A Platform for Personal Information Management and Integration

Xin (Luna) Dong

lunadong@cs.washington.edu
University of Washington, Seattle

Abstract

The explosion of the amount 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 tools.

We study the Personal Information Management (PIM) problem from the data management point of view. We argue that the key for building a successful PIM system is to provide a logical view of one's personal information, consisting of semantically meaningful objects and associations. The thesis of this research is to build a prototype of a PIM system based upon this logical view, and demonstrate how we can leverage such a view to address the many PIM challenges.

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. In fact, several recent venues have noted Personal Information Management (PIM) as an area of growing interest to the data management community [12, 8].

As early as 1946, Vannevar Bush described the vision of a Personal Memex [2], which was motivated by the observation that our mind does not think by way of directory hierarchies, but rather by following *associations* between related objects. For example, we may think of a person, emails sent by the person, then jump to think of her papers, and papers they cited, etc. Supporting such associative traversal requires a *logical view* of the objects in one's personal information and the relations between them. However, today's desktops typically store information *by application* and in directory hierarchies. As a result, we need to examine directory structures or open specific applications in order to find information. As an example of the mismatch, information about people is scattered across our emails, address book, text and presentation files. Even answering a simple query, such as finding all of one's co-authors, requires significant work.

We argue that the key for building a successful PIM system is to provide a logical view of one's personal information, based on *meaningful* objects and associations. For example, users can browse their personal information by objects such as Person, Publication and Message and associations such as AuthoredBy, Cites and AttachedTo. With such a logical view, a PIM system can support users in their own habitat, rather than trying to fit their activities into traditional data management.

Below we give two scenarios to illustrate the challenges a PIM system faces, and briefly show how we can leverage the logical view of personal information to meet the challenges.

1.1 Scenarios

Scenario 1 (Search and browse associated objects). *Prof. Jones is writing a survey on model management. She wishes her PIM system will help her collect all information on model management, including papers on model management, people working on model management, institutes that conduct model-management research, venues for publishing model-management papers, etc. When Prof. Jones browses the collection of information, she may want to see more details or related information. For example, she may start with a paper and see its authors and citations, and then jump to other publications of one author, etc.* □

Challenge 1. *Support information search and browsing at the granularity of objects and associations, independent of the location of the information.*

Challenge 2. *Bootstrap the system by automat-*

ically populating data instances and removing instance duplications.

Scenario 1 shows that a person views the world as individual objects connected by various associations. Thus, she often wishes to search information at a fine-grained level and navigate her personal information by following associations, not necessarily being aware of the location of the information. This raises the fundamental requirement for a PIM system, stated as Challenge 1.

Importantly, since users are typically not willing to tolerate any overhead associated with creating additional structure in their personal data, a PIM system needs to populate the data instances mostly automatically. One key problem in this automatic population process is that the same entity in the world might be referred to in many different ways. To truly follow chains of associations and find all the information about a particular individual (or publication, conference, etc.), a PIM system needs to be able to reconcile the many references to the same real-world object. We summarize the above as Challenge 2.

By providing a logical view of personal information, consisting of objects and associations between the objects, we are able to naturally support fine-grained search and browsing. Automatic population of such a view is possible because a lot of data formats implicitly entail object instances and associations. For example, an email contains a field for sender and a field for recipients, and a Latex file contains a field for authors. From them we can easily extract instances of classes Message, Person, Article, and associations in types of sendFrom, sendTo, authoredBy. Further, the logical view of personal information highlights the associations between the references and thus provides additional evidences for removing instance duplications.

Scenario 2 (Integrate external data). *Prof. Jones wishes to find out whom of her acquaintances has published in this year's SIGMOD, and meanwhile store information for papers in her area for future use. She has a webpage with Sigmod accepted papers. However, going through the list is tedious and error-prone, and picking out interesting information for storage (maybe in another format) is even more labor-intensive. A PIM system is expected to do the above for her automatically or semi-automatically.*

Next time when Prof. Jones gets a webpage with a list of papers, such as VLDB accepted papers, she may want to apply the same task and wish her PIM system to do it automatically for her. □

Challenge 3. *Support integration with external data on-the-fly.*

As illustrated in Scenario 2, a user often needs to integrate external data and query across personal and external information. We refer to such integration tasks as *on-the-fly integration*, because they are light-weight tasks performed by individuals for relatively transient goals. In contrast, today's data integration systems typically support heavy-weight tasks for queries that occur very frequently in organizations (*e.g.*, customer relationship management and integrated catalog search). An important challenge for a PIM system is to fundamentally change the cost-benefit equation associated with integrating data sources, and aid non-technical users to easily integrate diverse sources. A logical view of personal information can be used as an *anchor* into which we can integrate external sources.

