A Personalized Hotel Selection Engine

Jörg Garbers
Freie Universität Berlin
Institut für Produktion,
Wirtschaftsinformatik und OR
Garystr. 21, D-14195 Berlin
jgarbers@zedat.fu-berlin.de

Magnus Niemann
Freie Universität Berlin
Netzbasierte
Informationssysteme
Takustr. 9, D-14195 Berlin
maggi@inf.fu-berlin.de

Malgorzata Mochol
Freie Universität Berlin
Netzbasierte
Informationssysteme
Takustr. 9, D-14195 Berlin
mochol@inf.fu-berlin.de

ABSTRACT

In this poster we describe the results of our work in the *Reisewissen* project which adopts Semantic Web (SW) technologies for the tourism domain. In the prototypical implementation of a SW based portal for hotel booking we use RDF as a shared data representation scheme utilized by evaluation components written in Java and Prolog. We apply Prolog to transform data between external and internal ontologies and to represent expert knowledge. Since our prototypical implementation is to be integrated into a real world production system where performance plays a crucial role, the developed solution is based on a mix of SW technologies and classical knowledge representation tools.

Keywords

Tourism Domain, Semantic Web, Ontologies, OpenGuides

1. MOTIVATION

The past 10 years of Web evolution has led to the establishment of electronic markets. The next 10 years may be characterized by the transformation of the Web from a document publication medium intended for human utilization into a medium for intelligent knowledge exchange [3]. In the light of these developments the tourism domain, which is already exhibiting a gradual shift towards electronic transactions, can benefit from embracing new and emerging technologies.

Although current online travel systems support the customer in finding a suitable hotel or even a whole trip, most of the work is still up to the customer, who has to consider several sources of information (hotel review sites, booking portals, hotel websites) before deciding which hotel to book.

Our goal in the *Reisewissen* project (`reisewissen.ag-nbi.de`) is, on one hand, to support the user in the choice of a hotel by *selecting* and *ranking* suitable hotels and, on the other hand, to evaluate the usefulness of SW [1] technologies in this context.

The hotel selection and ranking is accomplished by the integration of several heterogeneous data sources into the hotel evaluation process and semantically matching the collected hotel information to the customer's individual profile. Furthermore, we enrich the collected data with domain expert knowledge (collected during "experts' interviews") represented with Prolog rules on top of RDF(S)/OWL [4]. The developed selection and ranking engines will be used by our industrial partner, *ehotel AG* (`www.ehotel.de`) to enhance its online hotel booking system. Since our prototypical implementation is to be integrated into a real world production system where performance plays a crucial role, the final system is based on usage of a mix of SW technologies and classical knowledge representation tools.

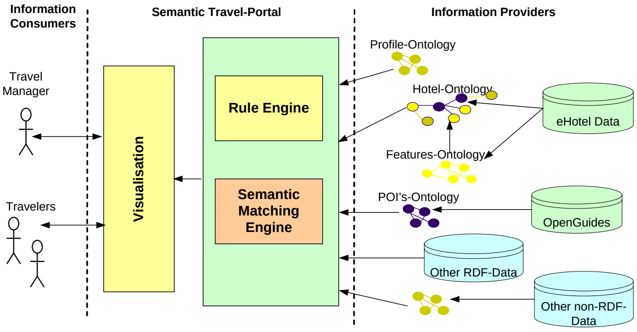


Figure 1: *Reisewissen* hotel recommendation engine

2. TECHNICAL REALIZATION

When we view our system (see Fig. 1) in a bottom-up way, we integrate several heterogeneous data sources, provide means to formalize and use expert knowledge and develop a suitable user interface for collecting customer requirements, expert knowledge and data from additional sources.

To rank hotels in a personalized way, we transform the customer's requirements into mathematical objective functions. A weighted combination of these functions is evaluated against each member of a set of hotels, yielding a ranking for the respective hotel (see Fig. 2).

2.1 Data Sources and Ontologies

We integrate various RDF and non-RDF data from different sources (see Fig. 1) such as hotel information (location, room rates, amenities etc.) provided by *ehotel AG* in XML

Demos and Posters of the *3rd European Semantic Web Conference (ESWC 2006)*, Budva, Montenegro, 11th - 14th June, 2006

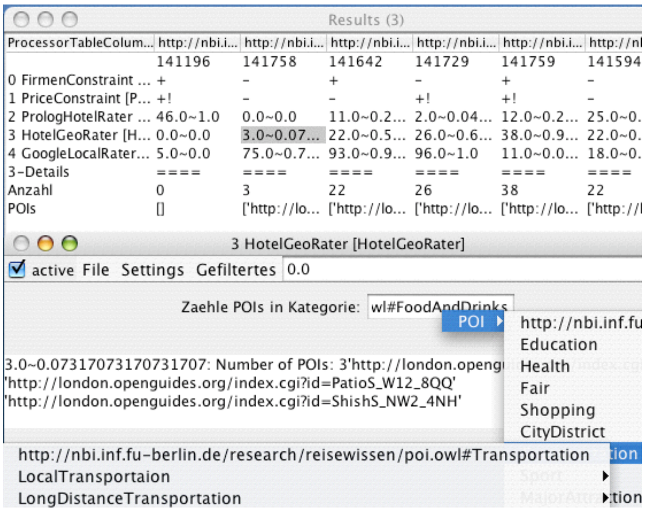
format according to the Open Travel specification (`www.opentravel.org`), hotel reviews (spreadsheets) and information about points of interest (POI) in RDF format from OpenGuides (`www.openguides.org`). External RDF data is transformed into our own domain specific ontology by means of *transformation rules* formalized in Prolog or Java while for proprietary, non-RDF data we provide an RDF view or RDF dump. Where performance is crucial, we keep or cache data in specialized data structures. For example, we extract location based RDF information from OpenGuides, translate it to our POI ontology and cache geocoded information in grid-like data structures for fast lookup.

