Conceptual Views over the Web

Tiziana Catarci, Luca Iocchi, Daniele Nardi, Giuseppe Santucci
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
{catarci,iocchi,nardi,santucci}@dis.uniroma1.it

Abstract

The Internet has made available an enormous quantity of information to a disparate variety of people. The amount of information, the typical access modality (i.e. browsing), and the open growth of the Net, force the puzzled user to search the information of interest in a labyrinth of billions of links. This is very different from traditional database querying, where the user has just the duty of specifying which are the data s/he wants to retrieve. We argue that, in order to provide the user with a powerful and friendly query mechanism for accessing information on the Web, the critical problem is to find effective ways to build models of the information of interest.

In this paper we motivate the above observation by presenting notable attempts to construct systems which model the information in the Web following different approaches. We first classify such systems in two categories, as being based on Database or Knowledge Representation techniques, and discuss their main advantages and disadvantages. Then, we briefly introduce an integrated approach, in which Knowledge Representation mechanisms and reasoning capabilities are coupled with traditional Database features, such as efficient data management and query optimization.

Proceedings of the 4th KRDB Workshop
Athens, Greece, 30-August-1997

(F. Baader, M.A. Jeusfeld, W. Nutt, eds.)

http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-8/

1 Introduction

The growth of the Internet has dramatically changed the way in which information is managed and accessed. We are moving from a world in which the information management was in the hand of a few devotees to a widespread diffused information consumption. Along with the excitement, there is also the recognition of the undeferring need for effective and efficient tools for information consumers, who must be able to easily locate, in the Web, disparate information, ranging from unstructured documents and pictures to structured, record-oriented, data. When doing this, one cannot just ask for the information s/he is interested in, instead one has to search for it. If the information happens to be found, it is scattered everywhere, in a piecemeal fashion. An appealing alternative is to allow the user to place a query and get a single integrated result. We argue that in order to provide the user with a powerful and friendly query mechanism for accessing information on the Web the critical problem is to find effective ways to build models of the information of interest.

The present paper aims at motivating the above observation, by discussing notable attempts to construct systems following this information modeling approach, here called Global Information Management Systems (GIMSs). The main goal of such systems is to provide a framework to integrate different and heterogeneous information sources into a common domain model. An information source can be an on-line database/knowledge base accessible through the Web or a simple HTML page or a plain text file. Information units are individual elements of information coming from information sources. The user interacts with the GIMS as a whole information system, so that s/he can ignore the data schema used in the sources and access information using a query-answering mechanism.

Popular keyword-based search engines can be regarded as first generation GIMSs that use feature-based representations (or keyword representations), modeling documents through feature vectors. Such representations make it easy to automatically classify documents, but offer limited capabilities for retrieving the information of interest, still burying the user under a heap of disomogeneous information.

In order to overcome such limitations, GIMSs need more sophisticated methods for representing information sources. Such methods can be roughly classified as being based on Database or Knowledge Representation techniques. The differences between the two approaches can be characterized by using a terminology that is borrowed from the literature on Data Warehouse. In a Database perspective the goal is to build a fully materialized Data Warehouse of the information in the Web. Conversely, Knowledge Representation based methods provide a solution that is close, in principle, to the idea of fully virtual Data Warehouse, in that non specific information is recorded locally, while the ability to answer queries relies on methods for dynamically accessing the information sources. In the following, we briefly compare the two approaches and show their main advantages and disadvantages.

A fully materialized approach carries a number of advantages in the ease and effectiveness of access once the data is stored in the database. However, it raises a number of issues that basically amount to the problem of database construction and maintenance. The need for modeling information is addressed by the construction of a conceptual schema of the information domain and by the realization of procedures for extracting the information to be stored based on the conceptual schema. We shall see that either there are strong assumptions on the structure and organization of the information sources or the issue is left to (user-made) ad hoc solutions. Another question that needs to be addressed in this kind of frameworks is that the dynamics of the information sources raises a maintenance problem for the materialized data.

On the other hand, a fully virtual approach is better suited to cope with the dynamics of the information sources, while it can be problematic with respect to the response time. In a knowledge based approach the idea is that the system handles an explicit representation of the information sources, which is used at query time to access the suitable information sources. The various proposals differ in their capabilities to handle lack of structure, incompleteness and inaccuracy of the information sources.

