Semantically Enhanced Deep Web

Witold Abramowicz, Dominik Flejter, Tomasz Kaczmarek, Monika Starzecka,
Adam Walczak
Poznan University of Economics, (*f.lastname*)@kie.ae.poznan.pl

Abstract: In this paper we address the problem of introducing semantics into Deep Web. Our main contribution is an annotation-based model for semantic, unified access to heterogeneous Deep Web sources. We discuss the problem of source navigation and semantics modeling, introduce personalized views on the ontology and demonstrate their use for navigation and extraction of Deep Web data.

1 Introduction

The World Wide Web evolved as an information space for human consumption, missing support for automated information processing. To overcome this obstacle, Semantic Web paradigm was proposed [BL98]. A lot of effort in recent years was focused on semantifying existing Web resources. Surprisingly, despite claims that the Semantic Web is most suited for bridging the gap between heterogeneous data, most of research in this area ignored challenges posed by Deep Web, which consists of on-line databases. The Deep Web could be very interesting application area for semantic technologies, since it is hundreds times larger than the surface Web as reported in [Ber01, HPZC07] and contains data of more value, quality and structure than surface Web. In this paper we introduce semantic means to address Deep Web challenges. Our main contribution is an annotation-based model for unified semantic access to heterogeneous Deep Web sources.

2 Related Work

There is a number of data processing challenges posed by Deep Web. Its indexing is hard because user input (e.g. filling in forms) is needed to access the complete content. Accessing Deep Web data often requires complex navigation (as data of interest is typically dispersed into multiple pages) and data extraction mechanisms. Finally, in order to allow automated processing of multiple Deep Web sources, schema mapping and introduction of shared semantics is necessary. Out of these challenges our special focus is on navigation, data extraction and shared semantics. This can be achieved by semantic annotation of navigation and data extraction model.

A lot of research focused on data extraction and wrapper generation (see [LRNdST02] for an extensive overview). However, only a few Deep Web data extraction solutions were de-

veloped, with many Deep Web systems (including [RGM01, CHZ05, NZC05, ARP+07]) focusing rather on accessing and indexing content than on extracting data from Deep Web sources. [LYE02] describes a system able to automatically fill in Deep Web forms, with some basic navigation (e.g. moving to next result page) and data processing capabilities (sentence-level filtering of duplicate records). However, it offers no actual data extraction and no means of flexible navigation modeling. This functionality may be found for instance in Lixto[BCL05].

While annotation of HTML documents is a well documented problem with many solutions (see [Nag03] for an overview), we are aware of only one annotation approach focused on Deep Web sources. [HSV03] describes "deep annotation" framework for cooperative (i.e. actively participating in Semantic Web) Deep Web sites. In this approach mappings linking server-side Web page markup (reflecting source's database structure) and client-side domain ontology are published on the Web and reused by any number of clients for querying Deep Web source semantically.

Our approach differs from previous work in the following aspects: (1) it strongly stresses the importance of navigation modeling; (2) it operates at the level of structured data rather than unstructured content, with capability to extract data records with attributes dispersed into multiple Web pages; (3) it proposes an integrated approach to extraction of data and information necessary for navigating the source; and (4) it uses annotation with domain ontologies to introduce semantics both to inputs (e.g. forms to be filled in) and outputs (i.e. the extracted data) of possibly uncooperative Deep Web sources. In this paper we focus on the fourth point, giving just a short description of other aspects.

3 Motivating Examples

In the first scenario, let's imagine a user willing to buy black Audi A4 Quattro 2.0, produced in 2007. She may use eBay Motors[1] as a source of information. At this site she fills make (Audi), model (A4) and year (2007) into advanced search form and manually selects all "Quattro 2.0 black" cars from list of results (*list page*). Information on car version (Quattro) and engine capacity (2.0) is present at list page. However, she needs to go to page of individual car (*detail page*) to check if car is black and learn other car details. Thus, this process is quite laborious. Moreover, if multiple sources are used, new challenges become apparent. Firstly, the number of relevant sites may be very high, making manual data gathering unrealistic. Secondly, each data source may have different navigation model, form querying capabilities and data presentation resulting in user confusion. Finally, data organization of sources may be very different from the one preferred by the user. Ideally, we would like to enable user to access all sources in one user interface consistent with her view of the domain of interest.

In the second scenario, let's imagine an automotive broker company, that has developed software able to analyze how competitive various car offers are. It would be interested in gathering car offers from a multitude of Web and Deep Web sources to implement large-

[1] http://motors.ebay.com/

scale discovery of candidates for good deal. While its employees may enter (i.e. copy & paste) offers to this system manually, automated processing would decrease significantly analysis time and enable much larger scale of operations.

Both presented situations share main common challenge which is providing shared conceptual view of different data sources. Moreover, in the first scenario, definition of personalized view on top of shared model should be possible. Our approach to these challenges is based on three components: semantically enhanced finite state transducers (FST) (Section 4), user views (Section 5) and Semantically Enhanced Extraction Engine (SE^3) (Section 6).

