A Semantic Web-Based Approach for Personalizing News

Kim Schouten
kim.schouten@gmail.com

Philip Ruijgrok
philip.ruijgrok@gmail.com

Jethro Borsje
jethroborsje@gmail.com

Flavius Frasincar
frasincar@ese.eur.nl

Leonard Levering
leonard@levering.eu

Frederik Hogenboom
fhogenboom@ese.eur.nl

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

ABSTRACT

Hermes is an ontology-based framework for building news personalization services. This framework consists of a news classification phase, which classifies the news, a knowledge base updating phase, which keeps the knowledge base up-to-date, a news querying phase, allowing the user to search the news for concepts of interest, and a results presentation phase, showing the returned news items. The focus of this paper is on how to keep the knowledge base up-to-date. For this purpose, we elaborate on the updating phase that searches for key events in the news. Using rules based on patterns and actions, these events can be extracted and the knowledge base is updated. This is a semi-automatic process since user validation is required before updating 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

News personalization, Ontology, Semantic Web

1. INTRODUCTION

Change - *noun* - 'an event that occurs when something passes from one state or phase to another'. Amongst others this definition can be found when searching for the word 'change' in WordNet. It's an intriguing concept because it involves everything in our lives. Sometimes we strive to change something and sometimes we work hard to prevent change. The world is constantly changing and the real alternative is to keep up with these changes and profit from them, either by reacting or perhaps by predicting these changes. Sometimes the state of the world changes unnoticed and there is hardly any effect on human beings. Sometimes however the world suddenly changes and people around the world are shocked, happy, or grieved. In economics knowing the state of the world is especially important since this knowledge can provide a source of profit and isn't that what most people are after?

To know the state of the world we have available different sources of news that provide the input for our personal knowledge base, ranging from news sites to newspapers, television, and friends. As the world seems to change more rapidly nowadays, it's even more vital to keep track of all that news. Unfortunately, while we call the world a big village, the information that's being generated in that big village is far too much to handle for humans whose village used to be their world.

Therefore, as humans have always done, we have developed tools to assist us in our endeavor to survive. Especially with the developing of the Web, it is now possible to search through online news items or filter them according to some text criteria. With news volumes still rising, this is however not enough. What we need is an intelligent system that knows what we are looking for. Such a system would not be searching for some mere words or phrases which we hope to be good proxies for what we are really after, but it would be searching for the desired items themselves. To be able to do that one needs to cope with human language, both its flexibility and its ambiguity.

In this paper, we propose a framework that is able to read news items and extract the valuable economic information from them, that allows the user to search for concepts instead of lexical representations for which one can only hope it matches the concepts one is looking for, and that is able to update its knowledge base to always represent the current state of the world. Using such a system will allow the semi-automatic discovery of world changes of interest for particular users. These changes would establish a more precise specification of the derived news items.

The concepts of interest are defined in a domain ontology and are associated to synsets from a semantic lexicon. For concept identification, we exploit the semantic lexicon from

SAC'10 March 22-26, 2010, Sierre, Switzerland.

which the concept lexical representations are retrieved (increase recall). After identifying a lexical representation in a news item, a word sense disambiguation procedure takes place to determine its corresponding sense (increase precision). For event recognition we use lexico-semantic patterns based on the previously identified concepts and subsequently use this information to update the ontology with new triples. Before updating the ontology the user has to validate the discovered information.

The paper is organized as follows. Section 2 discusses related work for news personalization. Then, in Section 3 we present the Hermes framework, which is the proposed framework for news personalization. Section 4 describes the Hermes News Portal, which is the implementation of the proposed framework. After that, in Section 5, our proposed solution is evaluated. The paper ends with Section 6 where we present our conclusions and possible future work.

2. RELATED WORK

Server for Adaptive News (SeAN) [2] previews news items of interest to the user based on the user model, which is created from user preferences and the user behavior. First the system requires the orthogonal stereotypes of the user (interests, domain expertise, cognitive characteristics, and life styles). All news is seen as a set of complex entities with attributes (title, abstract, body, pictures, videos, etc.). Using behavioral rules, the user model is updated with every interaction of the system and the user. Based on the user model, SeAN determines the appropriate level of interest per news item for the user and displays the news on this level of interest. Differently than SeAN, which uses a manual approach for news classification, the Hermes framework uses natural language processing techniques for classification of news items and Semantic Web technologies for representing the knowledge base.

YourNews [1] a keyword-based personalized news system, is open, editable, and transparent. The user can view their keyword profiles, and therefore the system is called open. However users can also update the keywords in their profiles, making the system editable. The matched keywords are highlighted for transparency. News items of interest are determined based on the weighted vectors of terms [22], with weights computed using TF×IDF [23]. For each visited news item, the weighted term factor is updated. With the cosine metric, the similarity between the user model and the news item is calculated. An interesting conclusion of this research is the decline in precision and recall compared with systems that do not have open and editable user profiles or user models. The Hermes framework differs from YourNews by using concepts instead of keywords for representing the users interests.

