User-oriented smart-cache for the Web: What You Seek is What You Get![*]

Zoé Lacroix
Institute for Research in Cognitive Science
University of Pennsylvania
lacroix@saul.cis.upenn.edu

Arnaud Sahuguet
Computer and Information Science
University of Pennsylvania
sahuguet@saul.cis.upenn.edu

Raman Chandrasekar
Institute for Research in Cognitive Science
& Center for the Advanced Study of India
University of Pennsylvania
mickeyc@linc.cis.upenn.edu

http://www.cis.upenn.edu/~AKIRA

Abstract

Standard database approaches to querying information on the Web focus on the source(s) and provide a query language based on a given predefined organization (schema) of the data: this is the *source-driven* approach. However, can the Web be seen as a standard database? There is no super-user in charge of monitoring the source(s) (the data is constantly updated), there is no homogeneous structure (no common explicit structure thus), the Web itself never stops growing, etc. For these reasons, we believe that the source-driven standard approach is not suitable to the Web.

As an alternative, we propose a *user-oriented* approach based on the idea that the schema is a posteriori expressed by the user's needs when asking a query. Given a user query, AKIRA (Agentive Knowledge-based Information Retrieval Architecture) [6] extracts a *target structure* (structure expressed in the query) and uses standard information retrieval and filtering techniques to access potentially relevant documents.

The user-oriented paradigm means that the structure through which the data is viewed does not come from the source but is extracted from the user query. When a user asks a query, the relevant information is retrieved from the Web and stored as is in a cache. Then the information is extracted from the raw data using computational linguistic techniques. The AKIRA cache (*smart-cache*) represents these extracted layers of meta-information on top of the raw data. The smart-cache is an object-oriented database whose schema is inferred from the user's target structure. It is designed on demand through a library of concepts that can be assembled together to match concepts and meta-concepts required in the user's query. The smart cache can be seen as a *view* of the Web.

To the best of our knowledge, AKIRA is the only system that uses information retrieval and extraction integrated with database techniques to provide maximum flexibility to the user and offer transparent access to the content of Web documents.

1 The AKIRA story

Three Characters

Our **User** wants to query the Web in a flexible and transparent way. He thus expects the system to provide information retrieval, information extraction and data manipulation. The **Web**, the most heterogeneous network one may think of, consists of *highly structured* sources (databases), providing a query language (wrapper), as well as *poorly structured* sources (HTML pages). The system **AKIRA** is user-oriented, agent-based and Web-aware.

The Plot

Our user wants information about upcoming conferences such as query **Q1:** *"Conferences with a submission deadline after July 31, 1998?"* or query **Q2:** *"Conferences located in the USA?"* Query **Q1** can be formulated by:

```
select c.name
from   c in Conference
where  c.submission_deadline.month > 7 and
       c.submission_deadline.year = 1998
```

When asking **Q1**, the user expresses a *target structure* (see Figure 1) composed of:

- concepts `Conference` and `Date`
- meta-concept `submission_deadline`
- attributes *name*, *month* and *year*

[*] This work was partially supported by NSF STC grant SBR-8920230, ARO grant DAAH04-95-I-0169, ARO grant DAAH04-93-G0129 and ARPA grant N00014-94-1-1086.

SIGMOD '98 Seattle, WA, USA
© 1998 ACM 0-89791-995-5/98/006...$5.00

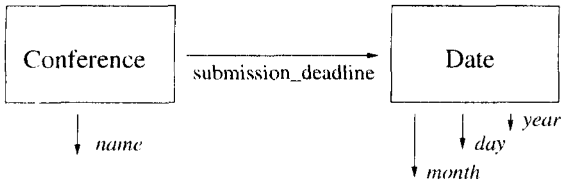

Figure 1: Target structure.

A sad story...
With usual tools, our user will browse "by hand" all pages (or send the query *"Call for papers"* to a search engine such as Altavista in order to have a list of thousands of potentially relevant webpages about conferences), read them and extract, still "by hand", the relevant information in the content.

... with a happy ending
In AKIRA, the *target structure* consisting of the classes and attributes invoked in the query is extracted and sent to the **View Factory**. The target structure is first analyzed to determine a suitable schema to represent information in the smart-cache. The system is user-oriented in the sense that the cache is structured with respect to the given target structure, and optionally completed by several classes in order to call the right Information Extraction (IE) tools to populate the database. The View Factory sends three queries: (1) a view definition creates the new classes and attributes, (2) a query is sent to the database in order to populate it, and (3) an Information Retrieval (IR) query is sent to the **IR** component in order to retrieve data from the Web.

To populate class `Conference` with attributes concerning the submission process, the system retrieves Calls for Papers (CFP's) of conferences. Retrieved documents are stored in a file before being processed by the **IE** component, invoked by methods from the database. They essentially consist of HTML textual documents often providing implicit structure such as zones "Important Dates", "Submission of Papers", etc. Extracted information populates the cache and the output of the OQL query is returned to the user.

Our user may refine his query by asking **Q2**. The query is processed as before but the structure of the cache is now extended to a new class `Location` with attributes *city*, *state* and *country* and to a new attribute `location` defined at class `Conference`.