4 Semantically Enhanced FST for Source Modeling

A Deep Web site typically contains multiple dynamically generated Web pages corresponding to different data views. Intuitively, some pages may share common function and presentation features. Such similar pages may be further generalized into *classes of pages*. Consecutively, data extraction rules may be assigned to each class in order to extract specific data records and attributes. In case eBay Motors there are three distinct classes: page containing form (*form page*), list pages and detail pages.

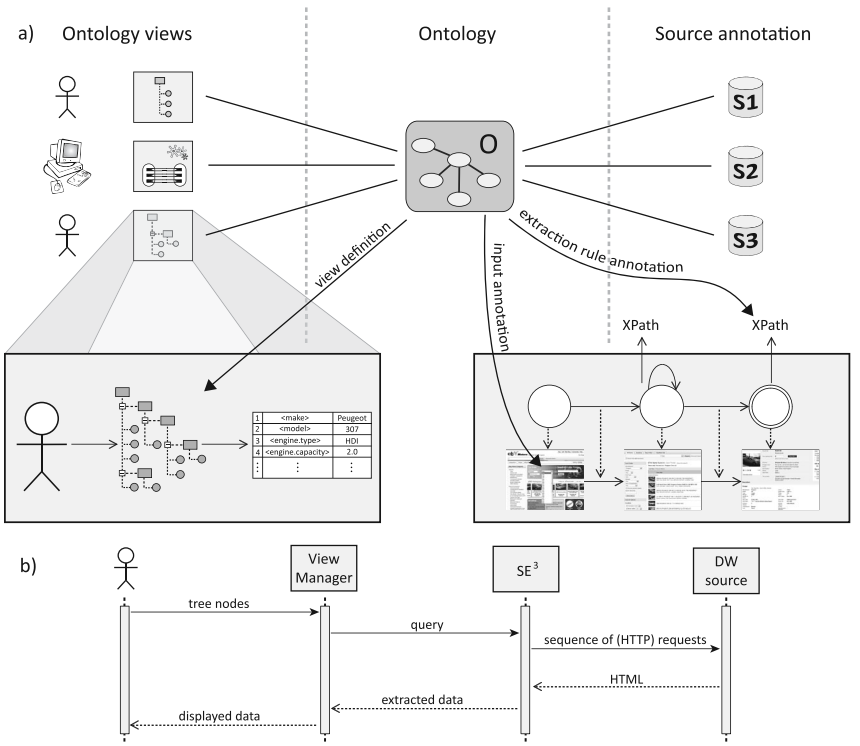
As illustrated by eBay Motors, a Deep Web site rarely gives direct access to data of interests. Typically, a sequence of navigation actions needs to be performed to access data and values of multiple attributes of the same record are dispersed into several classes of pages. Intuitively, generalization of pages into classes may be accompanied by generalization of their URLs into *request templates*. By associating links followed from given page with request templates, one may recognize the class of destination page. Altogether, page classes and request templates form the generalized model of Web site's navigation represented as a *finite state transducers (FST)*. The states of this FST correspond to classes of pages, input symbols - to requests templates and output symbols - to sets of data extraction rules. This representation is flexible and well captures both navigation and data extraction patterns. Exemplary FST for eBay Motors is presented on the right hand side of Fig. 1a).

Each request template contains any number of parts that differ for individual requests (called *input slots*). Some of them are filled with user input (e.g. make and model in eBay Motors form), while other require data extracted from current page (e.g. ID of specific car at list page) or previously visited pages (e.g. session ID). After values for all input slots are set, HTTP requests are built from request templates, allowing for further navigation.

Extraction rules combine regular expressions with XPath statements, that proved to be expressive enough and robust to page layout changes [KOKA06]. Extraction rules, similarly to request templates, may have input slots (e.g. allowing XPath extraction relative to TD containing model name) that need to be filled in before actual extraction is performed.

To allow unified approach to user input and extracted data, the FST is further annotated with attribute names from domain ontology, forming *semantically enhanced FST (SEFST)*. Two aspects of FST which are annotated are inputs slots (of both extraction rules and request templates) and outputs of extraction rules. In case of eBay Motors the annotation

Figure 1: Overview of System's a) Conceptual Architecture b) Data Flow



is attached to make and model attributes of form and to all extracted attributes. Thanks to annotation of input slots, automatic filling of forms with ontology instances is possible. Annotation of extraction rule outputs allows to assign semantics to extracted attributes (e.g. price, body type or mileage). As a result, all acquired data may be exported as ontology and are further processable by other programs (e.g. broker company analytical suite).

While currently both transducers and their annotations are created and maintained by domain experts, we plan to (at least partially) automate these processes in our future work.

5 User Views

As needs and domain understanding of individual users may differ significantly, one of key components of our system is View Manager. It allows each user to define her own view of ontology consisting of two elements. *View tree* is used for visualization of the ontology. While tree view is far less expressive than ontology graph, it is more intuitive and easier

to manipulate by human user. In our approach, each level of the tree corresponds to any set of ontology attributes defined by user. For example one user may prefer to group cars by make+model+body (level 1) and engine type+engine capacity (level 2), while another user may define make (level 1), model+body (level 2) and color (level 3) hierarchy. *Grid view* defines attributes of interest of user that should be acquired from Deep Web sources.