MyPlanet [12] is based on PlanetOnto, which aims at creating, delivering and querying internal news items of the Knowledge Media Institute. From ontology's, based on the specific ontology language OCML [15], the user can select concepts of interest. Classification is done based on the heuristics of cue phrases attached to ontology concepts. As an overview, the user can see an exhaustive tree representing the underlying ontology. Our approach uses OWL as the ontology language and classifies news items by NLP techniques. Furthermore, the knowledge base is represented as a graph instead of a tree, because a graph better visualizes the relationships stored in an ontology.

SemNews [10] is a Semantic Web based method for news presentations. News items are retrieved from RSS feeds and processed by the natural language processing engine OntoSem [17]. Using OntoSem the Text Meaning Representations (TMR) are retrieved; the TMR are converted to ontology's and these ontology's can be queried with RDQL [24], a precursor of the SPARQL [21] query language. Our approach differs from SemNews in two ways. First, the Hermes framework computes the word sense disambiguation on a widely available semantic lexicon, i.e., WordNet [7]. Second, querying can be done with graphical representations of the knowledge base, making the Hermes framework more intuitive.

The advantages of the Hermes framework over current approaches can be summarized as follows. First, the Hermes framework has an advanced NLP engine (using tokenization, part-of-speech tagging, ontology gazetteering, and word sense disambiguation) to 'understand' the news using a freely available semantic lexicon (i.e., WordNet). Querying upon the news can be done intuitively with graphical representations of the ontological graphs and with temporal constraints on the queries. Another important feature of Hermes, which represents the main focus of this paper, is the process of updating the knowledge base, where the system can semi-automatically update the underlying ontology, a feature that is absent in the investigated approaches. Also, the Hermes framework is not the implementation of a system, leaving the implementation free to use the most recent standards (like Semantic Web standards as OWL and SPARQL, or public tools as up-to-date Java libraries).

3. HERMES FRAMEWORK

The Hermes framework [5, 8] features the steps necessary to build a personalized news service. RSS news feeds are the input for the system which classifies and processes these news items, whereafter the user is able to search through these items using a graphical user interface having the news items of interest returned to him. At the heart of this system there is a knowledge base describing the general domain the user is interested in. This knowledge base is first used in classifying news items, then it is updated based on events that are described in the news, and finally a graphical representation of the updated knowledge base is used to generate the search query.

Even though the framework is able to update its knowledge base, it needs an initial knowledge base in order to do so. In other words, the framework does need a domain ontology to start with. For illustrative purposes we have used a domain ontology based on NASDAQ [13] listed companies. The knowledge base captures concepts like companies, chief executive officer (CEO)'s, products, and competitors. In addition, a news ontology is used to store the classified news with its metadata as well as a rule ontology, where rules are stored for updating the knowledge base with new information. These three ontologies are predefined and the first and the last ontologies are updated at run-time based on news and user provided information.

The Hermes framework has several phases. First, the *Classification* phase will read and classify the news using various NLP techniques. Here the news is stored in the news ontology. Second, the user has the possibility to search the news for specific events that might affect the knowledge base using the *Knowledge Base Updating* phase. After doing so,

the knowledge has incorporated the events featured in the news, and the user searches the news using the *News Querying* phase. Then, the *Results Presentation* phase takes over and shows the news the user was querying for. These main phases consist of various steps which will be explained in their corresponding sections. In [5, 8] we presented in detail the *Classification* phase, the *News Querying* phase, and the *Results Presentation* phase, and for the sake of completeness we will briefly review them in this paper too. Differently than in the previous papers, the focus here is on the *Knowledge Base Updating* phase which is the most important extension to our previous work.

It is important to note that our system has two types of user roles, the domain expert who creates and maintains the domain ontology and event rules, and the casual user which is interested in retrieving news items that match concepts of interest from the domain ontology. Thus, it is only the expert user who can validate the discovered events and allow for their effects to be propagated at run-time through the domain ontology.

3.1 Classification

In order to query the news for concepts of interest, the news itself must be first classified. After loading the news from the RSS feeds, it passes through a series of NLP steps, preparing the news for querying. The classification searches through the news to find all knowledge base concepts [8]. The news is then stored together with the found concepts in a news ontology, so that future queries can be done in a fast way, without going through all the NLP steps again.

The cornerstone of the Hermes framework is a domain ontology (knowledge base) that stores the most important concepts in the domain of interest. Concepts are mapped to their corresponding synsets (sets of synonyms) from a semantic lexicon. These synsets provide domain-independent lexical representations for the associated concepts, which complement the domain-specific lexical representation stored in the ontology. This broad range of lexical representations aims to increase the recall of concept identification in text. The domain ontology is developed and maintained by domain experts.