2 Architecture

The AKIRA system can be viewed as a personal proxy that provides the user with transparent access to the Web. The input to AKIRA is provided through a standard HTML form or through a parameterized URL while the output is an HTML page generated on-the-fly by the system. These "virtual" pages, similar to the virtual documents in [7], can be bookmarked and reconstructed on-demand.

The AKIRA system basically receives a query, creates an object-oriented database (a Web view), and returns the output of the query against the instance of the database. It has five components: the **Dispatcher**, the **DBMS** (DataBase Management System), the **View Factory**, the **IE** and the **IR** agent pools, as illustrated in Figure 2.

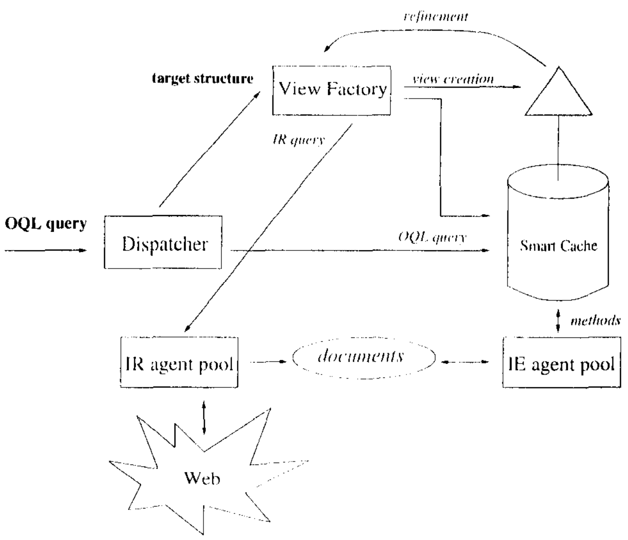

Figure 2: AKIRA's architecture.

The **Dispatcher** has a role similar to the one of a query processor for a database management system. It extracts the *target structure* from the user's query. The target structures a user may express correspond to the ability of IR agents to retrieve relevant documents and IE agents to extract expected information.

The **View Factory** is an essential part of the system. It is worth noting that AKIRA does not assume that it can start from a database representation (schema and instances) of the Web like many other systems dealing with site-restructuring (see for instance [3, 1, 4]). No information is preprocessed. The View Factory's task is to populate the cache with information extracted from documents retrieved from the Web by the Information Retrieval (IR) agents.

The **Database System** (DBMS) storing the expected Web view is object-oriented. Its associated query language OQL permits quick access to relations between objects, along attributes and methods. The Web view is defined with a view expression sent by the View Factory which specifies its schema as well as its population (through methods invoking IE agents).

The **IR** agent pool consists of wrappers to correspond with databases available on the Web (search engines or services), and information filtering tools such as Glean [2].

The **IE** component provides different IE tools extracting *concepts* and *meta-concepts*. IE agents such as conference acronym, location recognizers together with a coreference tool identify *concept* instances. *SuperTagging* [5], which provides rich syntactic labels, or zoning tools extract *meta-concepts*.

An **Output Formatter**, not shown in Figure 2, may be used to format the output according to the user's needs.

3 Conclusion

Using AKIRA to seek information on the Web provides the following benefits.

1. Based on integrated techniques from natural language processing, it enables access to explicit as well as implicit structure of textual content.

2. The separation between the logical view (concept and

meta-concepts) of Web documents and its storage in the smart-cache presents several advantages. The user-oriented approach (structure defined by user) does not require the integration of several heterogeneous sources in a global common representation. The use of a database with its query language provides a framework for optimization techniques.

3. AKIRA offers a transparent and unified architecture to access data of various media from the most loosely structured sources (newswire, press release, personal homepages or newsgroups) to highly structured sources (legacy databases, catalogues, digital libraries).

References

[1] G. Arocena and A. Mendelzon. WebOQL: Restructuring Documents, Databases and Webs. In *Proceedings of the International Conference on Data Engineering*, Orlando, February 1998.

[2] R. Chandrasekar and B. Srinivas. Using Syntactic Information in Document Filtering: A Comparative Study of Part-of-speech Tagging and Supertagging. In *In Proceedings of RIAO'97*, Montreal, June 1997.

[3] M. Fernandez, D. Florescu, A. Levy, and D. Suciu. A Query Language and Processor for a Web-Site Management System. In *ACM SIGMOD Workshop on Management of Semistructured Data*, Tucson, Arizona, May 1997.

[4] R. Goldman and J. Widom. DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. In *Proc. of Intl. Conf. on Very Large Data Bases*, Delphi, Greece, August 1997. to appear.

[5] A.K. Joshi and B. Srinivas. Disambiguation of Super Parts of Speech (or Supertags): Almost Parsing. In *Proceedings of the* 17th *International Conference on Computational Linguistics (COLING '94)*, Kyoto, Japan, August 1994.

[6] Z. Lacroix, A. Sahuguet, and R. Chandrasekar. Information Extraction & Database techniques: a user-oriented approach to query the Web. In *10th Conference on Advanced Information Systems Engineering (CAiSE'98)*, Pisa, Italy, June 1998.

[7] A-M. Vercoustre, J. Dell'Oro, and B. Hills. Reuse of Information through virtual documents. In *Proceedings of the 2nd Australian Document Computing Symposium*, Melbourne, Australia, April 1997.