Choosing specific node in tree view corresponds to defining user query. It is composed of a set of (attribute, value) pairs selected in tree view (selection part of query) and of list of grid attributes (projection part of the query). This query is next issued to SE^3 (see next section) and extracted data are fed into grid view, that allows further data manipulations (e.g. sorting, filtering, aggregation).

6 Semantically Enhanced Extraction Engine (SE^3)

Semantically Enhanced Extraction Engine (SE^3) performs actual navigation and data extraction based on SEFST and user query. Query execution consists of two major steps: transducer minimization, and data extraction.

During the transducer minimization phase, minimal subset of original transducer such that all selection attributes are used and all projection attributes are accessible is found. For example, if the user is interested only in version, mileage and price information, the minimal transducer will not contain the state corresponding to detail pages because it is not necessary to access it. It must be noted that selection part of query may be missing some input attributes required by SEFST. In eBay example, if user's selection contains make name but does not contain model name, model name is a missing attribute. This case is handled by assigning the set of all corresponding values present in the ontology to each missing attribute. In our example, the list of all models of selected make will be provided as user input for model name.

The second step consists in performing actual navigation and extraction based on minimal transducer. At the beginning one navigation path is created containing just the page corresponding to initial state (e.g. form page in eBay Motors). Next, steps of *data extraction*, *request determination*, *path branching* and *transition* are repeated in turns. Data extraction executes rules associated with current FST state and stores extracted data attributes in current navigation path. Request determination uses FST structure to select all request templates that may be constructed in current FST state. Next, request templates, extracted data and user input are used to find all following requests (e.g. to build URLs of all pages accessible from current one). At branching stage, a new copy of current path is created for each request and put into paths queue. Then, a single path is selected from the queue, transition of FST corresponding to this path is performed and corresponding URL is accessed. The whole procedure is repeated until all paths were completely downloaded. Finally, a single, semantically annotated data record is constructed from each navigation path.

7 Conclusions and Future Work

In this paper we presented an semantic annotation approach to describe navigation and extraction from Deep Web sources, and showed how it enables personalized data access with user-defined ontology views. Our future research include attaching richer vocabulary (e.g. synonyms) to ontology instances, using annotation model presented in this paper for ontology population, and automating construction and maintenance of SEFSTs.

References

[ARP+07] Manuel Alvarez, Juan Raposo, Alberto Pan, Fidel Cacheda, Fernando Bellas, and Victor Carneiro. DeepBot: A Focused Crawler for Accessing Hidden Web Content. In *3rd intl. workshop on Data enginering issues in E-commerce and services*, pages 18–25, 2007.

[BCL05] Robert Baumgartner, Michal Ceresna, and Gerald Ledermuller. DeepWeb Navigation in Web Data Extraction. In *CIMCA '05: Proceedings of the CIMCA-IAWTIC'06 Vol. 2*, pages 698–703, Washington, DC, USA, 2005. IEEE Computer Society.

[Ber01] Michael Bergman. The Deep Web: Surfacing Hidden Value. *JEP the Journal of Electronic Publishing*, 7(1), 8 2001.

[BL98] Tim Berners-Lee. Semantic web road map. Available at http://www.w3.org/DesignIssues/Semantic.html, September 1998.

[CHZ05] Kevin Chen-Chuan Chang, Bin He, and Zhen Zhang. MetaQuerier: querying structured web sources on-the-fly. In *2005 ACM SIGMOD International Conference on Management of Data*, pages 927–929, 2005.

[HPZC07] Bin He, Mitesh Patel, Zhen Zhang, and Kevin Chen-Chuan Chang. Accessing the deep web. *Commun. ACM*, 50(5):94–101, 2007.

[HSV03] Siegfried Handschuh, Steffen Staab, and Raphael Volz. On deep annotation. pages 431–438, 2003.

[KOKA06] Marek Kowalkiewicz, Maria E. Orlowska, Tomasz Kaczmarek, and Witold Abramowicz. Robust Web Content Extraction. In *15th Intl. Conf. on WWW*, 2006.

[LRNdST02] Alberto Laender, Berthier Ribeiro-Neto, Altigran da Silva, and Juliana Teixeira. A brief survey of web data extraction tools. *SIGMOD Rec.*, 31(2):84–93, 2002.

[LYE02] Stephen W. Liddle, Sai Ho Yau, and David W. Embley. On the Automatic Extraction of Data from the Hidden Web. In *20th International Conference on Conceptual Modeling (2)*, pages 212–226, 2002.

[Nag03] Katashi Nagao. *Digital Content Annotation and Transcoding*. Artech House Publishers, 2003.

[NZC05] Alexandros Ntoulas, Petros Zerfos, and Junghoo Cho. Downloading textual hidden web content through keyword queries. In *5th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 100–109, 2005.

[RGM01] Sriram Raghavan and Hector Garcia-Molina. Crawling the Hidden Web. In *27th International Conference on Very Large Data Bases*, pages 129–138, 2001.