For the natural processing pipeline we use tokenization (words and punctuation signs), sentence splitting (sentences), part-of-speech tagging (noun, adjective, verb, etc.), morphological analysis (word lemmas), ontology gazetteering, and word sense disambiguation (the Structural Semantic Interconnection algorithm [16]). Each time a lexical representation corresponding to a concept from the domain ontology is found, a "hit" relation is stored between the concept and the news item. At the end of this process all news items are indexed using concepts from the domain ontology.

3.2 Knowledge Base Updating

Updating the knowledge base is a necessary step for maintaining an up-to-date knowledge base which is a good representation of the real world. Each news item is a possible source for new information or changed information in the knowledge base. The goal of the Hermes framework is to scan the information in the news items, recognize concepts known in the knowledge base, and detect changes in the real world.

We propose the usage of rules, for mining news information and updating the knowledge base. A rule consists of a lexico-semantic pattern, which has syntactic arguments based on ontological classes, and one or more actions which should be executed once the pattern is found. Using these lexico-semantic patterns we mine news items in order to find occurrences of events. Based on the identified events the actions are executed, thereby updating the ontology. Events patterns and actions are defined and maintained by domain experts.

In the rest of this section, we will illustrate our approach with the `kb:newCEO` event, an event that is triggered when the appointment of a new CEO is found for a company in the news. We assume that the new CEO and the company are in the knowledge base and the `kb:newCEO` event has two actions. First action removes the current (old) CEO from the company, and the second action inserts the newly appointed CEO to the company. Alternatively, the new CEO should be added to the knowledge base, prior to the actions, if it is not stored yet. One can note that a company can only have one CEO (restriction).

3.2.1 Event Rule Construction

We start by creating rules. Rules have lexico-semantic patterns defined by a subject and a relation. Rule patterns can have objects, however these are not required. An example of that rule pattern is:

```
[kb:Company] kb:GoesBankrupt
```

The subject represents a class in the knowledge base, i.e., `[kb:Company]`. The relation is an individual of the class `[kb:Relation]`, i.e., `kb:GoesBankrupt`. The square parentheses stand for lexical representations of individuals of the enclosed type, in the above example "*IBM*, "*International Business Machines Corporation*", "*EBay*", etc., as lexical representations of instances of type `kb:Company`. The lack of square parentheses means that only lexical representations of the given instance are to be taken into account, in our example, "*goes bankrupt*", "*is ruined*", etc., as lexical representations of the relation instance `kb:GoesBankrupt`.

An example of a rule pattern with both a subject and an object is:

```
[kb:Person] kb:BecomesCEO [kb:Company]
```

The subject and object represent classes, i.e., `[kb:Company]` and `[kb:Person]`. The lexical representations of `kb:company` instances are determined as before, and examples of lexical representations of `kb:person` instances are "*Steve Ballmer*", "*Carol Bartz*", etc. The relation is an individual of the class `[kb:Relation]`, i.e., `kb:BecomesCEO`. The ontology should have lexical representations of the relation `kb:BecomesCEO`, like: "*new CEO*", "*appointed CEO*", and "*becomes chief executive officer*". These lexical representations are stored in the knowledge base and are retrieved by querying. Once these lexical representations are found and they match the senses of the given concepts, the lexical-semantic pattern can trigger the `kb:newCEO` event.

3.2.2 Event Detection

Using rules it is possible to extract events from news. With a `kb:BecomesCEO` relation instantiation, a subject of type `[kb:Person]`, and an object of type `[kb:Company]`, the Hermes framework can find the `kb:newCEO` event in news items based on the previous rule.

3.2.3 Event Validation

The Hermes framework is a semi-automatic system, therefore events need to be manually validated. The disadvantage is the human interaction, which makes our system not fully automatic. However, the advantage is that the user, in this case the domain expert, can validate events before the underlying knowledge base is incorrectly updated.

The technologies used (or developed) until now to 'understand' news are not flawless and getting a one hundred percent precision and recall is simply not achievable. Since updating the knowledge base is crucial for an up-to-date knowledge base, and we want to maintain a correct knowledge base, manual validation is necessary.

3.2.4 Action Rules Construction

Finally, the action rules are updating the ontology. We recognize two different types of actions: add (insert) actions and remove (delete) actions. With these two types of actions we can update the knowledge base ontology. An action is formulated as a triple, with a subject, a relation, and an object. Returning to our example, we must first remove the current CEO from the knowledge base:

```
[kb:Company] rb:removeCEO [kb:Person]
```

and then insert the newly appointed CEO in the knowledge base:

```
[kb:Company] rb:addCEO [kb:Person]
```

In the example above we assume that the company and new CEO are already present in the knowledge base. The execution order of the actions is also defined in such a way that the ontology updates properly model the change triggered by the discovered news event.

3.2.5 Action Execution

Once the event is manually validated, the Hermes framework should immediately update the ontology with the action rules previously constructed. The action rules are ordered, i.e., we first want to remove the old CEO before we insert a new CEO because else we would simply insert a new CEO, and possibly delete this CEO making the update worthless. Also, as the same action can appear in multiple rules, the order of rule action execution is given by the order in which the events are found in news items (most recent events update last the ontology). After executing the actions in the correct order the event effects are captured in the knowledge base.

