Using Semantics to Personalize Access to Data-Intensive Web Sources

Monika Starzecka and Adam Walczak

Poznan University of Economics, Al. Niepodleglosci 10, 61-875 Poznan, Poland
{m.starzecka,a.walczak}@kie.ue.poznan.pl
http://kie.ue.poznan.pl

Abstract. Data-intensive Web sites are an important and growing source of well-structured information on the Web. Their potential value remains largely unused as they pose a number of challenges to both machine and human users. They are dispersed and provide heterogeneous, rigid, site-specific and non-personalizable query and navigation interfaces. In this paper we present outline of method for accessing data from data-intensive Web sites in an uniform way. Our method is independent of the source and allows for personalization of the access to data. We describe how domain ontology is used for definition of personalized GUI. Then initial evaluation of described method is provided.

Keywords: Data-Intensive Web sites, Deep Web, personalization, semantics, ontologies.

1 Introduction

The contemporary World Wide Web contains unprecedented quantities of information. While basic Web technologies focus on unstructured text, huge part of Web sites (including Deep Web [7]) are *data-intensive* and contain semistructured content. Examples of such sites are on-line databases (e.g. Deep Web search engines), commercial Web sites (e.g. on-line stores or e-auctions) and Web applications (e.g. Web calendars, social networking sites). Due to size and quality of data-intensive Web sites content, they are priceless resource for both individual and organizational users.

From a user's perspective, data-intensive Web sites are quite challenging to access for several reasons. In most cases, they are dispersed and hard to find.

Web sites, even if they belong to the same domain, usually differ significantly with respect to their user interface (i.e. how data is presented to the user), navigation patterns (i.e. if navigation is form or link based) and data organization (i.e. how attributes and classes of entities are split into different types of pages). Typically navigation and data organization cannot be personalized and not necessarily correspond to user's needs and view of the domain. Advanced users build wrappers or copy the data manually to some application that allows for greater flexibility in data manipulation.

In this paper we present an approach for personalized access to data from data-intensive Web sites in an integrated and semantics-aware way.

W. Abramowicz and D. Flejter (Eds.): BIS 2009 Workshop, LNBIP 37, pp. 28–38, 2009.

2 Related Work

Research on Web data access, presentation and extraction started at the very dawn of World Wide Web, giving birth to the wrapper construction field [16]. As the Web grown, enterprises started to expose data in this environment and enable its querying. Thus Deep Web was born [5] and became a huge source of well structured Web data. There are many Deep Web systems (including [20,9,18,2]) focusing rather on accessing and indexing content than on extracting data from the sources. Merely few solutions dealing with data-intensive Web sites in a semantic way are presented. Gal et al. [23] propose a framework that supports extraction of ontologies from Web search interfaces, ranging from simple Search Engine forms to multiple-pages, complex reservation systems. OntoBuilder enables fully-automatic ontology matching, yet the solution doesn't deal with the problem of data extraction.

In [3] adding of an ontology layer to the Deep Web structure is proposed. With the use of given ontology (in provided example it is WordNet) synonyms for keywords from user's query are listed. Then, the appropriate attributes in Deep Web sources can be found, by comparison with provided keywords and listed synonyms (with given similarity rate). Similar approach is presented in [24], where the process of filling forms is automated by correlating web form labels to entries in a domain ontology. Matching the ontology concepts with appropriate web form labels is achieved through the use of a continually refined knowledge base and application of LSD system [10].

Data-intensive Web sites offer typically very simplistic and rigid query interfaces with limited querying capabilities. Such simplicity is sometimes desirable from a human point of view but comes at a cost of limited flexibility and constraints for automated querying of multiple sources. Methods for dealing with querying via interface with limited capabilities were studied e.g. in [12]. Source querying capabilities description and associated query rewriting problems were previously addressed for mediator systems working with dispersed databases (for example [19]).

So far little effort was devoted to personalizing access to the data-intensive Web sites. To certain extent all the previously mentioned work referring to wrappers or information integration touches this subject, but in fact provides just the first step - unified view on multiple sources. The second step - to provide personalized access for individual users is often neglected. We found the work of Bigham and colleagues to refer to personalized access [6], yet they too constrain themselves to extraction and navigation in the Web data sources.

