Managing Personal Information using iTrails

Position Paper: Tools and Techniques in Support of PIM

Jens Dittrich **Marcos Antonio Vaz Salles** **Lukas Blunschi**

ETH Zurich
8092 Zurich, Switzerland
dbis.ethz.ch | iMeMex.org

ABSTRACT

We would like to present and discuss the iTrails framework for pay-as-you-go information integration which was recently presented at VLDB 2007 [7]. iTrails allows users to provide mini-mappings on their data that sharply increase the quality of search results. The core idea is to extend the semantics of a standard graphical search engine such that the quality of search results approaches the quality of a full-blown information integration system. We would like to discuss the impact of iTrails on the CHI-side of Personal Information Management Tools. In particular, we would like to better understand how to provide interfaces and tools that allow end-users to make use of our framework.

Author Keywords

trails, PIM, dataspaces, search, pay-as-you-go information integration

INTRODUCTION

In 2005, Franklin et al. [5] introduced a new abstraction for data management called dataspaces and envisioned a type of system called DataSpace Support Platform (DSSP) to manage a dataspace.

One of the key challenges of a DSSP is to perform semantic integration of loosely connected and heterogeneous data sources. In contrast to an information integration system, where full semantic integration based on a global schema is required before any services over the data may be provided, a DSSP must perform semantic integration only when needed in a "pay-as-you-go" fashion. The iTrails [7] framework provides a powerful, generic building-block towards such a solution.

The core idea of our approach is to consider a graphical (Desktop/Web/Enterprise) search engine as a baseline and then add hints (aka *trails*) to this baseline such that the quality of query results approaches the quality of a full-blown information integration system.

ITRAILS USE-CASES

The iTrails framework consists of three different classes: *semantic trails*, *association trails*, and *lineage trails*. The first class was presented in [7]. We are currently exploring the other classes. For space constraints we reduce the following discussion to semantic trails. Note that the name 'trails' was inspired by [3].

The basic idea of a semantic trail is to provide a mapping from one query to another, i.e., $Trail = Q_L \rightarrow Q_R$. The semantics of this are that "whenever we query for Q_L, we should also query for Q_R." Trails are specified by the user or mined semi-automatically from user content (see below). Trail definitions are explored during query processing to enhance the quality of query results. Please see [7] for details. In the following we provide some use cases illustrating the power of our framework.

1. **Structural Shortcuts:**
   `yesterday` → `//*[date=today()-1]`
   This trail rewrites a search query containing the keyword `yesterday` to a complex query returning data items with the `date` attribute set to yesterday.
   Impact: by typing a simple keyword like `yesterday` a query is rewritten to a query considering schema knowledge that better captures the user's intent.

2. **Structural Rewrites:**
   `myProject` → `/jens/work/PIMLAB//*`
   This trail rewrites a keyword search for `myProject` to a path expression `/jens/work/PIMLAB//*` selecting all data items residing in directory /jens/work/PIMLAB or any of its subfolders.
   Impact: by typing a simple keyword like `myProject` a query is rewritten to a query considering structural knowledge of the data, again better capturing the user's intent.

3. **Language Agnostic Search:**
   `car` $\rightarrow_{0.8}$ `voiture`
   This trail rewrites a search query containing the keyword `car` to a search query containing the keyword `voiture`, i.e., the french word for 'car'. The weight 0.8 indicates that results obtained by `voiture` should be dampened in their rank by a factor of 0.8, i.e., results added by the trail rewrite will be less important than the original results. This use-case illustrates that *language agnostic search* [1] is in fact just a special case of the iTrails framework. The same holds for *thesauri* like wordnet [9], or *synonyms* [6]. The benefit of our technique is that it is not restricted to a specific use-case.
   Impact: by typing a simple keyword like `car` all search results having mentionings of cars in other languages will also be returned.

4. **User-specific Vocabulary:**
   `mike` $\rightarrow_{2.0}$ `mike jordan`
   `mike` $\rightarrow_{0.1}$ `mike smith`

   The first trail rewrites a search query containing the keyword `mike` to a search query containing the keywords `mike jordan`. Again, weight 2.0 is a weight indicating that results retrieved by `mike jordan` are two times more important than results retrieved by `mike`. Furthermore, the second trail indicates that results obtained by rewriting `mike` to `mike smith` are only 0.1 times as important as results obtained by `mike`, i.e., these results should obtain a very low total score.