3.3 News Querying

With an updated news repository and an updated knowledge base, the user is now able to search the news for items of interest. The Hermes framework supports the user in constructing his search query. Furthermore, the user can specify time-based constraints that the time stamps of the news items need to satisfy. The news querying step basically consists of two steps: query formulation and query execution. In the first step, the query is constructed by the casual user, and in the second step, the results of the query are computed [8].

3.3.1 Query Formulation

To search for news, the user first has to choose which concepts are of interest. This selection is made using a graph-based visualization of the knowledge base. Here, the user can simply select nodes of interest, while in the same view it is clear to what other concepts the selected concepts are related to.

All information about the selected node, including its relationships, are also explained in textual form outside the graph. To support the user in finding the concepts of interest, a keyword-based search functionality is present allowing the user to search lexically through all the nodes.

Finally, the user can specify time constraints the news items have to satisfy. Since news items can quickly lose their validity as time passes, it is important to be able to filter news item based on time. Therefore, the framework allows the user to use either pre-defined constraints, such as last week or last month, or to manually build complex time expressions.

3.3.2 Query Execution

When all conditions are specified a query is generated. This query is executed and the relevant news items are returned. At this moment, the news items are not returned in any specific order.

3.4 Results Presentation

The results of the query are presented to the user in a logical order, having the news items which are most relevant at the top, and news items that are less relevant beneath them. The relevance degree used for this ranking is based on a weighted sum of the query concept hits in a news item, where the weights of a hit in the news item title are higher than the weights of a hit in the news item body. Among news items having the same degree of relevance the most recent news items should be on top [8].

In order to explain why a certain news item is being returned as an answer to the user query, we highlight the lexical representations of the query concepts found in news items. Each query concept is presented to the user who can turn it off or on, thus providing additional filtering capabilities on the result set.

4. HERMES NEWS PORTAL

The Hermes News Portal (HNP) is an implementation of the Hermes framework, which allows the user to browse through a knowledge base, and query for relevant news items, and supports the semi-automatic process of updating the knowledge base of Hermes. We present the HNP using the previously described phases and steps.

Because OWL [3] is supported by the W3C and offers additional features, useful in Hermes, compared to RDF and RDF Schema, it is chosen as our ontology language. For example, it is possible to describe in Hermes the disjointness of classes, the cardinality constraint, and the role inverse using OWL. These features are necessary for specifying restrictions such as the `[kb:Company]` class is disjoint from the `[kb:Person]` class, a `[kb:Company]` can have only one CEO, and that `kb:isCEOof` is the inverse of `kb:hasCEO`. The query language SPARQL [21] is chosen to query the OWL ontologies, being also supported by the W3C as the query language for RDF languages including OWL. To allow for comparison/arithmetic time operators and functions for retrieving the current time, SPARQL has been extended with time-related functionality resulting in tSPARQL [5].

The Java language is chosen for the implementation, since many libraries for manipulating, reasoning with, querying,

and visualizing ontologies are available. Also, it combines well with GATE [6], which we use for most of the NLP tasks, since both GATE and its components and plugins are programmed in Java. Jena [14] is used for manipulating and reasoning with ontologies. ARQ [11] is a SPARQL implementation, which we use for querying. SPARQL/Update [25] extends ARQ with update queries and is used for the knowledge base updates. For visualizing ontologies we employ OWL2Prefuse [4], a convenient library for visualizing OWL graphs using the generic Prefuse visualization library [9].

4.1 Classification

In order to query the news, we need to first classify them. To avoid a classification step before each query, the classification is done once and the results are stored. One can say that after classifying, the news is annotated with labels carrying the information of found concepts in the text.

In order to represent the hits relations between the ontology concepts and the news items we use instances of a specific ontology class. These instances store all the properties of the hit, as for example the found concept lexical representation and its position in the news item text. We chose to model such relations as instances by following a best practice recommendation for modeling N-ary relationships on the Semantic Web [19].

For example in the news item

> SAN FRANCISCO (Reuters) - Web search leader Google Inc. on Monday said it agreed to acquire top video entertainment site YouTube Inc. for $1.65 billion in stock, putting a lofty new value on consumer-generated media sites. [October 9th, 2006 at 20:15:33 CET]

three concepts are discovered: `kb:Google`, represented by the lexical representation "*Google Inc.*", `kb:Buy` represented by the lexical representation "to acquire", and `kb:YouTube`, represented by the lexical representation "*YouTube Inc.*".

4.2 Knowledge Base Updating

The process of updating the knowledge base is critical for an accurate representation of reality in the HNP domain ontology. The process is divided in five steps, starting with the construction of the rules and ending with the execution of actions that update the domain ontology.

4.2.1 Event Rule Construction

The first step in knowledge base updating is the construction of rules. Rules have lexico-semantic patterns based on triples, with a subject, a relation, and an optional object. Both subject and object are classes in the knowledge base and can be selected from a graphical representation of all classes in the knowledge base as seen in Figure 1. The relations are individuals in the class `kb:Relation` of the knowledge base, which are filled with one or more lexical representations.