2 Database approaches

In the Database community we find two kinds of proposals: systems to integrate different information sources and declarative languages to query the Web.

In the first case the idea is to regard the WEB as a federation of databases. With the difference that database federations typically rely on the presence of a schema describing the sources and on highly structured data, while Web documents are usually unstructured or semi-structured.

One example of this first approach is Tsimmis [CGMH+94], which describes the common schema with the OEM (Object Exchange Model) language. The associated query language, OEM-QL, is an SQL-like language. Tsimmis makes use of *translators* to translate both data objects into a common information model and queries into requests for an information source, while *mediators* embed the knowledge needed for processing a specific type of information, once the content of information sources is known. Each mediator needs to know which sources it will use to retrieve information. Therefore, a model of information sources has to be explicitly specified, but it is possible to work without a global database schema. Classifiers and extractors can be used to extract information from unstructured documents (e.g. plain text files, mail messages, etc.) and classify them in terms of the domain model. The classifier/extractor components of Tsimmis is based on the Rufus system [SLS+93]. Rufus uses an object oriented database to store descriptive information about user's data, and a full text retrieval database to provide access to the textual content of data.

Another proposal along these lines is constituted by the ARANEUS Project [AMM97], whose aim is to make explicit the schema according to which the data are organized in so-called structured servers, and then use this schema to pose queries in a high level language instead of browsing the data. Even though the ability to construct structured description of the information in the Web enables the system to answer effectively user queries, the approach has the following drawbacks that are typical of a Database perspective: 1) Araneus works only on a particular kind of Web sites and pages, which have a clearly specified structure, not on generic ones; 2) the user has to completely specify the relational schema corresponding to the site data; there is no automatic translation from the site to the database; 3) there is no hint for automatic search and conceptualization of WWW sites similar to prototypical ones indicated by the user.

A second research line that is worth mentioning involves the development of declarative languages to query the Web. Note that this approach is weakly related with the idea of modeling the information stored in the Web sites. The main idea is to model the Web document network topology (in particular, in [MMM96] a "virtual graph" is used to represent the hypertextual documents in the Web), and to provide the user with a query language enriched with primitives for specifying query conditions on both the structure of single documents and their locality on the network. However, the user has no chance to query the Web information content. Relevant Web query languages proposed in literature are W3QL [KS95], WebLog [LSS96], WebSQL [MMM96].

As we said above, systems using a Web query language do not maintain a global model of the application domain, instead they allow the user to interact in a transparent way with Web search engines or indexes built from robots. Many of the problems one encounters using indexes, such as information changes or lack of representation of document structures, are not addressed in these systems. However, the possibility of capturing the structure of a hypermedia network, explicitly describing links between documents, and the introduction of the "query locality" concept to measure the cost to answer a query, are important elements, that need to be taken into account in the development of effective and efficient systems.

3 Knowledge-based approaches

Knowledge-based GIMSs are systems using a Knowledge Representation approach for information source representation, data acquisition and query processing. Many logical frameworks are used to represent information and reason about them.

The main design element for these systems is the Knowledge Representation language. Also relevant are automatic data acquisition techniques, that are useful to build and update knowledge bases, as well as query-planning techniques, adopted to answer user queries.

As for the Knowledge Representation language and data acquisition aspects, let us remark that a GIMS needs to represent both the application domain and the content of the information sources. Usually a single Knowledge Representation language is adopted. One typical example is constituted by Description Logic, which is suited to represent taxonomic knowledge.

In addition, a basic feature for a GIMS is the possibility of identifying interesting information sources unknown to the user and to automatically gather from them relevant information units. In other words, tools to scale up with the growth of the information space are needed.

The discovery of new information sources, the extraction of information units within them and the interpretation of data coming from these sources are all problems related to information acquisition. This issue is rarely addressed in most systems, as they force the user to hand-code information source models. The main exceptions are ShopBot and ILA [PDEW96]. ShopBot addresses the extraction problem learning how to access an on-line catalog (via an HTML form) and how to extract information about products. It uses an unsupervised learning algorithm with a small training set. Whereas ILA (Internet Learning Agent) is focused on the interpretation problem. It learns how to translate information source output into the domain model, using a set of descriptions of objects in the world.

