Personal Information Management with SEMEX

Yuhan Cai, Xin Luna Dong, Alon Halevy, Jing Michelle Liu, and Jayant Madhavan
University of Washington
Seattle, WA 98195
{yuhancai, lunadong, alon, liujing, jayant}@cs.washington.edu

1. INTRODUCTION

The explosion of information available in digital form has made search a hot research topic for the Information Management Community. While most of the research on search is focused on the WWW, individual computer users have developed their own vast collections of data on their desktops, and these collections are in critical need for good search and query tools. The problem is exacerbated by the proliferation of varied electronic devices (laptops, PDAs, cellphones) that are at our disposal, which often hold subsets or variations of our data. In fact, several recent venues have noted Personal Information Management (PIM) as an area of growing interest to the data management community [1, 8, 6].

We are building the SEMEX System (short for SEMantic EXplorer) that offers users a flexible platform for personal information management. The current focus of SEMEX is on desktop search: instead of searching through directories or performing keyword search, SEMEX offers *search-by-association*, thereby taking a step towards the vision of the Personal Memex [3].

In particular, one of the key impediments to building flexible PIM tools and services is the mismatch between the current organization of data and the organization that is required in order to naturally support services. Today data is stored *by application* and in static directory hierarchies. Our mind, as pointed out in formulating the Memex vision, works by following *associations* between objects. As a simple example of the mismatch, information about people is scattered across our email, address book, and text and presentation files. Even answering a simple query, such as finding all of one's co-authors, requires significant work.

To enable browsing by association, SEMEX constructs a database of objects and associations between them. The database is created *automatically* from information extracted from multiple types of data sources. In effect, SEMEX provides a single *logical view* of one's personal information, based on *meaningful* objects and associations. For example, users of SEMEX can browse their personal information by objects such as Person, Publication and Message and associations such as AuthoredBy, Cites and AttachedTo. In addition to supporting browsing by association, the logical view of one's personal information can also be used to support several PIM services across multiple devices and for easily integrating additional external sources that enrich one's personal information space.

The next section describes the architecture of SEMEX, and Section 3 describes some of the over-arching issues we have encountered as we started working on personal information management. For a more detailed description of the system and the related work, the reader is referred to [4].

SIGMOD 2005 June 14-16, 2005, Baltimore, Maryland, USA.

2. SEMEX ARCHITECTURE

The components of SEMEX are shown in Figure 1. SEMEX provides access to data stored in multiple applications and sources, such as emails and address book contacts, pages in the user's web cache, documents (e.g., Latex and Bibtex, PDF, Word, and Powerpoint) in the user's personal or shared file directory, and data in more structured sources (e.g., spreadsheets and databases). SEMEX creates the data repository of objects and associations using a collection of object-and-association extraction tools. The objects are processed to reconcile multiple references to the same object. This information is accessed through a domain model using an interface that supports a combination of browsing and querying. We now briefly explain each of these components in more detail.

Domain model: SEMEX users and applications interact with the system through a domain model of personal information. The domain model includes a set of classes such as Person, Publication and Message, and associations such as AuthorOf, Cites, Sender, MentionedIn. At the moment SEMEX uses a simple data model of classes and associations, but there is a clear need for supporting subclasses and sub-associations (e.g., AuthorOf is a sub-association of MentionedIn). In a sense, the domain ontology of SEMEX can be viewed as a *mediated schema* over the set of personal information sources. Clearly, one of the important features of a PIM system is that users be able to personalize their domain models. While SEMEX comes with a generic domain model, we are also considering several ways of manually personalizing it.

Associations and instances: The key architectural premise in SEMEX is that it should support a variety of mechanisms for obtaining object and association instances. The main ones are as below.

- *Simple:* In many cases, objects and associations are

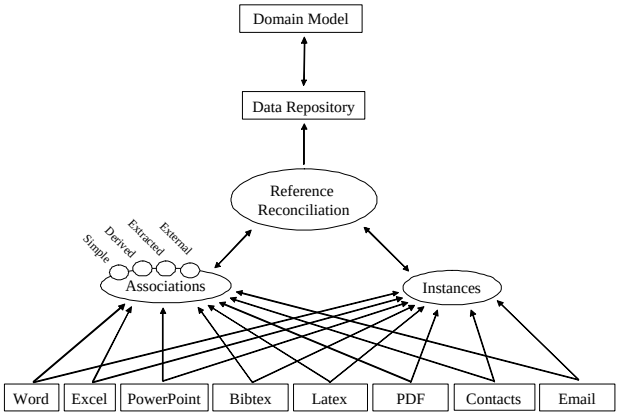

Figure 1: The architecture of SEMEX. SEMEX begins by extracting data from multiple sources. Such extractions create instances of classes in the domain model. SEMEX employs multiple modules for extracting associations, as well as allowing associations to be given by external sources or to be defined as views over other sets of associations. To combine all these associations seamlessly, SEMEX automatically reconciles multiple references to the same real-world object. The user browses and queries all this information through the domain model.