Figure 2 exemplifies the new CEO rule. The subject is of type `[kb:Person]`. The object is of type `[kb:Company]`. The relation is `[kb:BecomesCEO]`, which is further defined in Figure 3. The `kb:BecomesCEO` relation has multiple representations like "*becomes chief executive officer*", "*new chief executive*", and "*appointed CEO*". Any of these representations in the news message should trigger the `kb:BecomesCEO` relation.

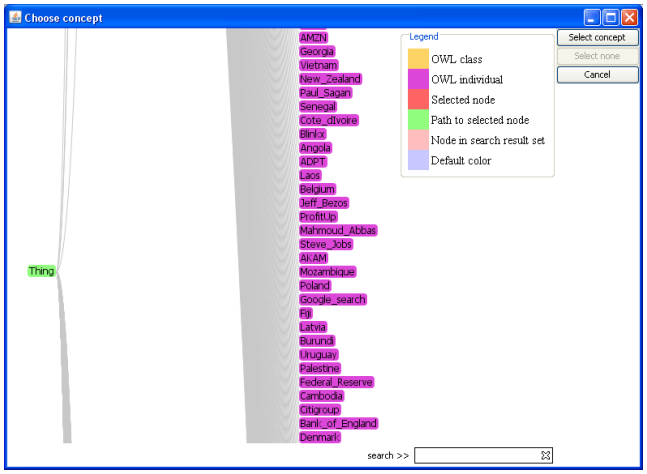

Figure 1: The rule editor - select concepts.

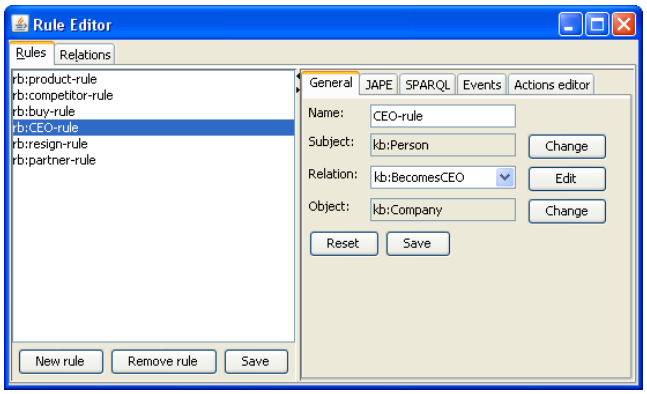

Figure 2: The rule editor - rules.

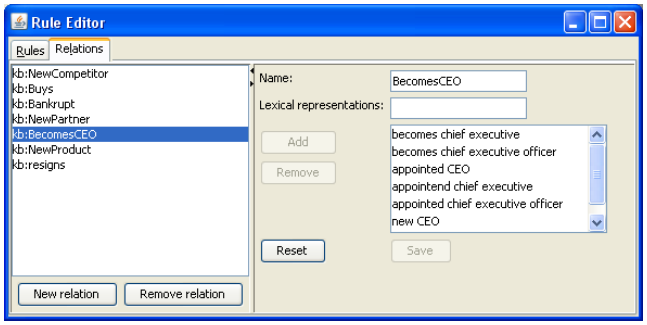

Figure 3: The rule editor - relations.

4.2.2 Event Detection

As explained in the previous section, even when the keyword for an event is found in the text, it is imperative that we find all the information related to the event. For example, when we have found a `kb:Buy` event, it remains a useless piece of information if we don't know who bought what. In other words, we have to know the subject of an event and if applicable also the object of an event. To find this information we use the so-called JAPE rules [6] to find patterns in the text around the previously identified event representation.

JAPE rules consist of two parts: a so-called left-hand side describing the pattern that needs to be matched, and a right-hand side describing the actions that will take place when the pattern is found. For the left-hand side, a form of regular expressions is used. For the right-hand side, a simple notation is developed that makes it easy to produce an annotation for the matched piece of text. When more elaborate actions are needed (e.g., for removing temporary annotations, manipulating previous annotations, etc.) it is allowed to introduce Java code for the whole right-hand side.

When again we use the `kb:Buy` event, we need to also find the buyer and the buyee. In our domain that would be companies, resulting in the following pattern to be matched: `[kb:Company] kb:Buy [kb:Company]`. Obviously, both companies have to be present in the knowledge base to be able to recognize them, stressing again the importance of a up-to-date knowledge base. When this pattern is found, the corresponding triple is saved in memory for validation.

4.2.3 *Event Validation*

Updating the knowledge base is a semi-automatic procedure, the HNP requires the user to validate the automatically recognized events. The user is shown the event, with the subject, relation, object, and the corresponding news item(s). The user has to decide whether this is a valid, invalid, or unknown event. Figure 4 shows the event validation in HNP. If the event is manually validated, the HNP updates the knowledge base with the following two steps.