3 System Overview

Building on our concept of navigating and extracting from the data-intensive Web sites using finite state machines we aim at personalizing the access to the data sources.

We adopted formal ontology as a way to describe the source and its data. The ontology is both underlying structure that is personalized according to user's

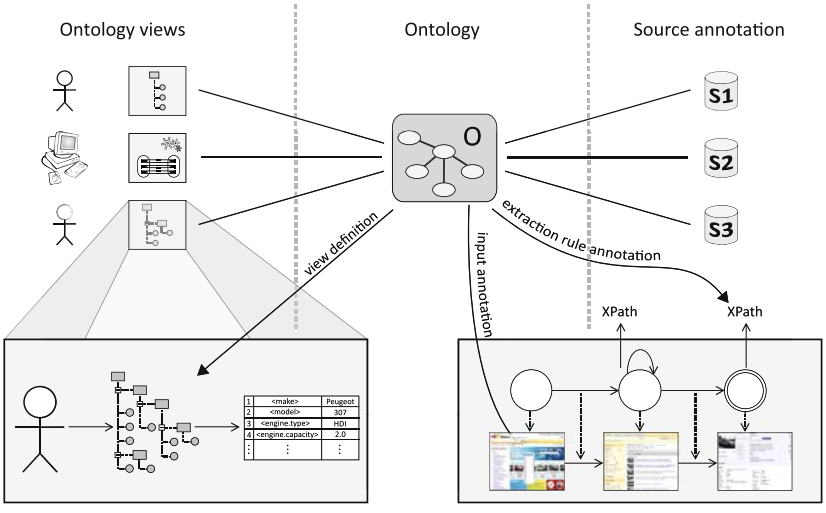

Fig. 1. Overview of System's – Conceptual Architecture

perception of the particular source, and provider of the domain data necessary for query execution. Conceptual architecture of the system shows Fig.1. The main components of our system are:

- Domain ontology - represents common domain model for all the sources considered for particular task (see Section 4).
- Views and Profile Manager - responsible for creating and managing user profiles (see Section 5).
- Source Description Manager - allows for maintenance of relations between particular source and the domain ontology e.g. referring form fields and presentation labels to the concepts in the ontology (description of site descriptions management is out of scope of this paper).
- Query planner - the component responsible for building source-specific query plans based on semantic description of individual data-intensive Web sites (detailed description of query planning is out of scope of this paper, basic idea was presented in [1]).
- Query execution engine - the component that actually navigates the data-intensive site and extracts information from relevant pages (using finite state machine model) based on the source description and its binding to the instances stored in the ontology together with user's profile (see [1]).
- Graphical User Interface - the component responsible for building profiles and queries and visualizing the retrieved data (see Section 5.2).

4 Domain Ontology

Automotive Ontology was developed from scratch for the needs of domain source description, and as a means to determine user informational needs. Its main

purpose is to provide shared conceptualization, that would enable car description standardization from the perspective of automotive Web sites. It includes all car attributes, that may be determined on the web site, or which must be provided by the user according to web sites navigational process. The system which part of we are describing in the paper is now during the stage of a prototype, therefore some simplistic assumption were made. We decided to analyze car description until the level of version definition, thus we are not handling the information concerning equipment and peripherals (such as air-conditioning, CD player, body or upholstery coloring, etc.). Essential for us is information about: make, model, body type, gearbox, engine, fuel, version and drive. Those notions constituted concepts in designed automotive ontology.

