Towards an update-enabled Mediator System using Semantic Web technology [*]

Stefan Haun, Sandro Schulze, Andreas Nürnberger
Otto-von-Guericke-University
Magdeburg
{shaun, sandro.schulze, nuernb}@iti.cs.uni-magdeburg.de

ABSTRACT

A large part of implementing information retrieval or data mining systems consists of joining data from different sources to connect related items, enrich data sets and find similarities or contradictions. Mediators take care of the conversion between different formats and access to the data sources to provide a unified view, however these systems either only support read operations or are limited to a schema derived from a fixed set of data sources.

We sketch a concept for a mediator based on Semantic Web technology – especially RDF/OWL and SPARQL – enriched with semantics from the wrappers and an ontology describing the schema integration. Our system is able to handle updates on the unified view, feeding them back to the respective data sources and therefore extends the capabilities of classic data warehouses of federated databases.

Furthermore we analyze how CARSA, a meta-search engine framework, can be extended to implement the concept and discuss upcoming problems and ideas for their solution.

1. MOTIVATION

With more and more data sources available, it gets harder to integrate the different data formats and maintain and link the provided information. Although algorithms for processing the data are independent from their representation, much time must be invested in the development for wrappers and transformation functions between an actual data representation and the one needed for a specific algorithm, often reimplemented for each distinct data source. Automated converters are available but often suffer from the lack of knowledge about semantics necessary to come up with correct conversions.

We elaborate a system that allows to easily plugin data sources and access the contained information, but still is able to propagate back changes on the integrated data. This is achieved by adding semantics to the data, using RDF/OWL in combination with Ontologies and custom wrappers, enabling a mediator to integrate arbitrary data sources. With knowledge of the datas' meaning we want to build a system which is able to determine how updates on the RDF-based representation have to be applied to the original data sources, therefore obtaining an integrated view on the data which is no more read-only.

The CARSA framework, developed at the *Data and Knowledge Engineering Group*, will be used to implement the concept. Existing solution for RDF-based mediation and query planning shall be integrated if possible.

This paper gives a short overview on related work according data integration and Semantic Web technologies and afterwards presents out hitherto ideas on such mediator system, closing with an outlook on the next steps and plans on future development.

2. RELATED WORK

Related work comes from the field of Databases, especially data integration, which is concerned with a unified view on data from multiple, heterogeneous sources, and the field of Data and Knowledge Engineering, providing Semantic Web technology which we use to represent information in our system.

2.1 On data integration

In general, Data Sources (DS) are autonomous with respect to design, schema, if any is defined, data model and management. Often they are created in advance on usually not with integration in mind. In a data integration system, queries are processed by translating them into (sub-)queries in a form appropriate for the DS which is known to contain the respective data. This is not a trivial task as different DS may have differing models and access limitations have to be considered. Among heterogeneity in hardware and communication protocols, *logical* differences are the most intricate problem. If the DS are limited to relational database systems, there are no model conflicts and defined schemas are available, thus lessening the problem. However, not all data sources encountered do have this form.

Distributed Database Systems (DBS) are integrated, mostly homogeneous DBS where the distribution is known in advance.[13] DS which are added to the DBS are known, i.e. their schema is known and must adhere to the give global schema. Only some characteristics meet the requirements of

[*] The work presented here was supported by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898.

the system we aim at, such as local autonomy, distributed query processing and, to some extent, distributed transaction handling.

Federated Database Systems (FDBS) consist of semi-autonomous components – in the role of DS – which have been extended by an interface to communicate with each other, where each component is a centralized or distributed DBMS of itself. Tightly coupled FDBS have one or a unified schema, which is static and therefore makes it hard to add additional DS. Loosely coupled FDBS do not have a unified schema and the components are more autonomous. With no global schema, each source can create its own federated schema. Logical heterogeneity has to be solved manually by experts. This concept is suitable for a small number of (autonomous) sources if their independence has to be retained. [5, 12]

Mediator Systems are an alternative architecture for data integration systems where integration is achieved by providing a global view in DS – the mediated schema for user queries. Basic components are the mediator, which offers a common interface to all DS, and a wrapper per DS. The mediator receives a query based on the unified schema, decomposes and distributes it to the particular DS with regard to an optimal query plan. The results from sub-queries are merged and handed back to the user. All interactions with the DS are going through the wrapper, which acts as a transformer between the mediator system and other participated DS. The wrapper converts the representation of the DS into a suitable schema according to the unified schema of the mediator system, which can be and most likely is different to the internal representation of the DS. Source descriptions are used to model the relationship of the global, unified schema and schemas local to the respective DS. In comparison to FDBS, data sources are not necessarily databases themselves and can be easily added and removed. However, mediated system usually only provide read access to the represented data, a shortcoming which we try to remedy with our system.