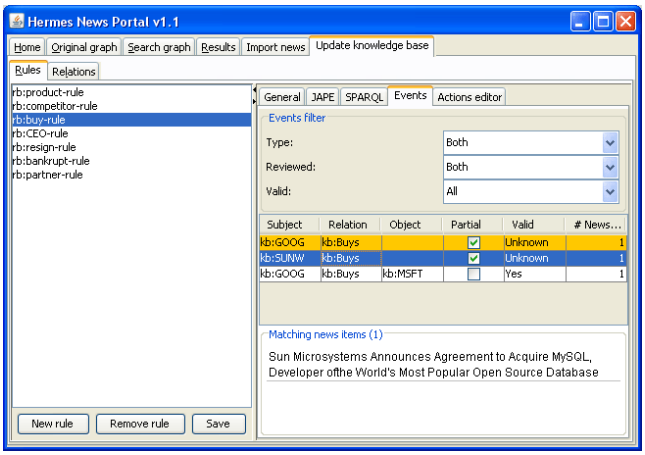

Figure 4: The rule editor - events.

4.2.4 *Rule Actions Construction*

Each rule has one or more actions attached. These actions are used to update the ontology. We differentiate two types of actions: add (insert) and remove (delete) actions. The actions are divided into a main clause and a where clause. These clauses contain one or multiple triples (subject - predicate - object).

The construction of the actions should be general, therefore the HNP allows the triple's subject and object to be represented as a template, replacing the subject and object by `<event subject>` and `<event object>`, respectively. The user also specifies the order in which the actions need to be executed. Figure 5 shows the action editor screen of the HNP.

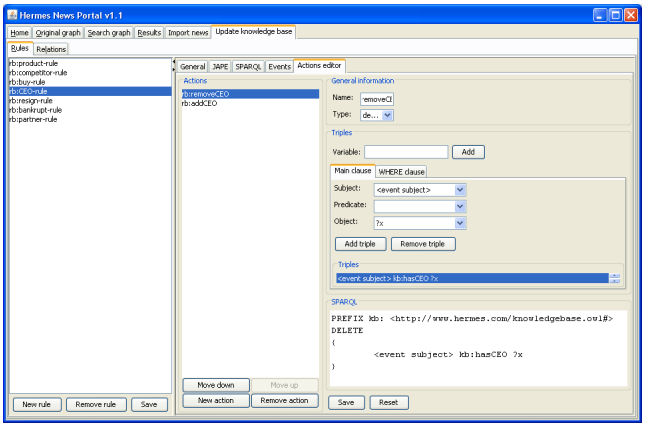

Figure 5: The rule editor - actions.

4.2.5 *Action Execution*

The final step of knowledge base updating is the execution of the actions, which happens directly when a user validates an event. The HNP executes the actions corresponding to the event, which are basically SPARQL queries being executed on the knowledge base.

4.3 News Querying

Querying the news in the HNP is rather easy with the graphical representation of the knowledge base. First, the user can select the concept(s) of interest and specify a time window. Second, the HNP creates the (complex) queries for the user.

4.3.1 *Query Formulation*

First, the user must select his concepts of interest from the knowledge base. To facilitate the user in this task, the HNP displays a graphical representation of the knowledge base using Prefuse [9] and OWL2Prefuse [4]. All concepts and their relations are shown in ovals and respectively arrows as depicted in Figure 6.

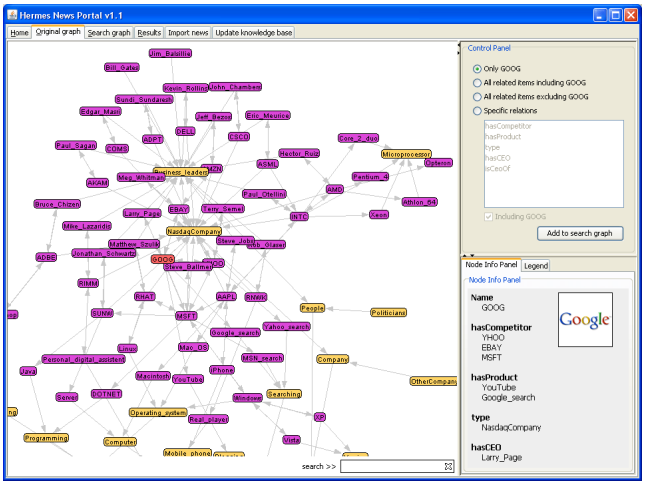

Figure 6: The search graph - select concepts.

The visualization uses different colors for classes and individuals, as well as color variations for selected nodes, neigh-

bors of the selected node, and nodes which correspond to a lexical search. A legend explaining all colors can be found on the right, under the `Legend` tab.

Figure 7 shows the search graph displaying all concepts added by the user. With the `Get news` button, the queries are generated and subsequently executed. The HNP uses SPARQL as the query language, extended with a time functionality. This allows the HNP to query ontologies with temporal constraints. This extended query language is called tSPARQL [8].