   Impact: by typing a simple keyword like `mike` a query is rewritten to a more precise query reflecting the intention of the user. The result set obtained by the query will be the same, however, the scoring of results is changed by the trail rewrites. This removes ambiguity assuming that the user's data contains many different mikes, e.g., 'mike smith', 'mike jones'.

5. **Web Bookmarks:**
   `news` $\rightarrow$ `http://www.nzz.ch`

   This trail rewrites a search containing the keyword `news` to a URL retrieving the content of the news web page http://www.nzz.ch. This illustrates that standard Web bookmarks are just a special case in our iTrails framework.

   Impact: by typing a simple keyword like `news` a query is rewritten to also consider data available on the Web.

6. **Deep Web Bookmarks:**
   `train home`
   $\rightarrow$ `http://www.sbb.ch/?from=florence%to=zurich`

   This trail rewrites a search query containing the keywords `train home` to a URL retrieving the result of a train search from a web site providing railway timetable information (in this case for swiss trains).

   Impact: by typing a simple keyword like `train home` a query is rewritten to a deep web query retrieving data from a web database.

Where do Trails Come From?

An interesting challenge for both the information management as well as the CHI community is to understand how to obtain trails. We argue that an initial set of trails should be shipped with the initial configuration of a system. Then, a user (or a company) may extend the trail set and taylor it to her specific needs in a pay-as-you-go fashion. For instance, whenever the user detects that a query result may be enhanced by adding another trail she may do so.

We see four principal ways of obtaining a set of trail definitions:

1. Define trails using a drag&drop frontend,
2. Create trails based on user-feedback in the spirit of relevance feedback in search engines [8],
3. Mine trails (semi-) automatically from content,
4. Obtain trail definitions from collections offered by third parties or on shared web platforms.

As an example of (4.), sites in the style of bookmark-sharing sites like `del.icio.us` could be used to share trails. As shown above, a bookmark is just as a special case of a trail, in which a set of keywords induce a specific resource on the web. Sites for trail sharing could include specialized categories of trails, such as trails related to personal information sources (e.g., email, blogs, etc), trails on web sources (e.g., dictionaries translating location names to maps of these locations), or even trails about a given scientific domain (e.g., gene data). We would like to discuss the different possibilities during the PIM 2008 workshop.

CONCLUSION

iTrails [7] provides a new way to enrich the semantics of a search engine by adding hints (aka trails). Using trails as provided by the iTrails framework users are enabled to perform pay-as-you-go information integration as envisioned in [5]. A prototype of iTrails is implemented in the iMeMex Dataspace Management System [4, 2] and available for download at `iMeMex.org`.

We would like to discuss the idea of trails with the CHI and PIM communities to better understand its impact on user front-ends, user support, and PIM tools in general.

REFERENCES

1. L. Ballesteros and W. B. Croft. Phrasal Translation and Query Expansion Techniques for Cross-language Information Retrieval. In *SIGIR*, 1997.
2. L. Blunschi, J.-P. Dittrich, O. R. Girard, S. K. Karakashian, and M. A. V. Salles. A Dataspace Odyssey: The iMeMex Personal Dataspace Management System (Demo). In *CIDR*, 2007.
3. V. Bush. As we may think. *Atlantic Monthly*, 1945.
4. J.-P. Dittrich, M. A. V. Salles, D. Kossmann, and L. Blunschi. iMeMex: Escapes from the Personal Information Jungle (Demo). In *VLDB*, 2005.
5. M. Franklin, A. Halevy, and D. Maier. From Databases to Dataspaces: A New Abstraction for Information Management. *SIGMOD Record*, 34(4), 2005.
6. Y. Qiu and H.-P. Frei. Concept Based Query Expansion. In *SIGIR*, 1993.
7. M. A. V. Salles, J.-P. Dittrich, S. K. Karakashian, O. R. Girard, and L. Blunschi. iTrails: Pay-as-you-go Information Integration in Dataspaces. In *VLDB*, 2007. slides at `vldb2007.org/program/slides/s663-vazsalles.pdf`, video at `http://www.youtube.com/watch?v=F24-UHYFjHY`.
8. R. Schenkel and M. Theobald. Feedback-Driven Structural Query Expansion for Ranked Retrieval of XML Data. In *EDBT*, 2006.
9. WordNet. `http://wordnet.princeton.edu/`.