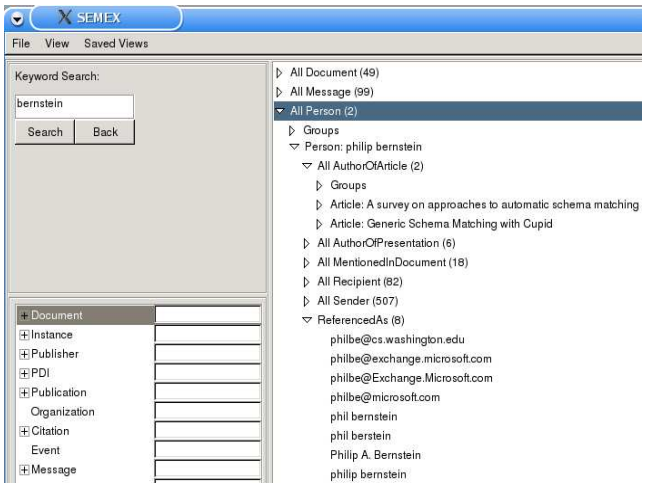

Figure 2: A sample screenshot of the SEMEX interface. The user can formulate either a keyword query (top left) or a more specific selection query (bottom left). SEMEX displays all the information about a particular individual and enables browsing the information by association. As seen in the bottom of the right pane, SEMEX needs to reconcile multiple references to the same real-world object.

already stored conveniently in the data sources and they only need to be extracted into the domain model. For example, a contact list already contains several important attributes of persons, and email messages contain several key fields indicating their senders and receivers.

- *Extracted:* A rich set of objects and associations can be extracted by analyzing specific file formats. For example, authors can be extracted from Latex files and Powerpoint presentations, and citations can be computed from the combination of Latex and Bibtex files.
- *External:* External sources can explicitly define many associations. For example, if CiteSeer were to publish a web interface, one could extract citation associations directly from there. Alternatively, a professor may wish to create a class MyGradStudents and populate the class with data in a department database.
- *Defined:* In the same way as views define interesting relations in a database, we can define objects and associations from simpler ones. As simple examples, we can define the association coAuthor, or the concept emailFromFamily.

Reference reconciliation: Since the data we manage in PIM is very heterogeneous and we need to support multiple sources of associations, it is crucial that the data instances mesh together seamlessly. To truly follow chains of associations and find all the information about a particular individual (or publication, conference, etc.), it is crucial that SEMEX be able to reconcile the many references to the same real-world object. Observe that reconciliation is much more challenging in our context. Most of our objects have very few properties (e.g. a Person instance may have only a name or an email address), and even this information is rather noisy because of automatic extraction from numerous sources. This is in contrast with traditional approaches that typically consider objects in one table with a larger number of attributes. The complete details of our approach is given in [5].

The result of the reconciliation algorithm is a high-quality reference list of a set of objects (e.g., people, publications). SEMEX can now leverage this list for generating new instances and associations. For example, we search for occurrences of person names, paper articles, etc., in email bodies, spreadsheets, Word and PDF files to create additional associations such as MentionedIn or isAbout. Further, we exploit the overlap between one's extracted instances and the instances found in a spreadsheet or a database for discovering and importing new instances.

Data repository: SEMEX stores the extracted instances and relationships in a separate database. We currently represent this information as RDF that is stored and retrieved using Jena [7]. We use Lucene [9] to index object instances by the text occurring in their attribute values. SEMEX is able to keep information up-to-date by periodically crawl the desktop and extract instances and associations in an incremental mode.

Browsing and querying interface: SEMEX offers an interface that combines intuitive browsing and a range of querying options. Figure 2 shows a sample screenshot from browsing SEMEX database. Initially, a user can simply type keywords into a search box and SEMEX will return all the objects that are somehow associated with the keyword. For example, typing `Bernstein` in the search box will produce a set of objects that mention Bernstein. Note that the answers to such a query can be a heterogeneous set of objects; SEMEX already classifies these objects into their classes (Person, Publication, etc.). When the `Bernstein` person object

is selected, the user can see *all* the information related to the person, and the relationship is explicitly specified. (e.g. AuthorOf, CitedIn). The user can then browse any of Bernstein's emails, papers (and then to the objects corresponding to other authors), etc. In addition, the user can see a chain of associations showing how the person object is associated with the object representing the user herself. Our keyword search has the following two important features.

