# Hermes: an Ontology-Based News Personalization Portal

Borsje, Jethro and Levering, Leonard and Embregts, Hanno and Frasincar, Flavius

Erasmus University Rotterdam

10 January 2007

# Hermes: an Ontology-Based News Personalization Portal

Jethro Borsje, Leonard Levering, Hanno Embregts, and Flavius Frasincar
{j.borsje,l.levering,h.embregts}@student.eur.nl, frasincar@few.eur.nl

Erasmus University Rotterdam
PO Box 1738
NL-3000 DR Rotterdam
The Netherlands

**Abstract**

Nowadays, news feeds provide Web users with access to an unlimited amount of news items, however only a subset of them is relevant. Therefore, users should be able to select the most relevant concepts, about which they want to retrieve news. Although keyword search engines provide users with the ability to filter news items, they lack the power of understanding the domain where the news items reside.
The aim of this paper is to propose a solution that provides users with the ability to ask for news items related to specific concepts they are interested in. This is accomplished by creating an ontology, developing a classifying system that populates the ontology by making use of a knowledge base, and providing an innovative graph representation of the ontology to retrieve relevant news items. A characteristic feature of our approach is the consideration of both concepts and concept relationships for the retrieval of user-relevant items.

## Introduction

The rapidly evolving techniques that have enabled the Internet to get heavily used for various purposes are nowadays key factors for the online publishing of news messages. At this moment, most people have access to the Internet, which is why the Internet has become an important platform for spreading these news messages around the world. Numerous new technologies have emerged since the introduction of the Semantic Web [1], which provide a perfect fit when confronted with the issue of assigning clear semantics to news messages.

In order to provide a structured framework for searching news messages on the Internet, we propose a solution (called Hermes) that is based on various 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. The technologies that we used include the data description languages RDF (Resource Description Framework) [2, 3], OWL (Web Ontology Language) [4], and the corresponding query language SPARQL (SPARQL Protocol And Query Language) [5]. Our approach is based on a conceptual model for storing concepts, news items, and relations between these concepts and news items; a model that came into being after an extensive researching phase. In this paper, we present our findings regarding this research and we show an implementation of our solution, with special focus on the NASDAQ domain.

In the news classification and ontology visualization fields a lot of research has already been done, of which we now give a brief introduction. In the field of news classification, the SemNews [6] application strives to extract meaningful information out of news items, which appear on the Internet as RSS feeds. The news items are analyzed by OntoSem [7], SemNews's underlying natural language processing engine, and are classified with respect to their proper meaning and then stored together with its 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.

The Artequakt [8] application is meant to extract knowledge from web pages, to populate a knowledge base, and to use that to generate personalized biographies. Its purpose is to discover how the different names, words, and terms are related, after which the results are exported to an ontology. A central ontology server stores the different ontologies with respect to the documents where the ontology data were originally derived from. On top of that, an inference engine is present to search through the ontologies.

In the past few years visualization of ontologies has become a popular research field. Protégé [9], for example, is an ontology editor that supports hierarchical views. By making use of plugins it is possible to obtain different views of the ontology. An example of one of these plugins is OntoSphere [10], which displays ontologies in a three-dimensional space. Another plugin is Jambalaya [11], which uses an hierarchical multi-perspective visualization technique to show the ontology.

IsaViz [12] 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 layout algorithm which determines the location of the elements. The layout as produced by IsaViz looks quite clear, although the usage of large data sets has a negative effect on the quality of the overview, as the excess of data requires nodes and edges to be displayed in smaller proportions.

### Our Approach

The goal of this project is to provide the user with an understandable news portal, which is capable of letting the user search for news items in his areas of interest by selecting concepts from a knowledge base. The pool of news items is automatically gathered by using input from various RSS feeds. By using a pre-defined knowledge base, the news items can be classified with respect to their relations with concepts in this knowledge base.

In contrast to work that has been done earlier, this approach focuses both on news classification and visualizing ontologies. By combining these two research fields, it is possible to create an understandable news portal, in which the user can choose his own interests and in which the results are displayed in a way that the user can understand. The project also clearly demonstrates the usefulness of the Semantic Web and its techniques: it enables computers to 'understand' the news items on the Internet. This provides the user with the ability to specify searches with respect to the context of the news items. Furthermore, it contributes to the amount of content that is available online in Semantic Web format.

The paper is organized as follows. In the next section we discuss the news domain we have chosen and how the information about this domain is represented. In the last part of that section we demonstrate three example search queries that will be supported by our application. The section thereafter further explains the news classification process, while the next section discusses the user interface of our software program and how the user can interact with the system. Then we describe the result extraction in more detail, after which the next section will introduce the visualization of the result that has been extracted earlier. To conclude the paper, the last section will summarize our findings, it will present our conclusions, and it will identify the areas for future research.

## Domain Analysis

### Choosing The Domain

To be able to classify the various news items, a knowledge base should be built based on a certain news domain. As much as we would like to build a knowledge base about all concepts that happen to be in the world news, this idea is not feasible in practice. This is due to the fact that there are simply too many concepts to take into account, which is why we need to limit the domain. Another reason to limit the domain can be found in the visualization of the ontology. If the ontology would describe all concepts that are related to the world news, visualizing that ontology would be very cumbersome.

Based on the reasons that are stated above, we have chosen a specific domain for which we want to classify the news, which is the NASDAQ stock market. People who hold shares in the NASDAQ index will be very interested in news items that relate to their shares portfolio, because of the high impact that a certain news item can have on the share prices. To keep things simple, we will focus on a subset of NASDAQ funds in our research, thereby using funds that are directly computer-related, but are part of different sectors. The selected funds are listed in Table 1.

| **Company** | **Symbol** |
|---|---|
| Apple | `AAPL` |
| Adobe | `ADBE` |
| Adaptec | `ADPT` |
| Akamai | `AKAM` |
| Advanced Micro Devices | `AMD` |
| Amazon | `AMZN` |
| ASML | `ASML` |
| 3Com | `COMS` |
| Cisco | `CSCO` |
| Dell | `DELL` |
| E-bay | `EBAY` |
| Google | `GOOG` |
| Intel | `INTC` |
| Microsoft | `MSFT` |
| Red Hat | `RHAT` |
| Research in Motion | `RIMM` |
| Real Networks | `RNWK` |
| Sun Microsystems | `SUNW` |
| Yahoo | `YHOO` |

Table 1: Selected group of NASDAQ funds.

