[5] S. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization*. Morgan Kaufmann, 1999.

[6] H. Cunningham. Gate, a general architecture for text engineering. *Journal Computers and the Humanities*, 36(2):223–254, 2002.

[7] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[8] F. Frasincar, A. Telea, and G. Houben. *Adapting Graph Visualization Techniques for the Visualization of RDF Data*, pages 154–171. In [9], 2006.

[2]`http://www.towl.org`

[9] V. Geroimenko and C. Chen. *Visualizing the Semantic Web, Second Edition*. Springer, 2006.

[10] T. R. Gruber. A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220, 1993.

[11] A. Java, T. Finin, and S. Nirenburg. Semnews: a semantic news framework. In *Twenty-First National Conference on Artificial Intelligence (AAAI 2006)*, pages 1939–1940. American Association of Artificial Intelligence, 2006.

[12] A. Java, T. Finin, and S. Nirenburg. Text understanding agents and the semantic web. In *39th Hawaii International Conference on System Sciences (HICSS 2006)*. IEEE Computer Society, 2006.

[13] Y. Kalfoglou, J. Domingue, E. Motta, M. Vargas-Vera, and S. B. Shum. myplanet: An ontology-driven web-based personalised news service. In *Workshop on Ontologies and Information Sharing (IJCAI 2001)*, 2001.

[14] E. Kandel and L. Marx. Nasdaq market structure and spreads patterns. *Journal of Financial Economics*, 45(1):61–89, 1997.

[15] G. Klyne and J. J. Carroll. Resource description framework (rdf): Concepts and abstract syntax. W3C Recommendation 10 February 2004, 2004. `http://www.w3.org/TR/rdf-concepts/`.

[16] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with protege-2000. *IEEE Intelligent Systems*, 16(2):60–71, 2001.

[17] E. Pietriga. Isaviz, a visual environment for browsing and authoring rdf models. In *The Eleventh International World Wide Web Conference (WWW 2002)*, 2002. Developer's Day.

[18] E. Prud'hommeaux and A. Seaborne. Sparql query language for rdf. W3C Candidate Recommendation 14 June 2007, 2007. `http://www.w3.org/TR/rdf-sparql-query/`.

[19] M. Sintek. Ontoviz tab: Visualizing protégé ontologies. Available online from `http://protege.stanford.edu/plugins/ontoviz/ontoviz.html`, 2004.

[20] M. Storey, M. Musen, J. Silva, C. Best, N. Ernst, R. Fergerson, and N. Noy. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in protégé. In *Workshop on Interactive Tools for Knowledge Capture (K-CAP-2001)*, 2001.

[21] C. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical systems. *IEEE Transactions of System, Man, and Cybernetics*, 11(2):109–125, 1981.

[22] A. Telea, A. Maccari, and C. Riva. An open toolkit for prototyping reverse engineering visualization. In *Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym 2002)*, pages 241–250. Eurographics, 2002.

[23] M. van Assem, A. Gangemi, and G. Schreiber. Rdf/owl representation of wordnet. *W3C Working Draft 19 June 2006*, 2006. `http://www.w3.org/TR/wordnet-rdf/`.

[24] D. Winer. Rss 2.0 specification. Berkman Center for Internet & Society at Harvard Law School, 2003. `http://cyber.law.harvard.edu/rss'/rss.html`.