1.2 Thesis hypothesis

The thesis of this research is to build a prototype of a PIM system that provides a logical view of one's personal information based on meaningful objects and associations, and to demonstrate how we can leverage this logical view to address the aforementioned PIM challenges. In particular, we expect the following deliverables of the thesis:

- An approach to automatically build a database of object instances and associations between the instances. In [4, 5] we described how we extract instances and reconcile references. Our future work is to do them in an incremental mode.
- An algorithm for integrating personal data and external data on-the-fly. In [4] we described a preliminary algorithm and we are going to implement and evaluate the algorithm.
- Algorithms of data analysis for supporting several personal-information search-and-browsing modes. This is ongoing work.
- A PIM system that integrates the above components and serves as a platform for other PIM services.

This proposal is organized as follows. Section 2 surveys the state-of-the-art PIM systems. Section 3 overviews our solution and briefly discusses several main technical challenges. Section 4 concludes.

2 State-of-the-art PIM systems

A number of PIM projects studied ways to effectively organize and search personal information. They all attempt to go beyond the traditional hierarchical directory model and present a unified user interface for personal data.

The Stuff I've Seen (SIS) project [7] and the Google Desktop Search toolkit *index* personal information and emphasize access through a unique full-text keyword-search. MyLifeBits [10] views personal data as a *graph* of documents: nodes represent documents and annotation metadata; edges represent the annotate relationship, where a file can be annotated by another file, by manually added text or audio. Placeless Documents [6] annotates documents with property/value pairs, and groups documents into *overlapping collections* according to property values. LifeStreams [9] views personal information as a *sequence* of documents in chronological order.

While the above PIM systems all capture some aspect of personal data and provide convenience for information access in certain modes, none of them can satisfy the user needs stated in the scenarios. The fundamental reason is that they do not support a logical view of one's personal data. In fact, among these PIM systems, only MyLifeBits manages to explicitly capture associations, but it captures associations at a coarse granularity: it does not distinguish different classes of instances or different types of associations. Further, the above PIM projects all consider information at the document level. As a result, they cannot integrate structured data in a semantically meaningful way, or effectively facilitate reference reconciliation. Even reconciling different versions of the same document in these systems is difficult.

The Haystack project [11] models personal information as objects and associations between objects, and has successfully leveraged this model for personalized information presentation. However, Haystack mainly focuses on information rendering, and has not addressed the problems of data integration, data cleaning, and data analysis, which will be the emphasis of our research.

3 Our Solution

In this section we propose Semex (short for SEMantic EXplorer). We first present Semex architecture, and then address several key technical issues. For each of them, we state the challenge in the PIM context and briefly propose our solution. In [3] we gave more details and compared our solutions with the related work.

3.1 System architecture

Figure 1 depicts the components of Semex. The key is to provide a *logical view* of one's personal information, based on *meaningful* objects and associations. This logical view, described by a *domain model*, is provided by constructing a database of instances and associations, called *association database*, which complements current data storage. We now describe the three sub-modules of Semex.

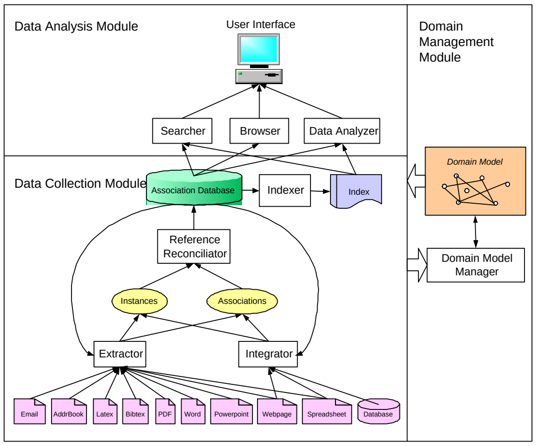


Figure 1: Semex architecture. Semex has three submodules: the domain management module plays the central role by providing and managing the domain model; the data collection module is responsible for data extraction, integration, cleaning and indexing; the data analysis module analyzes data for search and browsing.