## The News Ontology

All information about our defined NASDAQ news domain is stored in an *ontology*. The ontology definition we use is: "a formal, explicit specification of a shared conceptualization" [13]. This definition can be explained as: "A conceptualization, in this context, refers to an abstract model of how people think about things in the world, usually restricted to a particular subject area" [14]. We express our ontology using the Web Ontology Language (OWL), in which knowledge is stored in *classes*, *individuals*, *properties*, and *literals*.

Basically the information stored in the ontology can be divided into three different categories: the *knowledge base* containing the *concepts* which are relevant for the news domain that is to be analyzed, the *news items* that are classified, and the *relations* between the news items and the concepts in the knowledge base. These three categories all have their own class under the root class. The first class we discuss is the `News`-class, of which instances store the classified news items. The `News`-class has a few properties which are used to store the title and the text of the news item, a link to the news item, and the time at which the news item appeared. Moreover, we have one property that stores instances of the `Relation`-class.

The second class provided under the root is the `Relation`-class, of which instances store the relation between a news item and a concept in the knowledge base. It has the property `hit`, which can store multiple hits, a hit being the word in the news item that triggers the classifier to relate this particular news item to a concept in the knowledge base. Besides the concept names themselves being checked for appearance, their synonyms are also taken into account. The other property of the `Relation`-class is called `timesFound`, which indicates how many times certain concepts of the knowledge base have been found in a news item. This property makes it possible to sort news items based on their relevance. The `Relation`-class has one object property called `relatedTo`, which contains a reference to the concept the news item is related to.

The knowledge base is stored under the `Concept`-class in the ontology, which contains information related to the NASDAQ funds we want to track. Moreover, the knowledge base has to include general information, for example about different countries and regions in the world and the most important terms used in financial news messages.

Figure 1 depicts the structure of the ontology with an example news item.

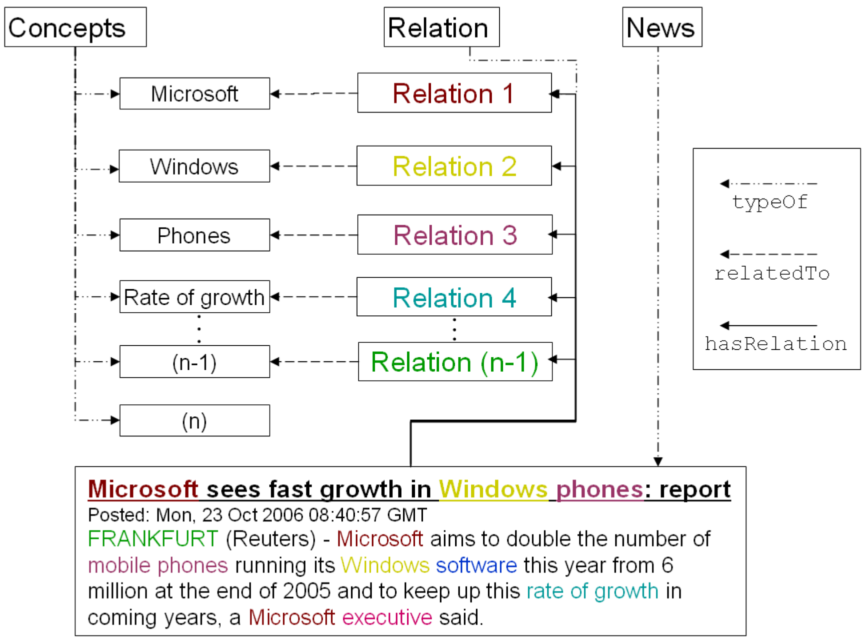

Concepts
Relation
News
Microsoft
Windows
Phones
Rate of growth
(n-1)
(n)
Relation 1
Relation 2
Relation 3
Relation 4
Relation (n-1)
typeOf
relatedTo
hasRelation
Microsoft sees fast growth in Windows phones: report
Posted: Mon, 23 Oct 2006 08:40:57 GMT
FRANKFURT (Reuters) - Microsoft aims to double the number of mobile phones running its Windows software this year from 6 million at the end of 2005 and to keep up this rate of growth in coming years, a Microsoft executive said.

Figure 1: An example of a news item in the ontology.

### Some Example Search Queries

As previously discussed, the Semantic Web is about understanding the semantics and context of information. Once this information is properly classified, it is possible to impose structured queries on it to extract relevant information. In our domain, users are interested in information about the companies of which they possess some shares. For example, a user who is particularly interested in the company Microsoft, would probably want to formulate the following search query:

"News about Microsoft."

Although this might be useful, this search can also be done by a simple keyword search. It does not really show the usefulness of semantics because `Microsoft` is just a single concept. The following query is a better example of the usages of semantics:

"News about Microsoft and its competitors."

The execution of this search query would result in news items about the company Microsoft and all of its competitor companies, without having to know which these competing companies are. In our implementation, the user can create such a query by selecting the concept `Microsoft` and the concepts that are related to it by the relation `hasCompetitor`.

Besides constraints on specific concepts, constraints on the appearance time of the news item might also be useful. In the case of the stock market, analysts are often interested in news items originating from a particular time frame. For example, when trying to predict the impact of the quarterly result message from a company, an analyst might be interested in the news items of the last quarter. The following search query might reflect what the analyst is interested in:

"News about Microsoft and its competitors from the past quarter."

## Classifying News

At this moment we have an ontology, containing the knowledge base about the NASDAQ news domain, a class that represents news items and the possibility to store relations between the news and the concepts in the knowledge base. We will now discuss how news items are classified, based on the concepts in the knowledge base.

The classifier takes the addresses of various news feeds as input. When the classifier starts, all the news items are aggregated and stored in an object by making use of the Informa library [15] for Java. After all news items are loaded, the classifier loads the ontology by making use of the Jena library [16], which is responsible for loading, editing and writing the ontology throughout Hermes. When the ontology is loaded, the process of classification starts.

The process of classification works ontology-centric, which means that the classification recurses the knowledge base in the ontology and checks every concept for a related news item. The classifier loads the `Concept`-class, which is the root of the knowledge base, and then recursively walks through the ontology. The name of every concept that is read from the ontology is checked for appearance in any of the news items. If this is the case, a relation between the concept and the news item is added.

