Dynamic Fusion of Web Data

Erhard Rahm, Andreas Thor, David Aumueller

University of Leipzig, Germany
http://dbs.uni-leipzig.de

Abstract: Mashups exemplify a workflow-like approach to dynamically integrate data and services from multiple web sources. Such integration workflows can build on existing services for web search, entity search, database querying, and information extraction and thus complement other data integration approaches. A key challenge is the efficient execution of integration workflows and their query and matching steps at runtime. We relate mashup data integration with other approaches, list major challenges, and outline features of a first prototype design.

1 Introduction

The need to fuse data from multiple web sources is rapidly increasing. This is demonstrated by the recent proliferation of mashup applications which combine content from multiple sources and services. Mashup applications are interactive and utilize flexible Web2.0 user interfaces. Content integration for such mashups is dynamic, i.e., it occurs at runtime (on demand) based on specific user input. Driving forces for the broad adoption of mashups are the availability of several development frameworks (e.g., Google Web Toolkit), as well as the proliferation of web APIs and information extraction tools for easy access to many websites, search engines or data feeds. Several tools (e.g., Yahoo Pipes, OpenKapow, Mashmaker [2]) also support visual interfaces to enable the construction of simple mashups without programming.

The potential for fast development makes mashups a highly attractive approach for integrating web data from several sources. This is because traditional, schema-focused data integration approaches (data warehouses, query mediators and – to a lesser degree – schema-oriented peer data management systems) suffer from a high upfront development effort for resolving semantic heterogeneity [3]. The effort needed to determine a global schema and/or precise schema mappings also limits scalability of schema-focused approaches to many sources. Web search engines, on the other hand, scale to many web sites but lack sufficient support for structured data sources of the "hidden web". Several approaches are being investigated to better provide integrated access to both unstructured and structured web sources with good scalability. For example, MetaQuerier provides unified entity search interfaces over many structured web sources of the hidden web [1]. PayGo aims at providing web-scale, domain-spanning access to structured sources [4]. It tries to cluster related schemas together and to improve search results by transforming keyword search queries into structured queries on relevant sources. One aspect missing from such search approaches is the post-processing of heterogeneous search results, in particular an online fusion of corresponding (matching) objects.

Mashups demonstrate a more programmatic, workflow-like integration approach, complementary to the query- and search-based data integration approaches. In fact, mashups are massively built on the idea of reusing and combining existing services so that they can also use existing search engines and query services. However, current mashups are mostly very simple and do not yet exploit the full potential of workflow-like data integration, e.g. as needed for enterprise applications or to analyze larger sets of web data. Hence, we see the need for a more powerful workflow-like data fusion approach which preserves mashup features like Web2.0 GUIs, support for reuse and fast development.

Providing such an approach incurs several challenges, including the definition of an architecture supporting mashups for integration at three levels, i.e. data, application, and presentation level. Furthermore, a powerful workflow and programming model is needed supporting the execution of existing web services and generic services (or operators) for information extraction, entity search, database queries, and object matching. The set of usable services and data sources should be listed and semantically described in a metadata repository similar as proposed in [3]. A limiting factor for interactive mashups is runtime. Hence, techniques are needed to solve more complex integration tasks, e.g. involving query, search and object matching of larger datasets, within a short time.

In the next section we discuss features of a first prototype for workflow-like dynamic data fusion.

2 Information Fusion with iFuice

We are currently extending our iFuice system for dynamic, mashup-like data fusion [6] [7]. In [7] we also report on a complex mashup implementation to generate on demand aggregated (Google Scholar) citation counts for (DBLP) publication lists of authors and venues. Here we summarize some key features of the iFuice design which we believe make it suitable for dynamic data fusion within mashup-like applications.

- *Workflow-like data integration* and *operator-based programming model***.** iFuice provides a high-level script language to define integration workflows or mashups. The language consists of powerful generic operators which can be applied to different data sources and services. For example, a query operator takes as input the id of a query service (data source) and a query specification. Most operators are set-oriented, i.e. they can be applied on a arbitrary set of input objects and generate a set of result objects. Intermediate results can be stored in variables for use by other operators. There are several operators for set operations (e.g., union, intersection, and difference) and data transformation (e.g., fuse, aggregate) which can be used to post-process query results.
- *Utilization of instance-level mappings:* iFuice utilizes instance-level mappings describing relationships between instances of entity types. Such mappings can relate instances of different sources (e.g. corresponding authors or publications of different bibliographic sources) and often exist already as hyperlinks. In addition, for structured sources we support instance-level associations between objects of a given source, e.g. to interrelate an author with her publications. Such instance-level map-

pings can efficiently be used to fuse together corresponding objects, even in the absence of schema mappings. Materializing such mappings supports their reuse for different integration workflows and use cases.

- *Support for structured and unstructured data sources*. By providing appropriate access services structured as well as unstructured web sources are supported. Each source may be accessed based on entity ids (e.g. URLs), or using structured queries or keyword search. Furthermore, we can leverage existing entity search engines or general search engines to reuse their results aggregated from many other sources.
- *Metadata repository:* Usable data sources and services are recorded in a repository and are assigned to entity types (e.g. publication, author). Furthermore, all available mappings and their semantic mapping type (e.g. publications of authors) are maintained. Entity and mapping types are part of a so-called domain model which can be incrementally extended as needed. A domain model is at a higher abstraction (ontological) level than a global database schema and helps to locate semantically relevant sources and services.
- *Iterative query strategies:* The use of existing search engines may require several queries for more complex integration tasks to obtain a sufficient number of relevant result entities. iFuice therefore allows to iteratively refine query results, where the execution of subsequent queries may be interactively controlled by the user. The OCS application [7] uses refining queries to the entity search engine Google Scholar to obtain citations for a set of publications. Intermediate results are shown to the user while the system executes additional queries to complete the result. Using such query strategies allows the quick generation of approximate results which can be further improved as needed.
- *On-the-fly object matching*: Dynamic data fusion requires to match corresponding objects from different sources and fuse their attribute values at run time. Using the MOMA framework [8] we provide a large spectrum of match strategies from which one can choose. In particular, the reuse of existing mappings can help to achieve a fast object matching.

Our investigations on dynamic mashup-like data fusion have just begun and several difficult research problems still need to be addressed, e.g. the automatic generation of iterative query strategies and on-the-fly object matching approaches.

References

1. Chang, K:, He, B:, Zhang; Z.: Toward Large Scale Integration: Building a MetaQuerier over Databases on the Web. Proc. CIDR (2005)
2. Ennals, R., Garofalakis, M.: MashMaker: Mashups for the Masses. Proc. Sigmod (2007)
3. Franklin, M., Halevy, A., Maier, D.: From Databases to Dataspaces: a New Abstraction for Information Management. SIGMOD Record 34(4): 27-33 (2005).
4. Madhavan, J., Jeffery, S. R., Cohen, S., Dong, X., Ko, D., Yu, C., Halevy, A.: Web-scale Data Integration: You can only afford to Pay As You Go. Proc. CIDR (2007)
5. Nie, Z., Wen, J.-R., Ma, W.-Y.: Object-level Vertical Search. IIWeb (2007)
6. Rahm, E., Thor, A., Aumueller, D., Do, H.-H., Golovin, N., Kirsten, T.: iFuice - Information Fusion utilizing Instance Correspondences and Peer Mappings. WebDB (2005)
7. Thor, A., Aumueller, D., Rahm, E.: Data Integration Support for Mashups. IIWeb (2007)
8. Thor, A., Rahm, E.: MOMA - A Mapping-based Object Matching System. CIDR (2007)