In a *Data Warehouse (DWH)*, also known as Materialized View, data is loaded into a separate database. A-priory knowledge is needed about what information can be exprected from each source, the views on the respective sources that should be materialized and the global schema employed by the DWH. Because there is no coupling between the DWH and its DS, explicit updates are needed to reflect changes in the underlying data sources to the DWH, therefore changes are not represented immediately. [8, 3]

The BISON project[1] uses a DWH-like structure to represent all incorporated DS in one single large graph representation. As the DS, which are mostly result data from experiments, publications and similar structures, do not change very often, this structure is feasible. However, recent changes will not show up until the next update process. Mediator systems, as well as FDBS, can be considered as Virtual View approaches, because they provide the user with a unified view on all DS without exposing their distinctiveness. As we want to reflect updates in the underlying DS immediately, mediator systems are more suitable for our approach.

Semantic Web technologies and data integration have already been combined, for example in [9], presenting a semantic web middleware for virtual data integration on the web, and concept-based querying in mediator systems is shown in [11]. Both systems serve a a basis for our approach.

2.2 On Semantic Web Technology

The Semantic Web[4] was originally intended to enrich the WorldWideWeb with machine-readable information. While this process is still under way, several concepts and frameworks have emerged which can be used to describe semantically annotated data.

The *Resource Description Framework (RDF)*[2] is a format for describing of relationships between a source (subject) and a target (object). Unified Resource Identifiers (URIs) are used to specify each component, making these descriptions unique. The *Web Ontology Language (OWL)*[3] can be used to describe ontologies, i.e. formal representations of concepts and their relationships. By combining both frameworks, it is possible to represent information and their relationships from arbitrary data sources.

As it is possible to describe concepts and relationships in different ways – resulting in different ontologies describing the same circumstance – ontologies need to be aligned. This process can be compared to creating a unified schema in DBS. If two ontologies can be mapped to each other, so can the information represented by two data sources using them. In our approach we rely on ontology matching as a replacement for schema mapping, however do not research the topic ourselves.

Similar to SQL queries in relational DBMS, *SPARQL*[4] has been developed as a query language for the Semantic Web. The syntax is similar to SQL, however it is tailored towards RDF-graphs. We use SPARQL as query language, matching the RDF representation of the integrated data.

2.3 Bridging RDBMS and RDF/OWL

Since RDF/OWL and RDBMS both describe data sets, there have been efforts to bridge the concepts and transform data between these models. In [6] is described how queries can be mapped to the Virtuoso database, an RDF triple store, i.e. a database optimized towards data in RDF representation. A formal description on semantics and algorithms for querying distributed data sources with SPARQL is presented in [10], a framework especially useful for mediator systems which themselves have to distribute queries among the several available data sources. On the other hand, there are tools mapping relational DBS to RDF graph. A survey on such tools can be found in [14].

Due to available XML representations of RDF graphs, it is tempting to use XML technology and XML databases to represent RDF. However, as pointed out in [1], a single RDF triple may have several representations which are distinct in a pure XML interpretation, although conveying the same information. Therefore it is mandatory to use RDF tools despite an XML representation being available.

[1] http://www.bisonet.eu

[2] http://www.w3.org/RDF/

[3] http://www.w3.org/2007/OWL

[4] http://www.w3.org/TR/rdf-sparql-query/

3. AN APPLICATION EXAMPLE

A typical task in PIM data integration can be to merge several address books from different sources, such as Social Networks, e.g. Facebook[5] or Xing[6], a personal address book in the PDA or mobile phone and a company-wide address book provided by LDAP or other software. Even though there are different sources to the user only the actual entry is relevant.