The lexical representation of a concept name is not unique: there are often many other words that can be used to identify the same concept. For example, the concept `George W. Bush` could also be referred to as 'President of the United States' or simply as 'President'. To attend to this issue, a synonym can be provided for each concept in the ontology by making use of the synonym property. The classifier checks the synonyms in the same way as it checks the concept names.

When defining an ontology, it is not always possible to conjure up all possible synonyms. The Princeton University provides a solution to this problem with their WordNet project [17], which provides an online dictionary that also includes synonyms. The WNWA project [18] even supplies a web service interface to the WordNet dictionary. The name of every concept is sent to the WNWA web service, after which the service returns an XML file in which the synonyms reside (amongst some other information about that particular word). These synonyms are compared to the text of the news item and if one of these words is found, the relation is stored.

After all concepts are processed, all news item objects contain a complete list of relations between the knowledge base and themselves. The classifier walks through all these news items and checks the number of relationships that are found. If more than two relationships have been found, the classifier adds the news item to the ontology together with the relationships that were found. The classifier only adds news items with more than two related concepts, which is an aspect that can be regarded as part of our heuristics.

Classification can be done quite successfully in terms of news item recognition. If the news feeds and the knowledge base in the ontology cover the same news domain, the classifier is able to classify more than 95% of the news items. The speed of the classification process depends on several factors, but when using a computer with an Intel Pentium M processor (1,3 Ghz) and 512 Mb of memory, the classifier can process 10 news items within 30 seconds and up to 500 news items within 300 seconds. This figure shows that it is possible to process many news items in a relatively short time, thereby enabling us to regularly update the ontology with new news items. An overview of this classification process is shown in Figure 2.

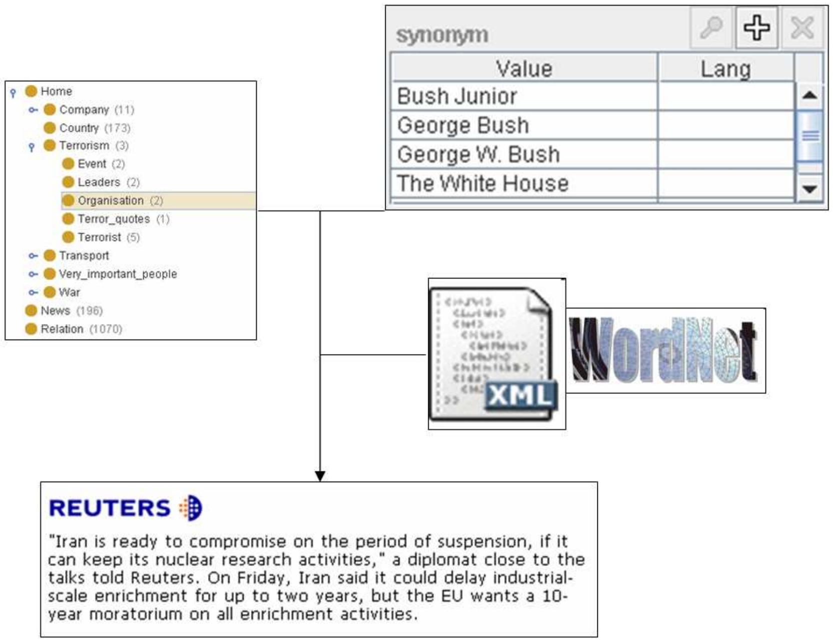
## User Interface

### Visualizing The Ontology

In order to present the knowledge base to the user, we depicted it as a directed graph. We decided to use a graph, because this results in a clear picture of the relations between concepts, even when the knowledge base gets very big. The user can use this graph to specify the concepts in which he is interested (see Figure 3 for an example of this). We use the Prefuse [19] library for the graph layout package for Java.

For the visualization we used a 'spring embedder' layout algorithm, which models a graph as a set of nodes connected by edges. The graph is placed in an arbitrary initial state, after which the edges move the nodes toward some steady state. This moving is done based on the intensity of the relations between nodes: the more relations there are between nodes, the closer they are placed together because they attract each other. In contrast, nodes which have no relations with one another repulse each other.

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.



| Value | Lang |
|---|---|
| Bush Junior | |
| George Bush | |
| George W. Bush | |
| The White House | |

"Iran is ready to compromise on the period of suspension, if it can keep its nuclear research activities," a diplomat close to the talks told Reuters. On Friday, Iran said it could delay industrial-scale enrichment for up to two years, but the EU wants a 10-year moratorium on all enrichment activities.

Figure 2: An overview of the classifier.

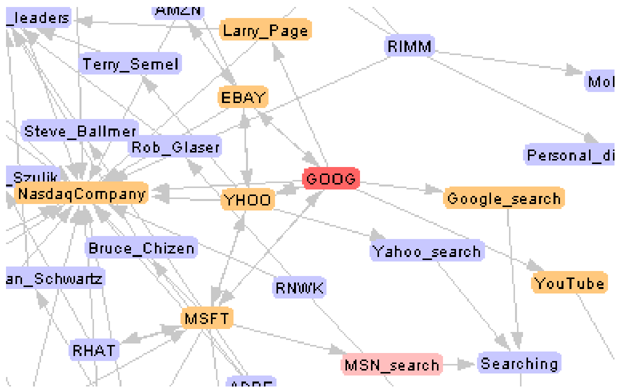


Figure 3: The graph.

Each edge in our graph represents a relation between two concepts in the knowledge base and each edge 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 `GOOG` (which stands for Google) and `Larry Page` with the label `hasCEO`. This edge shows that Google has a CEO named Larry Page; it is an outgoing edge for Google and an incoming edge for Larry Page.