It is worth noting that, especially when dealing with the automatic discovery and integration of information sources, the vocabulary problem is one of the most critical ones. The presence of possibly different terms representing the same concept in the description of a source or an information unit is a significant example. At least three possibilities have been explored to face this problem: 1) unique vocabulary, that is forcing the description of information sources and domain model to share the same vocabulary; 2) a manual mapping, that is relationships between similar concepts are hand-coded; 3) automatic (or semi-automatic) mapping, in which the system takes advantage of existing ontologies that provide synonym, hypernym and hyponym relationships between terms. The use of hypernym and hyponym relationships is a powerful tool to solve questions about the terminology, but involves loss of information when generalizations of terms are used.

As for query answering, a significant body of work on agents able to reason and make plans has been developed. In this case, the representation of the information sources is known to the system. The use of planning techniques to retrieve information requested by a user query has been very common in this context and is in general aimed at introducing a certain degree of flexibility in exploring the information sources and extracting information from them.

For instance, in Information Manifold [LRO96] the content of information sources is described by query expressions that are used to determine precisely which sources are needed to answer the query. The planning algorithm first computes information sources relevant to each subgoal, next conjunctive plans are constructed so that the soundness and completeness of information retrieval and the minimization of the number of information sources to be accessed are guaranteed. In this system, interleaving planning and execution is a useful way to obtain information for reducing the cost of the query during plan execution.

SIMS [AKS96] defines operators for query reformulation and uses them to select relevant sources and to integrate available information to satisfy the query. Since source selection is integrated into the planning system, SIMS can use information about resource availability and access costs to minimize the overall cost of a query.

A final note is on the closed world assumption adopted by all the above systems. That is, they work on the assumption that the domain model contains all information needed and that all unavailable information does not exist. On the contrary Internet Softbot [EW94] provides a framework to reason with incomplete information, executing sensing actions to provide forms of local closure, i.e., to verify the actual presence of information in the source during plan execution.

4 Web-At-a-Glance (WAG)

The work briefly surveyed in the previous sections shows the efforts that have been separately made by both the database and the knowledge representation communities to find effective ways to model the information contained in the Web. However, there are still many open problems and signals that let us favour an integrated approach, trying to combine the different contributions and harmonize the contrasts. This is the idea we are following in the WAG (Web-At-a-Glance) project [CCNS97], by coupling a database conceptual model (namely the Graph Model [CSA93, CSC97]) and its environment to interact with the user [CCC+96] with the CLASSIC knowledge representation system [BBMA89], in a system aiming to semi-automatically build conceptual views over information extracted from various Web sites and to allow the user to query such views.

The main differences with other Database approaches (e.g. Tsimmis and ARANEUS) are the following.

1. Instead of requiring an explicit description of the sources, WAG attempts to semi-automatically classify the information gathered from various sites based on the conceptual model of the domain of interest.

2. The result of such a classification is fully materialized. The idea is that a certain degree of inaccuracy in the classification is acceptable in "unknown" domains. In order to make feasible the site conceptualization, a set of tools are available to match the information acquired visiting new sites with the domain model owned by the system.

3. WAG provides a visual interface to query the databases (each one related with a specific domain or sub-domain) resulting from the integration of the information extracted from the various sites.

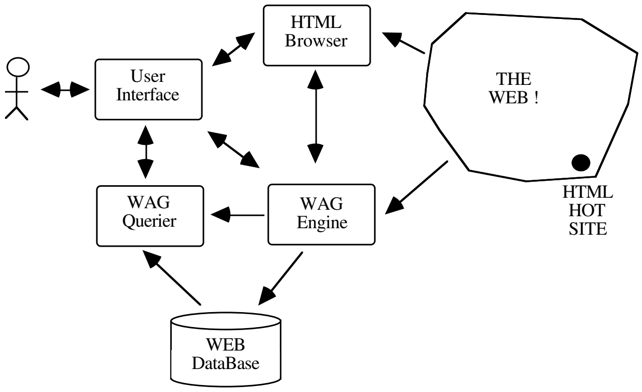

Figure 1: *The System Architecture*

In this section we present the architecture of WAG. In Fig. 1 the main components of the system are shown. WAG has a highly modular architecture, in which several components cooperate to accomplish the task. The user interacts with the user interface that allows for switching among a conventional HTML browser, a WAG querier, and the WAG engine. Each time the user meets a site containing pieces of information about a relevant matter s/he can activate the WAG engine in order to analyze it.