While views can often be integrated by merging all available data into a contact item, this often leads to information loss with respect to the source of information and relevance of conflicting parts. When, for example. the merge results in several phone numbers, the user has to choose which one would be correct in the current context, so he does not end up making business calls on the personal land-line. Semantic annotations, especially about the source, can be used to rank conflicting items in the integrated view.

Another question – and more relevant regarding this paper – is how updates on an integrated data set should be applied. When the user changes a specific phone number, in which data source must this change be committed? Information about the semantics of integrated data and their sources can be used to guide this decision, Section 5.2 however points out some problems which will arise due to intrinsic lack of information. Here a system can try to deduce where information must be put from the context and habits of a user, e.g. are phone numbers stored in Xing or rather in the personal address book and can a certain update be made at all?

4. CARSA AS RDF-BASED MEDIATOR

The CARSA system[7] as initially described in [2] has undergone changes to broaden its use and prepare it to be available to other researchers. Mainly there has been a split into several, distinct sections to be incorporated depending on the task at hand. Parts and features common to all use cases, mostly containing data sources and functions typical to information retrieval applications, have been moved to the *CARSA Public Commons* section. Based on this the *CARSA Public SearchEngine* is a generic architecture for meta-search engines, containing adapters to keyword based search as well as methods for clustering and classification of results.

At the time of writing this paper we have already begun to enhance the CARSA system by Semantic Web capabilities, i.e. handling of RDF graphs and SPARQL-based requests. The existing plugin architecture will then be used to incorporate wrappers between arbitrary data sources and the RDF-based internal representation. If necessary we extend these plugins to be general data access plugins, enabling them to not only read from, but also to update the data sources. The available query routing mechanism will be leveraged to implement the mediator structure described in [9].

However, the CARSA system itself was also designed for read-only access to the underlying search engines. The data representation and plugin meta-information must be enhanced to ascertain the source of a specific RDF triple and to be able to feedback changes to the original sources. It is still to be assessed whether the *CARSA Public SearchEngine* architecture is to be used or a distinct system will be implemented based on the *CARSA Public Commons* package only.

5. UPDATES THROUGH SEMANTICS

Instead of a unified schema, which would require adaption whenever a data source is plugged in or removed, we use the semantics of the represented data to determine how updates must be conducted. The user applies changes to the RDF view, i.e. updates, deletions and additions. The mediator then determines how the underlying data sources have to be manipulated in order to reflect the desired change in the RDF representation. By changing the RDF graph the user conveys the intention to change the underlying data sources in such way that their representation will result in the changed RDF graph.

5.1 Interaction

The interactions which are available on the RDF-based information view must match the interactions for similar integrated views (see Related Work) and should also allow to change the underlying data sources through the presented view.

The read-only interaction schemes on the view can be classified as Lookups as closed queries to the RDF representation. These are achieved via SPARQL queries and part of the implementation of available RDF frameworks. Rather than looking for specific structures directly, the user may have open queries and therefore must be able to Navigate through the data in order to Explore it. Hitherto existing solutions are already capable of supporting these interactions.

Additional to the above, we also want to allow changes in the data representation. These changes may be:

1. Update already existing items or their properties: For these changes, the data source of the item – or the property – is identified and the update applied there. Problems may arise if the change can only be reflected by moving the item between data sources.

2. Delete items or their properties: Similar to an update, the data source must be identified and the item deleted from there. If an item appears on several data sources and has been merged, it must be deleted from each of them or the removal must be disambiguated.

3. Add items or properties: In contrast to the above manipulations, there is no known data source for an added item. The mediator needs to determine the data source in which the item should be created in order to show up in the RDF view. Dependent on the incorporated sources, it may be necessary to disambiguate between several update actions, as elaborated in the *Disambiguation* section below.