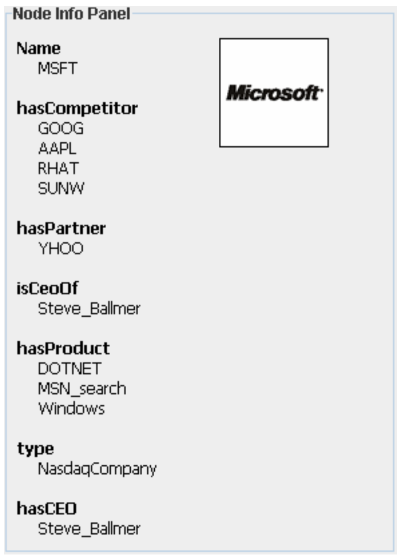
In our graph we use a node coloring scheme to emphasize the node's states, which is depicted in Table 2. Looking at Figure 3, this coloring scheme becomes clear. In this figure the node `GOOG` is selected so it is colored red, while the node `Larry Page` is one of the related nodes, which is why it is colored orange. At the same time a keyword search for the string `MSN` has been executed. The node `MSN_search` is in the result set of this keyword search, so it is colored pink. All the other nodes are blue-colored, thereby indicating that those nodes have no special meaning.

| Color | State |
|---|---|
| Blue | Regular nodes. |
| Red | Selected nodes. |
| Orange | Nodes directly related to the selected node. |
| Pink | Nodes in the keyword search result. |

Table 2: Node coloring scheme.

### User Assistance

Because the graph can become quite cluttered, we constructed a node info panel (see Figure 4) to provide some contextual information about the currently selected node. It shows all the incoming and outgoing relations of the node and the nodes to which these relations connect, which can be useful for gaining insight in the context of the selected node.



Node Info Panel

**Name**
MSFT

**hasCompetitor**
GOOG
AAPL
RHAT
SUNW

**hasPartner**
YHOO

**isCeoOf**
Steve_Ballmer

**hasProduct**
DOTNET
MSN_search
Windows

**type**
NasdaqCompany

**hasCEO**
Steve_Ballmer

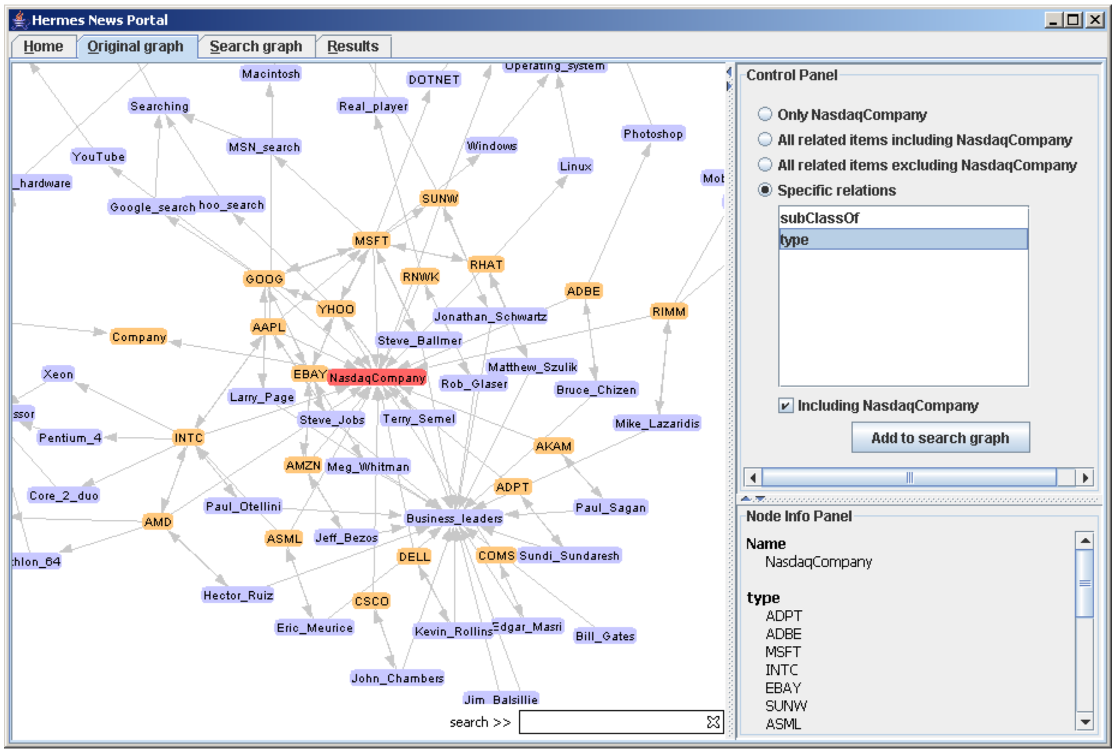
Figure 4: The node info panel.

The graph which presents the knowledge base can get quite big, so it can be a hard job to find out if a certain node is present in the graph. To make this easier, we have implemented a keyword search interface, of which a screen shot can be found in Figure 5. While the user types his search term, a search is automatically started based on the names of the nodes. After each character the user types, the result set is updated and the nodes that are in it are colored pink.

Figure 5: The search box.

### Selecting Concepts

When the user wants to start the concept selection process, he gets a screen like the one as shown in Figure 6, presenting 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 will now discuss how this selection procedure is done.



Control Panel

- Only NasdaqCompany
- All related items including NasdaqCompany
- All related items excluding NasdaqCompany
- [✓ Specific relations]

subClassOf
[✓ type]

[✓ Including NasdaqCompany]

Add to search graph

Node Info Panel

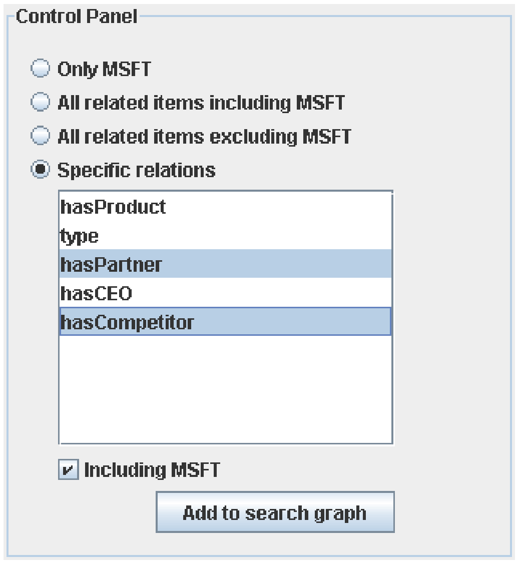
Name
NasdaqCompany

type
ADPT
ADBE
MSFT
INTC
EBAY
SUNW
ASML

Figure 6: 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 are the concepts to which the news messages must be related.