- Model - a car model is a particular brand of vehicle sold under a make. From an engineering point of view, a particular car model is usually defined and/or constrained by the use of a particular car chassis/body type combination.
- Make - a make is a brand name. For example, Chevrolet and Pontiac are marques of their maker, General Motors.
- Body type - body types are largely (though not completely) independent of a car's classification in terms of price, size and intended broad market; the same car model might be available in multiple body styles (or model ranges). Ex.: Sedan, Hardtop, Coup, Limousine
- Gearbox - gearbox provides a speed-torque conversion from a higher speed motor to a slower but more forceful output or vice-versa. For our needs we distinguish following gearbox types: manual, automatic, semi-automatic
- Engine - a car engine is a machinery in which the combustion of a fuel occurs with an oxidiser (usually air) in a combustion chamber. In a car engine the expansion of the high temperature and pressure gases (that are produced by the combustion) directly apply force to a movable component of the engine, such as the pistons or turbine blades and by moving it over a distance, generate useful mechanical energy. From our perspective, the most important attributes of an engine are: capacity, power and type (for example: TDI, CTDI, etc.)
- Fuel - fuel is any material that is burned or altered in order to obtain energy and to heat or to move an object. Fuel releases its energy through a chemical reaction means (such as combustion).
- Car Version - car version is a particular category of cars offered on the market with specific combination of attributes: model, engine, body and gearbox.

To make an ontology fully useful for determined purpose we needed to create instances of previously defined concepts. For this task we used information from data base that was facilitated to us, by our business partner from automotive industry. Current version of the ontology was developed in OWL, it has seven concepts, over a dozen of properties and approximately five hundred of instances.

5 Views and Profile Manager

It is common practice among data-intensive Web sites to provide a single, rigid user interface. Our proposal is to give the user possibility of defining his own interface independently of the one provided by source owner. Assuming that appropriate domain ontology is given, we designed a solution which enables user to create her own query interfaces and data views for data-intensive Web sites.

5.1 User Profile

In our approach, we allow user to represent her perception of the domain by selecting part of the domain ontology and saving it in *user profile*. It specifies what are the attributes typically used to access data, what is their order of importance, what combinations of attributes should be selected together to avoid too many clicks or improve readability (e.g. 'Opel Astra' - make and model together, or '2.0 TDI' for engine capacity and type) and which attributes should be retrieved in answer to a query. As a consequence, user profile defines what attributes are not important for specific user. User profile also contains the list of data-intensive Web sites that the user wants to query.

After a thorough examination of existing ontology visualization methods [22] we decided to use tree view for ontology representation in user interface. We found intuitivity and simplicity in data representation of this method advantageous. Together with numerous shortcomings pointed by researchers, this technique in few conducted tests (for example by [21,15,8,14]) proved to be more efficient in comparison with other techniques used for visualization of hierarchical structures. No particular explanation can be found for such a good tests results of this quite primitive technique. The most possible reason may be that user can find this technique intuitive, as it is similar to the way that data are represented in everyday life (for example: table of contents in a book, a list of tasks to do, etc.). For results presentation we propose to use grid view (definition of projection part of the query).

To enable personalized access to a variety of on-line data, we let the user define structures of both tree and grid. While defining tree structure, user determines the number of levels used in query definition and the list of attributes for each level. As the same user may have very different requirements in different usage scenarios, she is allowed to define multiple profiles in our prototypical implementation and switch between them when needed.

The user interface for definition of profiles is displayed on Fig.2. In this interface user may choose to apply an existing profile (by selection of a list option and pushing "OK" button) or manage her list of profiles. She may add a new empty profile ("Add Profile" button) or build a new profile starting with existing one ("Copy Profile" button). She may also modify any profile by redefining tree structure, grid structure or list of sources to be used. In this example, user has already defined the first level of the tree to contain make and model names, and is currently defining the second level.