For accumulated values or items, which have been assembled from multiple source, the update operations may not be as straightforward as listed above. For example an accumulated value, such as the number of books a person owns, can only be changed be adding or removing books – these values cannot be directly manipulated. Also, not every data

[5] http://www.facebook.com

[6] http://www.xing.com

[7] http://www.findke.ovgu.de/en/research/content/tools_demos/carsa.html

source may support the desired update. The mediator has to keep track of which interactions are available on the presented data. So far we came up with two solutions, both having their advantages and drawbacks:

1. On each intended update the system can perform a dry run and report whether the operation would be successful. While this solution is relatively easy to implement, it is not acceptable from a user's point of view. Neither is the system able to tell whether a planned operation would be possible, nor is it possible to enumerate available interactions to be presented to the user. These limitations deter the user from building a successful mental simulation towards the solution of a task at hand, therefore make it very difficult to achieve a specific goal when interacting with the system.
2. Based on limitations stated in meta-information about data sources, the mediator can keep track of constraints towards the interactions available for the system. From those constraints a set of interactions can be derived for each item in the RDF graph and be presented to the user. This solution, however, results in a much higher effort on developing, implementing and running the mediator system.

5.2 Disambiguation

As there may be data sources with similar semantics, e.g. person profiles from social networks, it may not always be possible to decide which data source should be changed in order to achieve a certain state. This especially applies to added information, as there is no history or meta-data for this piece of information which would allow to map it to a data source. In order to solve the problem, an explicit disambiguation is necessary, for which we have so far elaborated several approaches:

1. The user is presented with a list of possible actions through the user interfaces, from where he is asked to select one to his like. This solution has two major drawbacks: First, there must be a user interface at all, which might not necessarily be the case with agent-based systems. Second, the user might not know or might not want to be concerned with the selection of an appropriate data source to be changed. This form of presentation breaks the unified view on all data sources.
2. There is a reasoning mechanism which allows to determine the best action to be taken. This might be achieved by a ranking of all possible changes, based on meta-information about the data sources provided by their wrappers. This ranking, however, will be very closely tied to the actual application and must be carefully designed to reflect the user's needs, otherwise odd decisions may lead to confusion. Still, there is a Semantic Gap between a user's interaction and his intent, for example an application could not easily deduce in which contact to store a just added telephone number. Unless there are clear directions about where to put specific data, the user may still need to make the decision.
3. The reasoning may be supported by finding similar data and deducing the target data source by these elements. This approach is based on the assumption that a user intends to keep the principal structure of his data models. So when a telephone number is added to a contact, the system tries to determine the source which is most likely to contain telephone numbers and puts the number there. Previous choices by the user may be incorporated.

Selecting from these approaches is a choice between automation and accuracy. None of these approaches provides a perfect solution and it is most likely necessary to combine them to create a ranking of possible actions and present them to the user for a more informed choice.

5.3 Mapping Relational data to RDF

As mentioned in Section 2.3, there are solutions for an automated mapping of relational data schemas to RDF structures. However, in most database schemas semantics are not included and often even distorted by normalization or optimization, therefore many data sources are missing the necessary information needed to implement a semantic driven system.

This semantic gap is closed by writing wrappers which explicitly take care of adding and removing semantics during transformation. In [1] the process of adding semantics to data is called *lifting*, as it lifts the data to a higher description level. When data is sent to a data source, the semantics must be replaced by a representation according to the target schema, i.e. removed in a process called *lowering* as an opposite to lifting. The semantics, which are only implicit for many data sources, must be defined by a wrapper. This again leaves us with the necessity of writing wrappers for each non-RDF data source, however the transformation now includes the semantics needed to incorporate the resulting information into any further processing steps.

Having a wrapper which not only transforms data, but also adds the semantics necessary to interpret the data, allows for automated conversion and processing in later processes, thus saving the programmer from developing further wrappers. Once the data is lifted into the semantic representation, there is no further need for data source dependent treatment.

5.4 Putting it all together

Based on the CARSA framework described in Section 4 we plan to implement an architecture similar to the one described in [9]. Data sources, if not already in RDF format, are wrapped and added using the plugin architecture, using the lifting and lowering concept from [1]. Query federation can be achieved with the system described in [10].

However, for disambiguation as mentioned in Section 5.2 further systems may be needed which analyze the user's context and interaction to determine the most likely goal and rank or present the possible resolutions accordingly. Currently we do not care about this specific problem.