The WAG engine reaches the site pointed out by the user and collects the HTML pages belonging to it. Once the site pages are locally available, the WAG Engine starts its analysis. In doing that, some additional information on the domain of interest is needed; it is provided either by the system knowledge base or by the user, through an interactive session. In the latter case, the pieces of information gathered by the user are added to the knowledge base for further reuse. The main objective of the analysis process is to associate with the site under observation a conceptual data schema and to populate it. The results of such a process, that may again involve the user in an interactive fashion, are stored in the WEB DataBase. More precisely, the WEB DataBase contains both the data and the locations in which such data are available (e.g., the page URL, the page paragraph, etc.).

Once the site has been fully analyzed, the user is provided with a new powerful way to access the information stored in the site itself: s/he can query the WEB through the WAG Querier, according to several query modalities provided by the system. The WAG Querier handles all the phases needed to answer a user query: query formulation, query distribution, and query result presentation. In particular, it provides the user with a multiparadigmatic visual environment (see [CCC+96]), equipped with different visualizations and interaction mechanisms, including a synchronized browser on the database schema and instances and several ad-hoc features for interacting with multimedia data.

Below we focus on the two main submodules of the WAG engine: The *Page Classifier* and the *Conceptualizer*.

The Page Classifier analyzes the structure of the pages and classifies them as belonging to certain predefined categories. The categories are differentiated based on the various contribution they can give to the subsequent conceptualization phase, e.g. the home page of an individual will become an instance of some class, while the index page of a University will be transformed into a conceptual subschema; a page containing a form will provide a sketch of the underlying relational database, etc.

The Conceptualizer is the core of the system. It builds a conceptual schema from the HTML pages of a certain site, and then populates the schema with different kinds of instances (e.g., URL, tuples, multimedia objects, etc.) extracted from the site.

The Conceptualizer relies on two formal models, which are strictly related. The first one is an object-based data model, the Graph Model. It has two important features: 1) it is semantically rich; 2) it is equipped with a relationally-complete set of graphical query primitives, which have been already used as formal basis for a multiparadigmatic visual interface [CCC+96]. A Graph Model DataBase (GMDB) is a triple $< g, c, m >$, constituted by: 1) an intensional part, comprising the schema of the database, the so-called *Typed Graph* g, and a set of *constraints* c, which includes the standard ISA and cardinality constraints, and 2) an extensional part, i.e., the instances of the database, represented by m, which is called *Interpretation.*

As for the second model, the main need is to have schemata modeling the information of the domains of interest, plus knowledge representation mechanisms and reasoning services to support the construction and maintenance of such schemata. In WAG, we choose to express the information content of the various domains in a knowledge representation formalism of the family of Description Logics. The formalism is equipped with special reasoning capabilities (for example to detect containment relations between classes, or to classify new classes with respect to a set of existing ones) and has a strict relationship with semantic data models [CLN94, CSC95].

In particular, we have chosen to represent the system generic ontology using the knowledge representation system Classic [BBMA89], while the user's views on the various domains are represented using the Graph Model structures, namely as pairs $< g, c >$. The idea is to use a restricted representation to reason efficiently, while adopting a richer framework for modeling the data, thus providing the user with a suitable model for the interaction with the system.

While analyzing and structuring the site information, the Conceptualizer executes three main activities:

a) Site Structure Discovery, which results into a partial conceptual schema for the site;

b) Schema Definition, which matches the site conceptual schema against the description of the domain knowledge (which is either already part of the system ontology or is built up with the help of user's suggestions) in order to build a complete conceptual schema;

c) Schema Population, whose task is to populate the data base according to the conceptual schema resulting from the two phases above.

Finally, it merges the various local schemata exploiting additional pieces of information coming from the knowledge base, and produces a partial integrated schema. The basic ideas for the schema integration process (even if in a different context) have been already presented in [CSC95].

5 Conclusions

It is our opinion that the most important and difficult problem to be solved when aiming at a user-friendly access to the information in the Web is to find effective ways to model this information.

