Information Systems Integration and Evolution: Ontologies at Rescue

Carlo A. Curino
Politecnico di Milano
carlo.curino@polimi.it

Letizia Tanca
Politecnico di Milano
tanca@elet.polimi.it

Carlo Zaniolo
UCLA
zaniolo@cs.ucla.edu

Abstract

The life of a modern Information System is often characterized by (i) a push toward integration with other systems, and (ii) the evolution of its data management core in response to continuously changing application requirements. Most of the current proposals dealing with these issues from a database perspective rely on the formal notions of mapping *and* query rewriting. *This paper presents the research agenda of* ADAM (Advanced Data And Metadata Manager)*; by harvesting the recent theoretical advances in this area into a unified framework,* ADAM *seeks to deliver practical solutions to the problems of automatic schema mapping and assisted schema evolution. The evolution of an Information System (IS) reflects the changes occurring in the application reality that the IS is modelling: thus,* ADAM *exploits ontologies to capture such changes and provide traceability and automated documentation for such evolution. Initial results and immediate benefits of this approach are presented.*

1. Introduction

Every Information System (IS) is subject to a continuous pressure of evolution, to the point that the *waterfall* software development methodologies, where the design is relegated to an up-front phase, are considered inadequate to face the modern IS design, development and maintenance. Furthermore, the data management core of an IS is recognized to be one of the most critical and difficult portion of software to evolve and integrate [1]. Web Information Systems (WIS) are typically characterized by collaborative nature of both their development and fruition: in this context, the need for better support of the evolution process becomes even more pressing. This is proven by the analysis carried out in [5], where we dissected the evolution of the DB backend of MediaWiki[1], a software platform powering over 30,000 websites including Wikipedia. In such analysis we captured an history of 170+ schema versions (in 4.5 years of life) by means of an operational language of Schema Modification Operators (SMO) [6]. The statistical results we collected[2] on the impact of these changes on applications provide strong evidence of a major need for better support of schema evolution in Web Information Systems. Moreover, the growing trend in modern business reality toward frequent acquisitions and fusions of companies and organizations exacerbates the need for integration of IS data management cores, required to maximize assets exploitation and control.

This justifies the big research effort that has been recently devoted to the related problems of schema evolution [2, 6], data exchange [10, 12], and data integration [13]. The outcome of these works provides solid theoretical foundations for several problem-specific solutions. Interestingly enough, most of the proposed approaches are based on some form of *logical mapping* on top of which powerful query rewriting techniques have been designed [9]. These elegant formal approaches are far from being usable in the everyday practice of professionals, which are left almost unarmed to face the tasks of evolving and maintaining complex IS.

In the past we tackled the problem of automating and easing the use of the above mentioned logical mappings along two orthogonal directions: (i) in Schema Evolution: by designing an operational language of Schema Modification Operators (SMO) inspired to the SQL:2003 DDL syntax, which is automatically translated to logical mappings (Disjunctive Embedded Dependencies [9]) used to support automatic query rewriting, in our prototype *PRISM* [6]; (ii) in Data Integration: by designing automatic (ontological) wrapper generators extracting ontological representations of relational [11] and XML data sources and by developing *X-SOM* [4, 8] an automatic tool achieving high precision and recall in the task of ontology mapping [3].

This paper describes a research activity born from the collaboration of Politecnico di Milano and University of California, Los Angeles, aiming at combining the above mentioned efforts to design a unified framework for data and metadata evolution and integration named *Advanced*

[1] See: `http://www.mediawiki.org/`

[2] See our results at: `http://yellowstone.cs.ucla.edu/schema-evolution/index.php/Main_Page`.