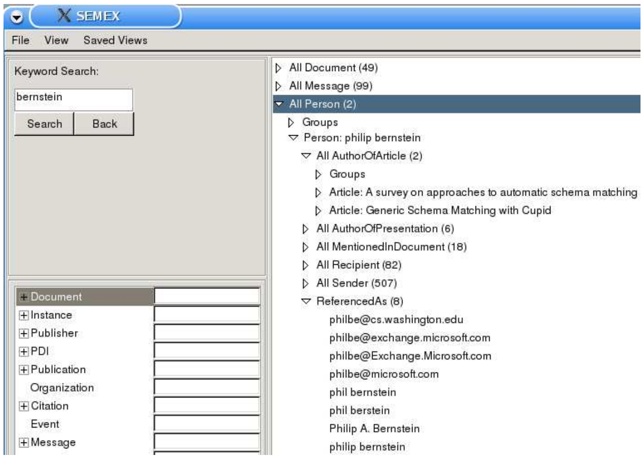
Domain management module: Semex provides a default domain model and meanwhile supports model personalization. The *domain model manager* component learns from previous integration and browsing experiences to suggest possible domain-model evolution.

Data collection module: Semex begins by extracting data from multiple sources using a set of object-and-association *extractors*. *Reference reconciliator* reconciles multiple references to the same object, and the results are stored in the *association database*. These instances further enable more extraction (such as association mentionedIn) and facilitate the *integrator* to integrate external structured data sources. Finally, instances are indexed for fast access.

Data analysis module: The *browser* and *searcher* components offer its users an interface that combines intuitive browsing and a variety of query facilities (see Figure 2). Further, *data analyzer* analyzes people's information and activities, and triggers certain notifications and alarms when an event occurs (for example, when a user opens a document, Semex will report the number of the user's acquaintances occurring in the document).

3.2 Reference reconciliation

Since the data Semex manages is very heterogeneous, it is crucial that the data instances mesh together seamlessly. In data extraction Semex generates *references*: each reference partially specifies an instance of a particular class; and several references may refer to the same real-world object. *The reconciliation algorithm partitions the set of references*



Keyword Search: bernstein

- Document
- Instance
- Publisher
- PDI
- Publication
- Organization
- Citation
- Event
- Message

```
▷ All Document (49)
▷ All Message (99)
▽ All Person (2)
  ▷ Groups
  ▽ Person: philip bernstein
    ▽ All AuthorOfArticle (2)
      ▷ Groups
      ▷ Article: A survey on approaches to automatic schema matching
      ▷ Article: Generic Schema Matching with Cupid
    ▷ All AuthorOfPresentation (6)
    ▷ All MentionedInDocument (18)
    ▷ All Recipient (82)
    ▷ All Sender (507)
    ▽ ReferencedAs (8)
      philbe@cs.washington.edu
      philbe@exchange.microsoft.com
      philbe@Exchange.Microsoft.com
      philbe@microsoft.com
      phil bernstein
      phil berstein
      Philip A. Bernstein
      philip bernstein
```

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.

in each class, such that each partition corresponds to a single unique real-world entity, and different partitions refer to different entities.

Reference reconciliation is a hard problem in general. Most of the previous works considered techniques for reconciling references to a *single* class (see [1] for recent surveys). However, in the PIM context there exist instances of multiple classes and rich associations between the instances. Though we can apply previous methods to each type of references in isolation, we miss the rich information carried in the associations. Furthermore, the previous techniques assume there are several attributes associated with every reference, and therefore a reasonable amount of information to consider in reconciliation. In the PIM context, a reference (such as one to a person) often has values for only one or two attributes. Finally, most previous techniques assume each attribute has a single value. In the PIM context, some attributes (e.g., person email) are multi-valued, so the fact that two attribute values are different does not imply that the two references refer to different real-world objects.

In [5] we described a reference reconciliation algorithm that is well suited for the PIM context. The key idea is to make extensive use of associations as extra evidences. For example, given two references to persons, we will consider whether they have authored the same paper or have common email contacts to help decide whether to reconcile them. To facilitate exploiting this information, we *propagate* information between reconciliation decisions for different pairs of references. For example, when we decide to reconcile two papers, we obtain additional evidence for reconciling the person references to their authors. This, in turn, can further increase the confidence in reconciling other papers authored by the reconciled persons.

3.3 On-the-fly data integration

One of the main objectives of SEMEX is to leverage the logical view of personal information as an *anchor* into which we can integrate external data sources on-the-fly. The key is to establish the semantic relationship between the external data source and the SEMEX domain model. *Formally, it takes as input the domain model and an external schema (wrapped to a relational database schema or an XML schema), and generates a set of queries such that evaluating the queries on the external data source will generate a set of class and association instances of the domain model.*

Previous works divided schema mapping into two separate steps (surveyed in [14]). In the first, called *schema matching*, we find *candidate* correspondences between the attributes of two schemas. In the second, referred to as *query discovery* [13], we build on *exact* correspondences and create mapping expressions that specify how to translate data from one schema to another. Note the gap between the output of the first step and the input of the second step. User's input is important to choose the exact correspondences from the candidate ones to fill in the gap. Manually filtering inappropriate candidates is often tedious and requires a good understanding of the domain model, so contradicts the spirit of on-the-fly integration. This is especially true in the PIM context, where many object classes may have attributes with common names (such as name and title).