Figure 7 shows the control panel when the node MSFT (which stands for Microsoft) is selected. The panel offers the user a few options related to the selected concept. The default behaviour is that only the selected node is added to the concepts of interest.


Control Panel

- Only MSFT
- All related items including MSFT
- All related items excluding MSFT
- [✓ Specific relations]

hasProduct
type
[✓ hasPartner]
hasCEO
[✓ hasCompetitor]

[✓ Including MSFT]

Add to search graph

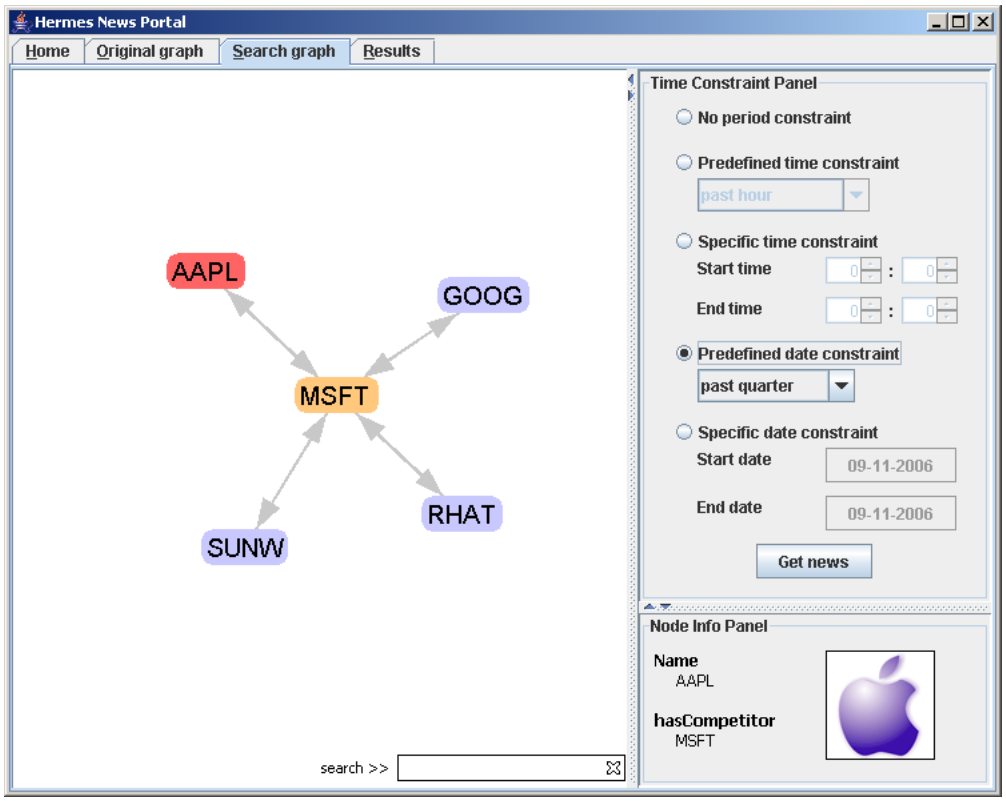
Figure 7: The control panel.

As can be seen in Figure 7, there are three more options concerned with adding concepts to the concepts of interest. The first additional option allows the user to add all the related items, including `MSFT` to the concepts of interest. Using this option, the user adds `MSFT` and all the nodes that are directly linked to it to the concepts of interest. The second additional option comprises the same as the first one, with the exception that the selected node itself (in this case `MSFT`) is not added to the concepts of interest.

The third option provides the most functionality, because it allows the user to specify which related concepts should be added. By specifying the names of the relations to add to the concepts of interest, only the nodes which fulfill these relations with respect to the clicked node are added. Using this option, it is possible to specify if the selected node should also be added to the concepts of interest. The user might, for example, only be interested in news items about the competitors of Microsoft. In such a case, the user specifies that only concepts related to Microsoft with the relation `hasCompetitor`, excluding the concept Microsoft itself should be added to the concepts of interest.

## Time Constraints

The selected concepts of interest of the user are displayed in another graph, which is accessible through the 'Search graph' tab, shown in Figure 8. Besides containing a graph with the selected concepts, this tab also contains a 'Node Info Panel' (similar to the one we previously discussed) and a 'Time Constraint Panel'.



Time Constraint Panel

No period constraint

Predefined time constraint: past hour

Specific time constraint
Start time 0 : 0
End time 0 : 0

[✓ Predefined date constraint]: past quarter

Specific date constraint
Start date 09-11-2006
End date 09-11-2006

Node Info Panel

Name
AAPL

hasCompetitor
MSFT

Figure 8: An overview of the application showing the selected concepts of interest.

The 'Time Constraint Panel', as shown on the right side in Figure 8, allows the user to impose constraints on the time period in which the extracted news items appeared. There can be constraints on either the time or the date of the news items. An example of a time constraint is "news of the past four hours" or "news from 09:00 until 14:00", which are time constraints that apply to the past 24 hours. There are a few predefined time constraints, like 'past hour', 'past six hours' and so on. Besides that, the user can customize his own specific time constraints to suit his own tastes.

It is also possible to specify constraints on the dates of the news. Again there are some predefined constraints, like ‘past day’, ‘past month’, ‘past quarter’, and so on. The user can also specify his own date constraints, as shown in Figure 9, by making use of a calendar which assists the user in picking the dates.

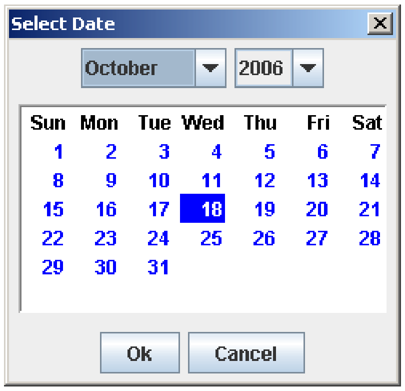


Figure 9: The date panel.

## Result extraction

In order to retrieve news that is 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, which is a query language capable of extracting data from ontologies, based on path expressions and filters. SPARQL is implemented in Java using ARQ [20], a SPARQL processor for Jena. However, SPARQL does not support advanced time functions, so we wrote a time extension for SPARQL which contains the more advanced time functions we need in order to apply time and date constraints to the news items.