2.2 Rules

To use concrete RDF data to estimate hotel's quality w.r.t specific customer's requirements we use Java methods that may query Prolog rules. The latter are used for two purposes in our project:

Rules for consolidating data into knowledge, e.g. the rule of thumb that everywhere in London there are Indian restaurants because of the high Indian population. Customers who require vegetarian food will be satisfied everywhere, without knowing about particular instances of vegetarian restaurants near a hotel.

Rules used in matching process to compare customer profile properties on predicated knowledge with the characteristic of a hotel, e.g. the physically disabled customer who is planing to use public transportation needs a hotel near a bus/subway station which also has access to an elevator.



ProcessorTableColum...	http://nbi.i...	http://nbi.i...	http://nbi.i...	http://nbi.i...	http://nbi.i...	http://nl
	141196	141758	141642	141729	141759	141594
0 FirmenConstraint ...	+	-	+	-	+	-
1 PriceConstraint [P...	+!	-	-	+!	+!	-
2 PrologHotelRater ...	46.0~1.0	0.0~0.0	11.0~0.2...	2.0~0.04...	12.0~0.2...	25.0~0.
3 HotelGeoRater [H...	0.0~0.0	3.0~0.07...	22.0~0.5...	26.0~0.6...	38.0~0.9...	22.0~0.
4 GoogleLocalRater...	5.0~0.0	75.0~0.7...	93.0~0.9...	96.0~1.0	11.0~0.0...	18.0~0.
3-Details	====	====	====	====	====	====
Anzahl	0	3	22	26	38	22
POIs	[]	['http://lo...	['http://lo...	['http://lo...	['http://lo...	['http://l

☑ active File Settings Gefiltertes 0.0

Zaehle POIs in Kategorie: wl#FoodAndDrinks

```
3.0~0.073170731707317 07: Number of POIs: 3'http://london.openg
'http://london.openguides.org/index.cgi?id=PatioS_W12_8QQ'
'http://london.openguides.org/index.cgi?id=ShishS_NW2_4NH'
```

POI
http://nbi.inf.fu
Education
Health
Fair
Shopping
CityDistrict
tion
http://nbi.inf.fu-berlin.de/research/reisewissen/poi.owl#Transportation
LocalTransportaion
LongDistanceTransportation
tion

Figure 2: The domain expert's testing environment

2.3 (Expert) User interfaces

In order to develop an application which can support all types of prospective users (domain experts, end and power users) we implement different user interfaces.

Tourism domain experts need a workbench to test the quality of data and rules (see Fig. 2). A number of hotels (table headers) are evaluated against several objective functions (leftmost column in the table). A function is implemented as a java class that specifies the parametrization user interface in terms of text fields and choices. A function instance is parameterized by the expert. Text field values may be set by picking URIs from an ontology driven popup menu. The experts formulate basic evaluation functions and compare the ranking result with their expectations. They can also combine several basic evaluation functions into complex ones which can be chosen by the end user (e.g. suitability of the hotel for wellness, business). *Advanced users* may wish to build their own objective function by building a preference tree over given basic evaluation functions. Evaluation functions can be parameterized using a form-based interface and ontology based functions using a tree of weighted ontology concepts.

3. CONCLUSIONS

The requirements analysis of the hotel recommendation engine raised several constraints which we have to take into account. We have to ensure that the engine is flexible with regard to later adaption to the production system and efficient regarding the end user querying process. Furthermore it has to allow an easy integration of information sources and provide means to generate new information from appropriately formalized expert knowledge. The excess value of semantic web technologies lies in ontology-based applications like semantic matching and similarity searches, moreover we make use of already published RDF metadata. When it comes to querying at runtime and numerical evaluation, those technologies proved to be less efficient.

From the implementation point of view we found that using Java/Jena as a RDF framework was cumbersome and slow. For the most cases where data could be kept in memory, SWI prolog's Semantic Web Library (`www.swi-prolog.org/packages/semweb.html`) was the better choice, thus using Prolog as the rule language appeared reasonable. In the existing prototype Java/Swing is used for the user interface and for simple evaluation tasks. Prolog is connected via the java native interface. For system simplicity we consider to dump the prolog generated knowledge into SQL databases, from where the data could be provided as virtual Jena graphs via D2RQ[2].

4. ACKNOWLEDGMENTS

This work is part of the *Reisewissen* project, a cooperation between Freie Universität Berlin and ehotel AG; supported by the Investitionsbank Berlin (IBB) and by the European Regional Development Fund (ERDF).

5. ADDITIONAL AUTHORS

Additional authors: R. Oldakowski (`radol@gmx.de`), R. Tolksdorf (`tolk@inf.fu-berlin.de`) and C. Bizer (`bizer@zedat.fu-berlin.de`) – all Freie Universität Berlin.

6. REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, pages 34–43, May 01.

[2] C. Bizer and A. Seaborne. D2RQ – Treating Non-RDF Databases as Virtual RDF Graphs. *Proc. of the 3rd Int. Semantic Web Conference (ISWC2004)*, 2004.

[3] J. Hendler, T. Berners-Lee, and E. Miller. Integrating Applications on the Semantic Web. *Journal of the Institute of Electronic Engineers of Japan*, 122(10):676–680, Oct. 02.

[4] G. Klyne and J. J. Carroll. Resource Description Framework (RDF): Concepts and Abstract Syntax. http://www.w3.org/TR/rdf-concepts/, Feb. 04.