Efforts to provide tools to support this kind of modeling have been made from both a Database and a Knowledge Representation perspective. However, one can realize that there is still a gap between the model chosen and the possibility to (semi)-automatically construct schemata in such a model which represent and structure the information extracted from the Web. In fact, either the model is too poor to build schemata which are really effective for retrieving the data of interest or such schemata need to be fully hand-coded.

We address the modeling problem by proposing an integrated approach which relies on a conceptual modeling language equipped with a powerful visual environment and on a knowledge representation tool which is meant to provide a simpler representation of the information, but the ability to reason about it.

References

[AKS96] Y. Arens, C. A. Knoblock, and W. Shen. Query reformulation for dynamic information integration. *Journal of Intelligent Information Systems*, 1996.

[AMM97] P. Atzeni, G. Mecca, and P. Merialdo. Design and maintenance of data-intensive web sites. Technical Report 25, Dipartimento di Informatica e Automazione, Università di Roma Tre, 1997.

[BBMA89] Alexander Borgida, Ronald J. Brachman, Deborah L. McGuinness, and Lori Alperin Resnick. CLASSIC: A structural data model for objects. In *Proc. ACM-SIGMOD Conference on the Management of Data*, pages 59–67, 1989.

[CCC+96] T. Catarci, S. K. Chang, M. F. Costabile, S. Levialdi, and G. Santucci. A graph-based framework for multiparadigmatic visual access to databases. *IEEE Transactions on Software Engineering*, 8(3):455–475, 1996.

[CCNS97] T. Catarci, S. K. Chang, D. Nardi, and G. Santucci. Wag: Web-at-a-glance. Technical Report 03-97, Dipartimento di Informatica e Sistemistica, Università "La Sapienza" di Roma, 1997.

[CGMH+94] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The TSIMMIS Project: Integration of Heterogeneous Information Sources. In *Proc. of IPSJ Conference*, pages 7–18, 1994.

[CLN94] Diego Calvanese, Maurizio Lenzerini, and Daniele Nardi. A unified framework for class based representation formalisms. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proceedings of the Fourth International Conference on the Principles of Knowledge Representation and Reasoning (KR-94)*, pages 109–120, Bonn, 1994. Morgan Kaufmann, Los Altos.

[CSA93] T. Catarci, G. Santucci, and M. Angelaccio. Fundamental graphical primitives for visual query languages. *Information Systems*, 18(2):75–98, 1993.

[CSC95] T. Catarci, G. Santucci, and J. Cardiff. Knowledge-based schema integration in a heterogeneous environment. In *Proc. of the Second International Workshop on Next Generation Information Technologies and Systems (NGITS95)*, 1995.

[CSC97] T. Catarci, G. Santucci, and J. Cardiff. Graphical interaction with heterogeneous databases. *VLDB Journal*, 6(2):97–120, 1997.

[EW94] Oren Etzioni and Daniel Weld. A Softbot-Based Interface to the Internet. *CACM*, 37(7), 1994.

[KS95] D. Konopnicki and O. Shmueli. W3QS: A query system for the World Wide Web. In *Proceedings of the Twentyfirst International Conference on Very Large Data Bases (VLDB-95)*, pages 54–65, 1995.

[LRO96] A. Y. Levy, A. Rajaraman, and J. J. Ordille. Querying Heterogeneous Information Sources Using Source Descriptions. In *Proc. of 22nd International Conference on Very Large Databases (VLDB-96)*, 1996.

[LSS96] L. Lakshmanan, F. Sadri, and I. N. Subramanian. A declarative language for querying and resctructuring the Web. In *Proc. of 6th International Workshop on Research Issue in Data Engineering (RIDE-96)*, 1996.

[MMM96] A. Mendelzon, G. A. Mihaila, and T. Milo. Querying the World Wide Web. In *Proc. of PDIS'96*, 1996.

[PDEW96] Mike Perkowitz, Robert B. Doorebons, Oren Etzioni, and Daniel S. Weld. Learning to understand information on the Internet: an example-based approach. *Journal of Intelligent Information Systems*, 1996.

[SLS+93] K. Shoens, A. Luniewski, P. Schwarz, J. Stamos, and J. Thomas. The Rufus system: Information organization for semi-structured data. In *Proceedings of the Nineteenth International Conference on Very Large Data Bases (VLDB-93)*, 1993.