### 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:

“News about Microsoft from the past quarter.”

If this question is asked on the 27th of October 2006, then the SPARQL equivalent of this question can be found in Table 3.

```
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:Microsoft .
   FILTER(?date > "2006-07-27T22:39:36.000+00:01")
}
```

**Table 3:** An example of a default time-constraint query.

Although using a filter with a hard coded date can help in this case, it is more convenient to use functions like `currentDate()` and `currentTime()`. It might also be useful to perform arithmetic with

| **Function** | **Output** | **Explanation** |
|---|---|---|
| currentDate() | `xsd:date` | Returns the current date. |
| currentTime() | `xsd:time` | Returns the current time. |
| now() | `xsd:dateTime` | Returns the current date and time. |
| dateTime-add(A, B) | `xsd:dateTime` | Returns date A incremented with interval B. |
| dateTime-subtract(A, B) | `xsd:dateTime` | Returns date A decremented with interval B. |

Table 4: Custom time functions.

| `Unit` **value** | `Expression` **format** |
|---|---|
| MICROSECOND | Number of microseconds. |
| SECOND | Number of seconds. |
| MINUTE | Number of minutes. |
| HOUR | Number of hours. |
| DAY | Number of days. |
| WEEK | Number of weeks. |
| MONTH | Number of months. |
| QUARTER | Number of quarters. |
| YEAR | Number of years. |
| HOUR_SECOND | hours:minutes:seconds |
| HOUR_MINUTE | hours:minutes |

Table 5: The interval `unit` values and `expression` formats.

times and dates to create powerful time and date filters. Table 4 shows the custom time functions we have created for SPARQL.

Both the `dateTime-add` and `dateTime-substract` functions need two arguments, A and B. The first argument (A) is of type `xsd:dateTime`, while the second argument (B) is of a type we defined ourselves: `hermes:dateTimeInterval`. A `hermes:dateTimeInterval` consists of a `unit` specifier with a matching `expression` argument. Table 5 shows the expected form of the `expression` argument for each valid `unit` value.

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 6, the same query as in Table 3 is depicted, only now the time constraint in the `FILTER` is constructed using the 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:Microsoft .
   FILTER(?date > hermes:dateTime-substract(hermes:now(), "1 QUARTER"))
}
```

**Table 6:** An example of a time-constraint query, using custom functions.

## News Item Extraction

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. This query is then passed to ARQ, the SPARQL processor for Jena. ARQ returns a result set which is used by our application to visualize the output by displaying the relevant news items to the user. The visualization of the result is outlined in the next chapter.

As previously discussed, an example of a search query might be:

"News about Microsoft and its competitors from the past quarter."

The graph in Figure 8 shows the relevant concepts with respect to the question stated above. This graph shows that `Microsoft` has four concepts which are related to it through the relation `hasCompetitor`: `AAPL`, `GOOG`, `RHAT`, and `SUNW`. Based on this information, the SPARQL query which is shown in Table 7 can be formulated, which provides the relevant news items. For each concept of interest, a *triple expression* with the `relatedTo` relation is constructed. A `UNION` is performed to make sure that news items related to any of the relevant concepts are selected.