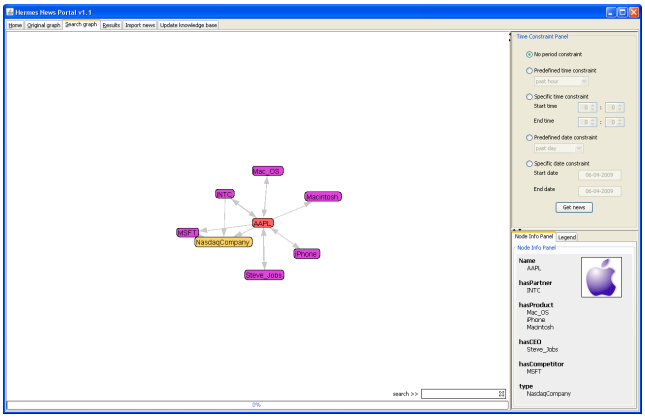

Figure 7: The search graph - view selected concepts.

4.3.2 *Query Execution*

Based upon the user selections in the previous graphs and panels, the query has been formulated and is now ready for execution. After its creation, the query is executed, displaying the results in the `Results` tab.

4.4 Results Presentation

Figure 8 shows the `Results` tab where the user is presented all news items which are of his interest. Within this screen the user can refine his search query by deselecting/reselecting concepts from his search query.

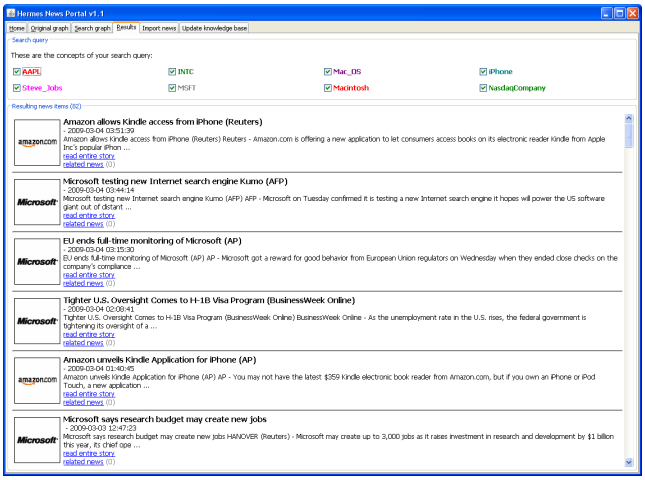

Figure 8: The results - news items of interest.

5. EVALUATION

For the evaluation of the HNP we propose four performance measures: precision, recall, latency, usability, and expressivity. The precision is measured as the number of concepts correctly found in the news divided by the number of concepts found in the news. The recall is defined as the number of concepts correctly classified in the news divided by the number of concepts that is in the news according to domain experts. The latency is computed based upon the time the classification process consumed, or from pressing the start button till the news has been written to disk.

For the experiments we have constructed one newsfeed (a combination from the Yahoo! business and technology newsfeeds), which contains 200 news items. This newsfeed was used for evaluation of the precision, recall, and latency. The precision for the concept identification in news items is 86% and the recall 81%, which is similar to results reported in other studies [18, 20]. Nevertheless an objective comparison, between our system and existing systems is difficult to achieve as the reported figures for precision and recall are based on experiments performed on different data sets.

The errors come from the word sense disambiguation step, missing lexical representations of the knowledge base concepts, and missing concepts from the knowledge base. The latency is around 1 second per item. Note that this time includes the relatively slow process of writing and reading the ontology's to and from disk.

The usability was determined by 9 users (students at Erasmus University Rotterdam following a course on the Semantic Web, which includes RDF(S), OWL, and SPARQL). To determine the usability the users had to query the news for all new items related to Google or one of its competitors that appeared in the last three months. The usability is measured in three ways. First, the query has to be correctly built, second, the time to build a query is measured, and third, the user's impressions after interacting with the systems are recorded. All users were able to query the news by the search graph correctly, and all users could do this faster than manually writing queries in SPARQL. Note these results are obtained querying not any RDF graph, but querying the graph corresponding to the knowledge base of the HNP. The users appreciated the graphical representation of the knowledge base, the predefined time functionality, and the transparency obtained by highlighting the concepts found in the news items.

The expressivity of the HNP is limited to all functions described in the previous section. This implementation of the HNP is sufficient for simple usage, and was created to be an easy-to-use, clear, and reliable program. However, we recognize that advanced flexible usage of querying is not possible, there is no possibility to construct more advanced queries as for example conjunctive queries, e.g., retrieve all news items that mention both Google and Yahoo!, and pattern-based queries, e.g., retrieve all news items that refer to Google acquiring another company.