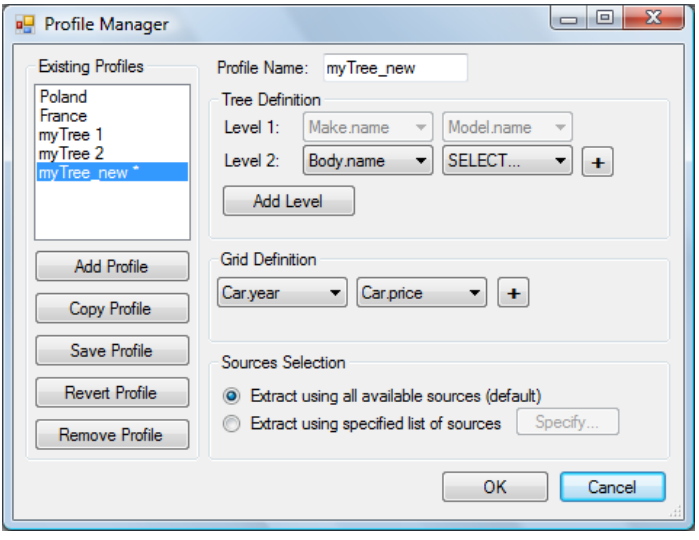

Fig. 2. Profile management and definition user interface

Common view of specific domain may reflect cultural differences between countries or organizational differences between companies. For example in automotive domain, on German sites fuel type is a very important attribute of the car, while on Polish Web sites the user is rarely asked to specify this attribute. Thus the list of profiles may also include predefined profiles for specific countries ("Poland" and "France" in Fig.2.). Predefined profiles may be good starting point for definition of new tree or grid structure.

In the remaining of this paper we assume the following definition of the user profile. The grid includes year, price, mileage, fuel type and number of bids. The tree has following three levels:

- level 1: make and model,
- level 2: body and gearbox type, and
- level 3: engine capacity.

5.2 Graphical User Interface

In our system we propose two main ways of querying data-intensive Web sites. The first approach is based on user profiles described in two previous sections and corresponds to *long-term personalized access to data-intensive Web sites*. The second approach enables dynamic, intuitive definition of user query by selecting attributes in any arbitrary order (possibly very different from Web sites navigation pattern), thus enabling *ad hoc personalized access to data-intensive Web sites*.

The user interface enabling access to data-intensive Web site in the first case is displayed in Fig. 3. The tree in this view is constructed based on tree definition in user's profile, and filled in with values acquired from domain ontology. After

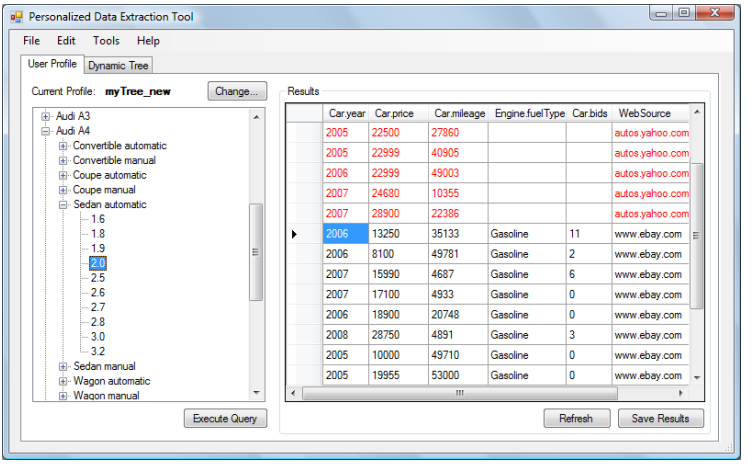

Fig. 3. Profile-based personalized view of automotive domain for profile defined in Fig.2. Data returned for specific query shown in the grid; records from sources not containing some of attributes are marked in red.

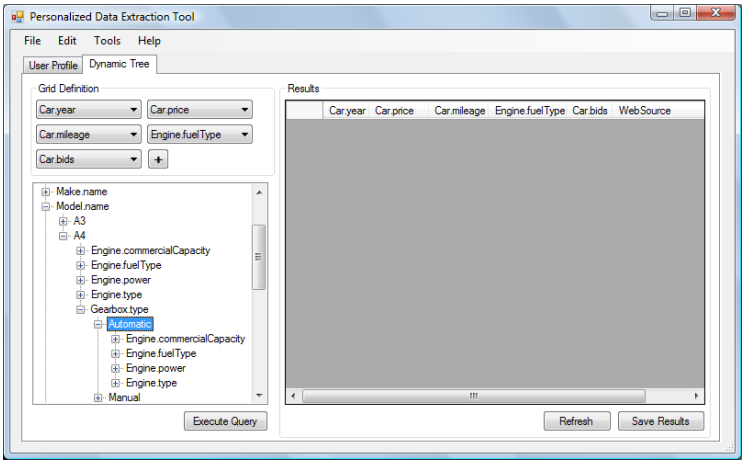

Fig. 4. Dynamic tree view of automotive domain

the user selects any item of the tree and presses "Execute" button, the query definition, rewriting and execution is performed and the result is displayed in the grid in the right part of the window. As some of data attributes may be missing or empty in some sources, some cells in the grid may remain empty.