```
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:MSFT .
  }
  UNION {
    ?relation hermes:relatedTo hermes:AAPL .
  }
  UNION {
    ?relation hermes:relatedTo hermes:GOOG .
  }
  UNION {
    ?relation hermes:relatedTo hermes:RHAT .
  }
  UNION {
    ?relation hermes:relatedTo hermes:SUNW .
  }
  FILTER(?date > hermes:dateTime-substract(hermes:now(), "1 QUARTER"))
}
```

**Table 7:** An example of an extraction query, using custom functions.

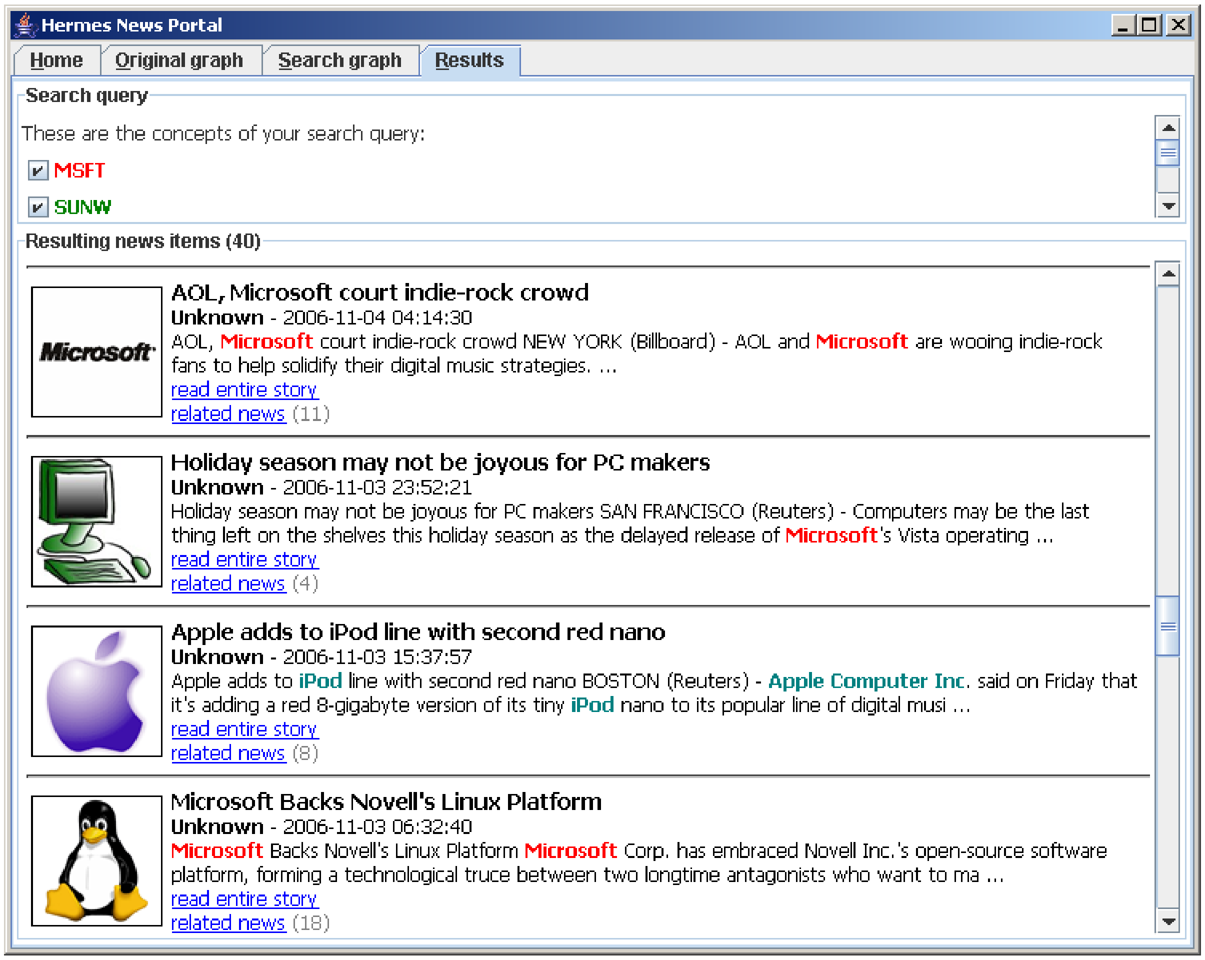
## Result visualization

### Selected News

The ultimate goal of Hermes is to provide the user with news that is related to the concepts that the user has specified. In order to do so, these news items must be logically represented. The result is shown to the user as a summary of all the news items which are related to the selected concepts, of which an example can be found in Figure 10. For each news item the title, the source, and the date are shown together with the first few lines of the item. If the user is interested in the entire news item, he can select the 'Read entire story' link.

### Related News

Because we make use of an ontology to store the concepts, the news items and the relations between these concepts and news items, it is quite easy to relate news items to one another. News items are



These are the concepts of your search query:

✓ MSFT

✓ SUNW

**AOL, Microsoft court indie-rock crowd**
**Unknown** - 2006-11-04 04:14:30
AOL, Microsoft court indie-rock crowd NEW YORK (Billboard) - AOL and Microsoft are wooing indie-rock fans to help solidify their digital music strategies. ...
read entire story
related news (11)

**Holiday season may not be joyous for PC makers**
**Unknown** - 2006-11-03 23:52:21
Holiday season may not be joyous for PC makers SAN FRANCISCO (Reuters) - Computers may be the last thing left on the shelves this holiday season as the delayed release of Microsoft's Vista operating ...
read entire story
related news (4)

**Apple adds to iPod line with second red nano**
**Unknown** - 2006-11-03 15:37:57
Apple adds to iPod line with second red nano BOSTON (Reuters) - Apple Computer Inc. said on Friday that it's adding a red 8-gigabyte version of its tiny iPod nano to its popular line of digital musi ...
read entire story
related news (8)

**Microsoft Backs Novell's Linux Platform**
**Unknown** - 2006-11-03 06:32:40
Microsoft Backs Novell's Linux Platform Microsoft Corp. has embraced Novell Inc.'s open-source software platform, forming a technological truce between two longtime antagonists who want to ma ...
read entire story
related news (18)

Figure 10: A visualization of various news items.

related to each other if their classification was triggered by the same concepts, which means that they share concepts.

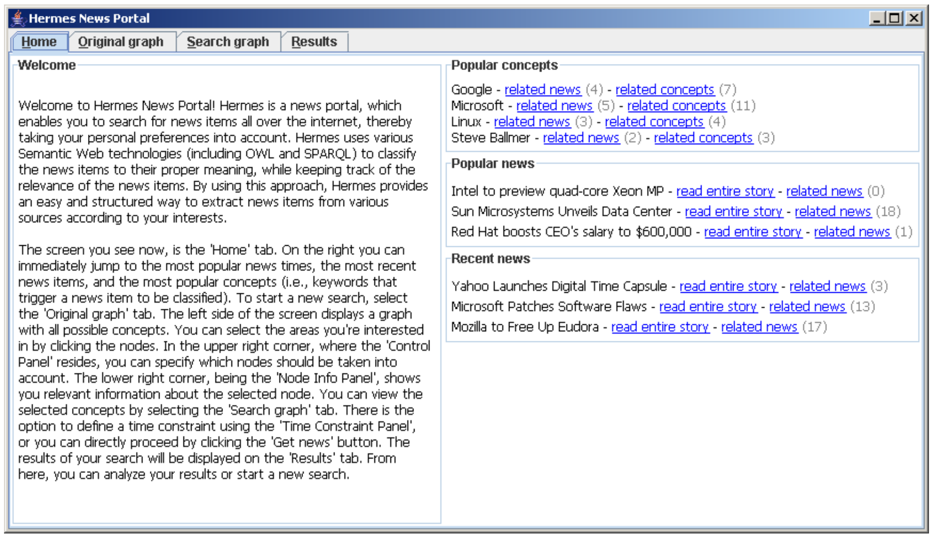
To clarify this, we introduce two example news items: `newsItemA` and `newsItemB`. Assume that `newsItemA` is classified because of the concepts `Microsoft` and `Google`, while `newsItemB` is classified because of the concepts `YouTube` and `Google`. These two news items are related, because they share a relation to the concept `Google`. The more concepts news items share, the more closely they are related. For each news item Hermes provides a link to 'Related news items', the grey number that follows this link indicates how many related news items have been found.

### Popular Items

Every time the user searches for news items in Hermes, the concepts that he selects and the news he reads are saved. This enables Hermes to show the most popular concepts and most popular news, as can be seen in Figure 11, which displays the screen presented to the user when the application is started. By clicking on the 'related news' link besides a concept, the user automatically obtains all the news related to that particular concept.

## Conclusion

With the Hermes news portal, we show the usefulness of Semantic Web technologies in the domain of analyzing news items and searching through them. In this paper we introduced an ontology-based news



Welcome

Welcome to Hermes News Portal! Hermes is a news portal, which enables you to search for news items all over the internet, thereby taking your personal preferences into account. Hermes uses various Semantic Web technologies (including OWL and SPARQL) to classify the news items to their proper meaning, while keeping track of the relevance of the news items. By using this approach, Hermes provides an easy and structured way to extract news items from various sources according to your interests.

The screen you see now, is the 'Home' tab. On the right you can immediately jump to the most popular news times, the most recent news items, and the most popular concepts (i.e., keywords that trigger a news item to be classified). To start a new search, select the 'Original graph' tab. The left side of the screen displays a graph with all possible concepts. You can select the areas you're interested in by clicking the nodes. In the upper right corner, where the 'Control Panel' resides, you can specify which nodes should be taken into account. The lower right corner, being the 'Node Info Panel', shows you relevant information about the selected node. You can view the selected concepts by selecting the 'Search graph' tab. There is the option to define a time constraint using the 'Time Constraint Panel', or you can directly proceed by clicking the 'Get news' button. The results of your search will be displayed on the 'Results' tab. From here, you can analyze your results or start a new search.

Popular concepts

Google - related news (4) - related concepts (7)
Microsoft - related news (5) - related concepts (11)
Linux - related news (3) - related concepts (4)
Steve Ballmer - related news (2) - related concepts (3)

Popular news

Intel to preview quad-core Xeon MP - read entire story - related news (0)
Sun Microsystems Unveils Data Center - read entire story - related news (18)
Red Hat boosts CEO's salary to $600,000 - read entire story - related news (1)