6. CONCLUSION

We sketched a system that combines techniques both from data integration and the Semantic Web to integrate multiple data sources while keeping their semantics, which can be used for further processing without the need of data source specific converters. Having the semantics of each item allows not only to read, but also to apply updates on the unified view, as the mediator can determine the original source in which the changes have to be made. Some approaches for the disambiguation of changes have been presented, as it may not always be decidable to which source an update must be forwarded.

Since we do not build a Data Warehouse, but access the sources directly, changes – either external or from within the system – have immediate effect, making the mediator also usable for short-living data or high update rates, as they may appear for example in Personal Information Management (PIM).

The system will serve as a basis for unified navigation, exploration and manipulation of integrated data sources. While starting with a system for read-only, yet dynamic access, we strongly aim at update support using the semantic annotations.

7. FUTURE WORK

First thing to do will be the extension of the CARSA system by RDF capabilities as described in Section 4. Together with this development, we will assert how exactly the mentioned solutions for query federation can be incorporated into our framework. Afterwards, a prototype system integrating several sources from PIM and knowledge bases is built to research aspects of performance and and efficiency, alongside with a user interfaces that allows interaction with the data made available via the RDF mediator, leveraging existing tools for graph exploration and information visualization based on the represented data structures.

An interesting notion is the alignment of the definition of information provided by [7] as data with meaning, which is provided by an RDF graph. Therefore it might be worthwhile to investigate whether RDF-based mediation is rather information integration than data integration.

We left out the whole aspect of transactions. In our prototype we assume that each operation will be successful and executed towards the ACID criteria. A system which is able to ensure this environment with multiple, independent data sources is left for later research.

8. REFERENCES

[1] W. Akhtar, J. Kopechky, T. Krennwallner, and A. Polleres. XPARQL: Traveling between the XML and RDF worlds – and Avoiding the XSLT Pilgrimage. In *The Semantic Web: Research and Applications*, volume Volume 5021/2008, pages 432–447. Springer Berlin / Heidelberg, 2008.

[2] K. Bade, E. W. De Luca, A. Nürnberger, and S. Stober. CARSA - an architecture for the development of context adaptive retrieval systems. In *Adaptive multimedia retrieval: user, context, and feedback*, volume Volume 3877/2006 of *Lecture notes in computer science*, pages 91–101. Springer Berlin / Heidelberg, 2006.

[3] A. Bauer and H. Günzel. *Data-Warehouse-Systeme - Architektur, Entwicklung, Anwendung.* Number 2. dpunkt Verlag, 2004.

[4] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 2001.

[5] S. Conrad. *Föderierte Datenbanksysteme: Konzepte der Datenintegration.* Springer, 1997.

[6] O. Erling and I. Mikhailov. Integrating Open Sources and Relational Data with SPARQL. In *The Semantic Web: Research and Applications*, volume Volume 5021/2008, pages 838–842. Springer Berlin / Heidelberg, 2008.

[7] L. Floridi. Semantic Concepts of Information. In *Stanford Encylopedia of Philosophy.* 2005.

[8] W. Inmon. *Building the Data Warehouse.* Number 4. Wiley, 2005.

[9] A. Langegger, W. Wöß, and M. Blöchl. A Semantic Web Middleware for Virtual Data Integration on the Web. In *The Semantic Web: Research and Applications*, volume Volume 5021/2008 of *Lecture Notes in Computer Science*, pages 493–507. Springer Berlin / Heidelberg, 2008.

[10] B. Quilitz and U. Leser. Querying Distributed RDF Data Sources with SPARQL. In *The Semantic Web: Research and Applications*, volume Volume 5021/2008 of *Lecture Notes in Computer Science*, pages 524–538. Springer Berlin / Heidelberg, 2008.

[11] K.-U. Sattler, I. Geist, and E. Schallehn. Concept-based querying in mediator systems. *The VLDB Journal*, 14(1):97–111, 2005.

[12] A. P. Sheth and J. A. Larson. Federated Database Systems for managing distributed, heterogeneous, and autonomous Databases. *ACM Comput. Surv.*, 22(3):183–236, 1990.

[13] M. Tamer Øzsu and P. , Valduriez. *Principles of Distributed Database Systems.* Prentice Hall, 1999.

[14] W3C RDB2RDF Incubator Group. A Survey of Current Appraches for Mapping of Relational Databases to RDF. Technical report, W3C, 2009.