The user interface enabling ad hoc personalization is displayed in Fig. 4. While it is similar to the previous view, few significant differences should be emphasized. Firstly, while by default the grid structure from active profile is reused, it is possible to change this structure at query time. Secondly, the contents of the tree is very different. At each odd level of the tree, user has to choose the name of

attribute that will be determined next, and at each even level she selects specific value for the attribute. The set of attributes presented in the dynamic tree is rigid and corresponds to all ontology instances. Every two steps (attribute selection and value determination) allow to determine single attribute. While, such method gives more flexibility than currently existing solutions, its usefulness might decrease when the number of used attributes is high.

6 Idea Validation

The examination described in this section was performed to verify presented idea in comparison with standard navigation in data intensive sources from a perspective of their efficiency. Assumption made during validation process was that we have a complete set of sources descriptions needed for automatic navigation through the analyzed data sources. As the efficiency measure we decided to use a number of activities that need to be performed by the user in order to reach the information she is looking for. Under the notion of activity we understand click, choice from drop-down list, radio button selection etc. as well as visual scanning of big table of results. The fewer activities need to be performed, the more efficient is navigation. The size of test set was amounted to 46 sources[1] The web pages that composed final test set were chosen from around 300 automotive data sources indicated by practitioners as reliable and useful sources of information.

Just as example in Section 5, the examination was made for automotive industry domain. It covered two scenarios of situations when user wants to find current price of particular car. First one: the user knows exactly what kind of a car she is looking for and has defined navigational tree as: 1st level: make, model, body; 2nd level: engine, fuel; 3rd level: gear box, car version. In second scenario we assumed that the user has exact preferences regarding just three attributes of the car: make, model and body. Every one of the attributes constitutes separate level in navigational tree.

Figures 5 and 6 shows results respectively for first and second scenario. Charts represents how number of required activities grows with the number of searched sources. It can be easily seen that in both cases the examination came off better for our method.

The more sources were searched the bigger difference in efficiency between analyzed navigational methods was noticed. It is so, because in standard method user needs to determine searched car attributes for every single source separately. Our proposed method after defining tree structure and single car attributes determination uses the information for every chosen source. For all 46 sources it amounted to 202 activities for the first scenario and 1012 for the second one. For second scenario the difference was significantly bigger. The reason is inflexibility

[1] The number was calculated by the equation: $n = \left(\frac{z\sigma}{e}\right)^2$, where: n - searched test set size, z - value of cumulative distribution function for normal distribution with given statistical significance(90%), σ - standard deviation estimator, e - value of permissible error (0,6).

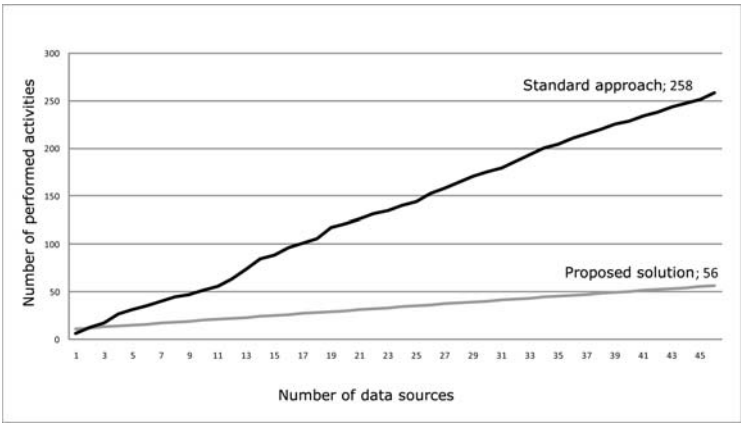

Fig. 5. Relation between number of activities required to achieve user goal and number of data sources - first scenario