Recent news

Yahoo Launches Digital Time Capsule - read entire story - related news (3)
Microsoft Patches Software Flaws - read entire story - related news (13)
Mozilla to Free Up Eudora - read entire story - related news (17)

Figure 11: The view of the application when it is started.

classification, which allows the user to make use of relations between concepts when searching through the pool of news items. The state-of-the-art ontology visualization technologies assist the user by building queries, thereby enabling novice users to retrieve news about specific concepts in a powerful way. Our approach also facilitates displaying relevant related news and popular news items in a dynamic fashion, which leads the user to the most important news items of his interest.

Although Hermes is already a strong implementation of a future proof Semantic Web news portal, there is much room for improvement. The classifying performance in terms of concept recognition could be improved by adding support for hyponyms and meronyms that can be enabled for relevant concepts. Furthermore the world progressively changes, the ontology gets outdated. Therefore, Hermes should be extended with an adaptive self-learning ontology component, which should evaluate news items that reflect changes to the conceptual world of the ontology. For example, when news about a company taking over a competitor is released, the self-learning component builds a `hasProduct` relation between the company and the competitor's product (in fact, the self-learning component should even trigger the removal of the competitor from the ontology, because it has ceased to exist). To take this feature to the next level, it should also discover new concepts that come into existence (i.e., starting companies that challenge the existing companies).

Another feature that increases end-user satisfaction could be listing frequently used searches that the user may want to repeat. To keep things simple, we should not display the search query itself, but instead show the first results of this query. When the user is interested, he should be able to retrieve all results. By presenting the results of these frequently asked queries on the first page of the portal, we will then automatically provide a personalized news portal.

## About the Authors

**Jethro Borsje**, **Leonard Levering** and **Hanno Embregts** are all master students at the Erasmus University of Technology (the Netherlands), following the master program 'Economics & ICT'. They have experience in various research areas regarding the Semantic Web, including data querying, natural language processing, ontology visualization and news classification. Furthermore, they are active members of the open-source community, especially in Semantic Web-related projects such as writing plugins for IsaViz.
**Dr. Flavius Frasincar** is an assistant professor in information systems at Erasmus University of Technology (the Netherlands). From 2003 to 2005 he was an assistant professor at the Eindhoven University of Technology (the Netherlands). He holds a doctorate in computer science from the Eindhoven University of Technology. His main research areas are methodologies for developing Web Information Systems,

adaptive hypermedia, (Semantic) Web data representation languages (XML, RDF, OWL), (Semantic) Web query languages (XQuery, SeRQL, RDQL, OWL-QL), Web data transformation languages (XSLT), and Web data visualization. He is author of numerous publications in the areas of databases and Web Engineering. He (co-)organized and served as a PC member for several events in the Web Engineering field.

## References

[1] Tim Berners-Lee, James Hendler, and Ora Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001.

[2] Dan Brickley and R. V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft 10 February 2004, 2004.

[3] Graham Klyne and Jeremy J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. W3C Recommendation 10 February 2004, 2004.

[4] Deborah L. McGuinness and Frank van Hamelen. OWL Web Ontology Language. *W3C Recommendation*, 2004. `http://www.w3.org/TR/owl-features/`.

[5] Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. W3C Working Draft 20 February 2006, 2006.

[6] Akshay Java, Tim Finin, and Sergei Nirenburg. SemNews: A Semantic News Framework. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-06)*, pages 1939–1940. American Association of Artificial Intelligence, 2006.

[7] Akshay Java et al. Text Understanding Agents and the Semantic Web. *Proceedings of the 39th Hawaii International Conference on System Sciences*, 2006.

[8] H. Alani et al. Automatic Ontology-Based Knowledge Extraction from Web Documents. *IEEE Intelligent Systems*, 18(1):14–21, 2003.

[9] The Protg Ontology Editor and Knowledge Acquisition System. `http://protege.stanford.edu/`.

[10] Alessio Bosca et al. Ontosphere: More Than a 3d Ontology Visualization Tool. In *SWAP*, 2005.

[11] M.A. Storey et al.Akshay Java, Tim Finin, and Sergei Nirenburg. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in Protégé. In *Workshop on Interactive Tools for Knowledge Capture (K-CAP-2001)*, 2001.

[12] 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.

[13] Thomas R. Gruber. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human-Computer Studies*, 1995.

[14] Amarnath Gupta, Bertram Ludäscher, and Reagan W. Moore. Ontology Services for Curriculum Development in NSDL. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 219–220, 2002.

[15] Informa: RSS Library for Java. `http://informa.sourceforge.net/`.

[16] Brian McBride. Jena: A Semantic Web Toolkit. *IEEE Internet Computing*, 6(6):55–59, 2002.

[17] WordNet. `http://wordnet.princeton.edu/`.

[18] WordNet Web Application-Wordnet Enterprise Java Bean. `http://wnwa.sourceforge.net/`.

[19] Jeffrey Heer. Prefuse, Information Visualization Toolkit. `http://prefuse.org`.

[20] ARQ, a SPARQL Processor for Jena. `http://jena.sourceforge.net/ARQ/`.