Retrieve associated objects: SEMEX returns not only objects whose attribute values contain the required keywords, but also objects that are strongly related to multiple such objects. As an example, when a user performs a search on "Model Management", SEMEX retrieves all papers, presentations, and emails containing these keywords. In addition, SEMEX reports a list of persons, such as Bernstein, Melnik and Pottinger: while their names do not contain the required keywords, they have authored many papers and presentations on this topic.

Provide different views on data: By default, objects in a search result are ranked by a combination of (1) a *keyword score* computed using the TF/IDF [10] metric, (2) a *usage score* reflecting the create time, latest visit time, and visit frequency of the instance, and (3) a significance score that measures the importance of the object in the database. The significance score is obtained in a way similar to the page-rank algorithm [2], using associations as links between objects; however, associations are weighted differently based on their types (e.g., AuthorOf is more important than MentionedIn). Alternately, the user can view all objects in a chronological order, so that she can easily see the development or evolution of a topic or a project.

3. OVERARCHING PIM THEMES

Beyond the specific technical challenges, our experience with SEMEX has highlighted several higher-level themes that we believe will pervade many of the challenges in PIM. First, many of the challenges arise because PIM manages *long-lived* and *evolving* data. In contrast, most data management is used to model database states that capture snapshots of the world. The evolution occurs at the instance level as well as the schema level. So far, the evolution has manifested itself in challenges to querying, reference reconciliation and schema mapping. The second theme is finding the right *granularity* for modeling personal data. It is often possible to model the data at a very fine level. However, since PIM tools are geared toward users who are not necessarily technically savvy, it is important to keep the models as simple as possible. As we continue to investigate this tradeoff, we may find an interesting middle point between the models traditionally used for structured data and those for unstructured data. Another aspect of this tradeoff involves the amount of schema knowledge we want to endow our application. For example, when we know that an object on the desktop is an email message or a contact, we can possibly leverage that information in information searching and visualization. However, we would like the system to be as open-ended as possible to adding objects whose structures and/or schemas are unknown. Third, when designing PIM systems it is important to think from the perspective of the user and her interactions with data in her daily routine, rather than from the perspective of the database. We need to build systems to support users in their *own* habitat, rather than trying to fit their activities into traditional data management. Finally, there has been a lot of interest in systems that combine structured and unstructured data in a seamless fashion. We believe that PIM is an excellent application to drive the development of such systems, raising challenges concerning storing, modeling and querying hybrid data.

4. CONCLUSIONS

The SEMEX Project has two main goals. The first is to provide an indispensable tool for browsing the data on one's computer and additional devices. The system should enable easy location and browsing of the information and should hide the boundaries that exist today between data sitting in disparate applications. Second, once SEMEX has constructed a database of personal information, it should be able to leverage the database to increase the productivity of the user. A first example of leveraging such a database was given in [4] to enable *on-the-fly* information integration. Other examples include improved web search and useful visualizations of personal information. As a first step towards these goals, we have demonstrated how SEMEX automatically creates a database of objects and associations from one's desktop.

5. REFERENCES

[1] S. Abiteboul, R. Agrawal, P. Bernstein, M. Carey, S. Ceri, B. Croft, D. DeWitt, M. Franklin, H. Garcia-Molina, D. Galwick, J. Gray, L. Haas, A. Halevy, J. Hellerstein, Y. Ioannidis, M. Kersten, M. Pazzani, M. Lesk, D. Maier, J. Naughton, H. Schek, T. Sellis, A. Silberschatz, M. Stonebraker, R. Snodgrass, J. Ullman, G. Weikum, J. Widom, and S. Zdonik. The lowell database research self assessment. *CoRR cs.DB/0310006*, 2003.

[2] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7), 1998.

[3] V. Bush. As we may think. *The Atlantic Monthly*, July 1945.

[4] X. L. Dong and A. Halevy. A platform for personal information management and integration. In *Proc. of CIDR*, 2005.

[5] X. L. Dong, A. Halevy, and J. Madhaven. Reference reconciliation in complex information spaces. In *Proc. of SIGMOD*, 2005.

[6] M. Franklin, M. Cherniack, and S. Zdonik. Data management for pervasive computing: A tutorial. Tutorial at the 2001 VLDB Conference, 2001.

[7] Jena – A Semantic Web Framework for Java. http://jena.sourceforge.net.

[8] M. Kersten, G. Weikum, M. Franklin, D. Keim, A. Buchmann, and S. Chaudhuri. Panel: A database striptease, or how to manage your personal databases. In *Proc. of VLDB*, 2003.

[9] Jakarta Lucene. http://jakarta.apache.org/lucene/.

[10] G. Salton, editor. *The SMART Retrieval System—Experiments in Automatic Document Retrieval*. Prentice Hall Inc., Englewood Cliffs, NJ, 1971.