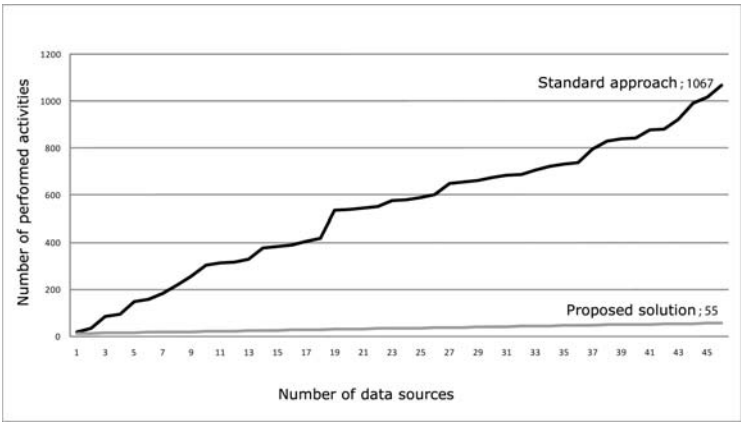

Fig. 6. Relation between number of activities required to achieve user goal and number of data sources - second scenario

of standard navigational structure provided by web sites. User has no possibility to change navigational path of the web page, and when her needs are incoherent with provided structure, then very often more activities must be performed to find needed information. In our scenario for the user fuel type was indifferent, but in some of the web pages it was second (after model) attribute to determine. In practice it generated few parallel navigational paths (one for each existing fuel type) which generated more activities necessary to perform.

Presented examination clearly shows that even if user must define tree structure and expected results in the system, while searching the same information in many different sources our method is far more efficient then the standard one. These were very simple scenarios. 38 from analyzed sources are dedicated

to only one make. On average each make has 11,36 models, every model is available with 1,46 body types. Standard navigation through the web page, when the user defined only model and body type requires on average 26,36 activities. If the user wants to check car price in all 38 sources - she would have to perform 16320 activities. Remaining 8 sources provide information about 50 makes on average. If our user would like to analyze those sources the number of performed activities would grow to more than 200000.

7 Future Work

In this paper we presented an approach that enables users to construct their personalized and uniform access interface for data-intensive Web sites in specific domain, described by an ontology. There are few more challenges that we plan to solve in our future work. Firstly, to deal with lexical and value-encoding variations, the problem of synonyms and more complex lexical relations (e.g. hyponymy or hypernymy) and data translation rules (e.g. currency or unit transformation) needs to be addressed. Secondly, while proposed simple tree interface works well for textual data attributes, support for more complex tree rules (e.g. by comparison or range operators on integer attributes) would be beneficiary. Thirdly, in order to support arbitrary-depth tree presentation of cyclic relations in domain ontologies (such as similarity between models of cars). Also, as our future work, we plan to handle the problem of emerging web technologies and navigation patterns (such as AJAX, flash).

References

1. Abramowicz, W., Flejter, D., Kaczmarek, T., Starzecka, M., Walczak, A.: Semantically Enhanced Deep Web. In: INFORMATIK 2008 Beherrschbare Systeme dank Informatik, 38. Jahrestagung der Gesellschaft für Informatik, Gesellschaft für Informatik e.V (GI), München, September, 8. bis 13, pp. 673–679 (2008)
2. Alvarez, M., Raposo, J., Pan, A., Cacheda, F., Bellas, F., Carneiro, V.: Deepbot: A focused crawler for accessing hidden web content. In: 3rd international workshop on Data engineering issues in E-commerce and services, pp. 18–25 (2007)
3. An, Y.J., Geller, J., Wu, Y.-T., Chun, S.A.: Semantic deep web: automatic attribute extraction from the deep web data sources. In: SAC 2007: Proceedings of the 2007 ACM symposium on Applied computing, pp. 1667–1672. ACM, New York (2007)
4. Anupam, V., Freire, J., Kumar, B., Lieuwen, D.: Automating web navigation with the webvcr. In: 9th International Conference on World Wide Web, pp. 503–517 (2000)
5. Bergman, M.K.: The deep web: Surfacing hidden value. The Journal of Electronic Publishing 7(1) (2001)
6. Bigham, J.P., Cavender, A.C., Kaminsky, R.S., Prince, C.M., Robinson, T.S.: Transcendence: Enabling a personal view of the deep web. In: International Conference on Intelligent User Interfaces (2008)
7. Chang, K.C.-C., He, B., Zhang, Z.: Mining semantics for large scale integration on the web: evidences, insights, and challenges. SIGKDD Exploration Newsletter 6(2), 67–76 (2004)