Furthermore, previous works assumed schemas for mapping are fixed and well established. In the PIM context, however, the integration process may involve *extending* the user's domain model when it misses some required classes or associations.

We propose exploiting the associations for pruning inappropriate matching candidates, rather than requiring users to do it by hand. Specifically, we explore the heuristic that a database tuple seldom corresponds to several real-world objects that are not associated with each other. The logical view we provide treats associations as first-class citizens and enables applying this intuition.

3.4 Personal information search

The most important usage of a PIM system is to aid finding information on one's desktop and we now demonstrate several data search modes.

Finding instances related to a keyword: Given the logical view of personal information, SEMEX provides a search mechanism that is more intelligent than simple keyword search. Consider Scenario 1. When a user searches "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 on this topic. The challenge is to explore the associations between instances efficiently at runtime.

Now consider the ranking of the returned instances. We propose to rank search results by a combination of (1) a *relevance score* computed using the TF/IDF metric, (2) a *usage score* reflecting the creation time, last modification time, and visit frequency, and (3) a *significance score* that measures the importance of the object in the database using a PageRank alike algorithm.

Finding association chains: A user often tries to remember how she gets to know a person, an article, etc. SEMEX attempts to answer such questions by finding the *association chains*; an example is, a person is *mentioned in* an email that is *sent to* the user. We are especially interested in the chain that reflects the first time or the most recent time that the user interact with the given instance.

Finding patterns in external data: When a user opens a webpage, a document, etc., SEMEX searches for instances already existing in the association database. A more general requirement is to find similar instances; e.g., finding papers in the same field. The main challenge is to conduct this task efficiently and be tolerant with the different representations of an instance.

4 Conclusions

Personal information management is increasingly attracting attention as an area of study. This proposal examines the problem from the data management perspective. In particular, we ask us the big question: what is the right model for personal information and how can we leverage the model for data collection, data analysis, search and browsing? We propose providing a logical view of one's personal data, consisting of meaningful objects and associations between the objects. We aim to build a prototype of a PIM system based on such a logical view.

A PIM system faces many challenges, including automatic data population, reference reconciliation, on-the-fly data integration, intelligent keyword search, and model personalization. Each has received significant attention in the literature, and is known to be rather challenging in itself. Our goal is to demonstrate how we can leverage the logical view of personal data to address these challenges, and eventually improve the user's productivity in her daily life.

References

[1] M. Bilenko, R. Mooney, W. Cohen, P. Ravikumar, and S. Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems Special Issue on Information Integration on the Web*, September 2003.

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

[3] X. Dong. Thesis proposal: A platform for personal infromation and management and integration. http://www.cs.washington.edu/homes/lunadong/publication/semex_proposal.pdf, 2005.

[4] X. Dong and A. Halevy. A Platform for Personal Information Management and Integration. In *CIDR*, 2005.

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

[6] P. Dourish, W. K. Edwards, A. LaMarca, J. Lamping, K. Petersen, M. Salisbury, D. B. Terry, and J. Thornton. Extending document management systems with user-specific active properties. *ACM TOIS*, 18(2), 2000.

[7] S. Dumais, E. Cutrell, J. Cadiz, G. Jancke, R. Sarin, and D. C. Robbins. Stuff i've seen: A system for personal information retrieval and reuse. In *SIGIR*, 2003.

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

[9] E. Freeman and D. Gelernter. Lifestreams: a storage model for personal data. *SIGMOD Bulletin*, 1996.

[10] J. Gemmell, G. Bell, R. Lueder, S. Drucker, and C. Wong. Mylifebits: Fulfilling the memex vision. In *ACM Multimedia*, 2002.

[11] D. Karger, K. Bakshi, D. Huynh, D. Quan, and V. Sinha. Haystack: A general-purpose information management tool for end users based on semistructured data. In *CIDR*, 2005.

[12] 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.

[13] R. J. Miller, L. M. Haas, and M. A. Hernandez. Schema mapping as query discovery. In *VLDB*, 2000.

[14] E. Rahm and P. A. Bernstein. A survey on approaches to automatic schema matching. *VLDB Journal*, 10(4), 2001.