6. CONCLUSIONS

This paper describes the Hermes framework, a set of steps that can be used for building and maintaining a news personalization service. The system can be described by input, internal processing, and output. The input is composed of pre-defined RSS feeds of news items and concepts selected

by the user. The internal processing is the classification of these news item, finding concepts from a knowledge base, and the updating of the knowledge base underlying the system. For updating the knowledge base, which is the core functionality presented in this paper, we defined rules based on lexico-semantic patterns and actions. The output is given by the personalized news items based on selected concepts.

The Hermes News Portal (HNP) is our implementation of the Hermes framework. It allows the user to query the news and view the knowledge base. The domain ontology is represented in OWL, querying is done with SPARQL, and time functionalities were added to SPARQL. Classification is done using GATE and the WordNet semantic lexicon. The knowledge base updates are performed using SPARQL/Update.

The current Hermes framework and the HNP can be extended in several ways. For example, we can increase the number of concepts, events, and patterns in the current knowledge and rule base. This should result in a higher recall when querying the system for news. Also, adding new domains to the system would increase functionality, currently the only domain supported being financial news. Last, we would like to add new types of user queries as for example conjunctive queries and pattern-based queries.

7. ACKNOWLEDGMENTS

The authors are partially supported by the EU funded IST-STREP Project FP6-26896: Time-Determined Ontology-Based Information System for Real Time Stock Market Analysis (TOWL). More information is available on the official website of the TOWL project [26]. Also, we would like to thank Wouter Rijvordt, Maarten Mulders, and Hanno Embregts for their contribution to the Hermes framework.

8. REFERENCES

[1] J. Ahn, P. Brusilovsky, J. Grady, D. He, and S. Y. Syn. Open user profiles for adaptive news systems: Help or harm? In *16th International Conference on World Wide Web*, pages 11–20. ACM, 2007.

[2] L. Ardissone, L. Console, and I. Torre. An adaptive system for the personalized access to news. *AI Communications*, 14(3):129–147, 2001.

[3] 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.

[4] J. Borsje. OWL2Prefuse, 2009. `http://owl2prefuse.sourceforge.net/`.

[5] J. Borsje, L. Levering, and F. Frasincar. Hermes: a Semantic Web-based news decision support system. In *23rd Annual ACM Symposium on Applied Computing (SAC 2008)*, pages 2415–2420. ACM, 2008.

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

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

[8] F. Frasincar, J. Borsje, and L. Levering. A Semantic Web-based approach for building persoanlized news services. *International Journal of E-Business Research*, 5(3):35–53, 2009.

[9] J. Heer. Prefuse, information visualization toolkit, 2009. `http://prefuse.org`.

[10] 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.

[11] Jena Development Team. ARQ, a SPARQL processor for Jena, 2009. `http://jena.sourceforge.net/ARQ/`.

[12] 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*, pages 140–148, 2001.

[13] E. Kandel and L. M. Marx. NASDAQ market structure and spread patterns. *Journal of Financial Economics*, 45(1):61–89, 1997.

[14] B. McBride. Jena: Semantic Web toolkit. *IEEE Internet Computing*, 6(6):55–59, 2002.

[15] E. Motta. *Reusable Components for Knowledge Modelling: Case Studies in Parametric Design Problem Solving*, volume 53 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 1999.

[16] R. Navigli and P. Velardi. Structural Semantic Interconnections: A knowledge-based approach to word sense disambiguation. *IEEE Transations Pattern Analalysis and Machine Intelligence*, 27(7):1075–1086, 2005.

[17] S. Nirenburg and V. Raskin. Ontological semantics, formal ontology, and ambiguity. In *Formal Ontology in Information Systems*, pages 151–161. ACM, 2001.

[18] I. Novalija and D. Mladenic. Semi-automatic ontology extension using text mining. In *Conference on Data Mining and Data Warehousing (SiKDD 2008)*, 2009. `http://kt.ijs.si/dunja/SiKDD2009/Papers/InnaNovalija.pdf`.

[19] N. Noy and A. Rector. Defining N-ary relations on the Semantic Web. W3C Working Group Note 12 April 2006, 2006.

[20] B. Popov, A. Kiryakov, A. Kirilov, D. Manov, D. Ognyanoff, and M. Goranov. KIM - Semantic annotation platform. In *Second International Semantic Web Conference (ISWC 2003)*, volume 2870 of *Lecture Notes in Computer Science*, pages 834–849. Springer, 2003.

[21] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. W3C Recommendation 15 January 2008, 2008.

[22] G. Salton. The smart retrieval system. In *Experiments in Automatic Document Processing*. Prentice Hall, 1971.

[23] G. Salton and M. McGill. *Introduction to Modern Retrieval*. McGraw-Hill, 1983.

[24] A. Seaborne. RDQL - a query language for RDF. W3C Member Submission 9 January 2004, 2004.

[25] A. Seaborne and G. Manjunath. SPARQL/Update, a language for updating RDF graphs, 2009. `http://jena.hpl.hp.com/~afs/SPARQL-Update.html`.

[26] TOWL Consortium. Time-determined ontology-based information system for real time stock market analysis (TOWL). `http://www.semlab.nl/towl`, 2009.