8. Cockburn, A., McKenzie, D.: An evaluation of cone trees. In: Proceedings of the 2000 British Computer Society Conference on Human Computer Interaction (2000)
9. Chang, K.C.-C., He, B., Zhang, Z.: Metaquerier: querying structured web sources on-the-fly. In: 2005 ACM SIGMOD International Conference on Management of Data, pp. 927–929 (2005)
10. Doan, A., Domingos, P., Halevy, A.Y.: Reconciling schemas of disparate data sources: A machine-learning approach. In: SIGMOD Conference (2001)
11. Flesca, S., Gottlob, G., Baumgartner, R.: Supervised wrapper generation with lixto. In: 27th International Conference on Very Large Data Bases, pp. 715–716 (2001)
12. Halevy, A.: Theory of answering queries using views. SIGMOD Record 29(4), 40–47 (2000)
13. Handschuh, S., Staab, S., Volz, R.: On deep annotation. In: WWW 2003: Proceedings of the 12th international conference on World Wide Web, pp. 431–438. ACM, New York (2003)
14. Katifori, A., Torou, E., Halatsis, C., Lepouras, G., Vassilakis, C.: A comparative study of four ontology visualization techniques in protege: Experiment setup and preliminary results. In: IV 2006: Proceedings of the conference on Information Visualization, Washington, DC, USA, pp. 417–423. IEEE Computer Society, Los Alamitos (2006)
15. Kobsa, A.: User experiments with tree visualization systems. In: INFOVIS 2004: Proceedings of the IEEE Symposium on Information Visualization, Washington, DC, USA, pp. 9–16. IEEE Computer Society, Los Alamitos (2004)
16. Teixeira, J.S., Ribeiro-Neto, B.A., Laender, A.H.F., da Silva, A.S.: A brief survey of web data extraction tools. SIGMOD Record 31(2), 84–93 (2002)
17. Nagao, K.: Digital Content Annotation and Transcoding. Artech House Publishers, Norwood (2003)
18. Ntoulas, A., Zerfos, P., Cho, J.: Downloading textual hidden web content through keyword queries. In: 5th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 100–109 (2005)
19. Papakonstantinou, Y., Gupta, A., Haas, L.: Capabilities-based query rewriting in mediator systems. In: 4th International Conference on Parallel and Distributed Information Systems (1996)
20. Raghavan, S., Garcia-Molina, H.: Crawling the hidden web. In: 27th International Conference on Very Large Data Bases, pp. 129–138 (2001)
21. Rivandeneira, W., Benderson, B.B.: A study of search result clustering interfaces: Comparing textual and zoomable interfaces. Technical report, University of Maryland HCIL (2003)
22. Starzecka, M.: Nawigacja w serwisach www na podstawie ontologicznego opisu zrode. Master's thesis. Akademia Ekonomiczna w Poznaniu (2008)
23. Gal, A., Modica, G., Jamil, H.: OntoBuilder: Fully Automatic Extraction and Consolidation of Ontologies from Web Sources. In: International Conference on Data Engineering. IEEE Computer Society, Los Alamitos (1996)
24. Walny, J.: Semaform: Semantic wrapper generation for querying deep web data sources. CPSC 502 project under the supervision of Dr. Denilson Barbosa (2007), http://www.ucalgary.ca/~jkwalny/502/index.html