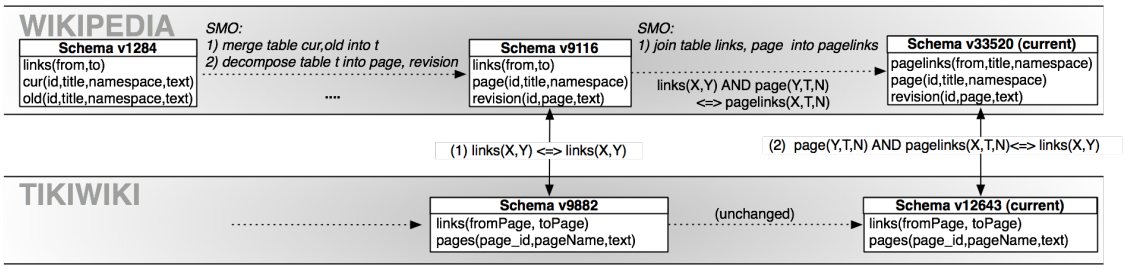

Figure 1. Two simplified fragments of the schema history of MediaWiki and TikiWiki (the schema revision numbers refer to the SVN repositories).

Data And Metadata Manager (ADAM). The main goal of *ADAM* is to factorize in a common core, based on (ontology) mapping and query rewriting, our prior contributions to the problems of schema evolution and data integration. *ADAM* exploits ontologies to support documentation of database integration and evolution, on the grounds that ontologies are a powerful resource to enable unmanned operations on data and metadata.

In this paper we briefly summarize the current state of our research presenting short-term milestones, which highlight immediate benefits of this integrated approach, and long term research goals. Among the immediate benefits we consider the automatic derivation of machine-readable documentation (i.e., an ontological specification) of the schema evolution managed by *PRISM* and the opportunity of exploiting historical information to improve the accuracy of the ontology mapping process in *X-SOM*, by developing history-aware, informed heuristics for schema matching. Our long term goals aim at enabling the Semantic Web vision on existing data, while simplifying everyday activities of DB Administrators and Application Developers.

The rest of this paper is organized as follows: Section 2 introduces our motivating and running example, Section 3 presents an extension of one of our tools to enable recording of machine-readable documentation of the schema evolution, Section 4 sketches the design of an history-aware heuristics, Section 5 presents our long term goals to develop the unified framework *ADAM*, and Section 6 draws our conclusions.

2. Running Example

Our running example is an excerpt of the MediaWiki[3] schema history, in particular the evolution of the relations representing `links` and `pages`. We compare this to the corresponding portion of the schema of a competing platform: TikiWiki[4]. Each of these systems has seen over 150 schema versions in few months/years of lifetime. We set our hypothetical goal to the evolution and integration of the data management cores of these two Web Information Systems. Figure 1 shows (a simplified version) of three revisions of the MediaWiki schema and two (unmodified) versions of the TikiWiki schema. The relationships between subsequent schema versions are captured both in terms of Schema modification Operators (SMO) and logical mapping (several details are omitted for the sake of presentation). We further comment on this example throughout the paper to highlight the benefits of an integrated framework.

3. Semantic Historical Metadata Manager

Our framework buils on *PRISM* [6], a system capable of assisting a DataBase Administrator (DBA) during the process of schema evolution. The *PRISM* user characterizes each step of schema evolution by means of Schema Modification Operators (SMO). Given this SMO-based specification, the system: (i) analyzes the proposed evolution w.r.t. information preservation and redundancy and then, upon user requests, (ii) automatically migrates data across schema versions, (iii) derives a logical mappings (Disjunctive Embedded Dependencies [9]) between schema versions, as shown in Figure 1, and (iv) automatically translates queries expressed on old schema versions into equivalent ones against the current schema. In [7] we showed how this specification, properly recorded and managed by a tool named *Historical Metadata Manager* enables the DBA to issue temporal queries on the history of the metadata itself. Here we present an extension of such tool, the *Semantic Historical Metadata Manager (SHMM)*, that exploits the technologies of Semantic Web to allow semantic querying and automatic reasoning on top of schema evolution histories. The use of history enhances the current mechanism for both schema evolution and integration, in two complementary ways: (i) an ontology can be derived from the schemas and mapping of Figure 1 as machine-readable documentation of the schema evolution, and (ii) the same representation enables better automatic schema integration, by exploiting ontology mappers that leverage the extra-information derived from history-awareness. In order

[3] See: `http://mediawiki.org`.
[4] See: `http://tikiwiki.org/`

to derive an ontology-based representation of the schema evolution we need to: (i) represent each DB schema in terms of an ontology, and (ii) capture the logical mappings between subsequent schema versions as ontological mapping. The first issue was basically solved in [11] where we defined effective heuristics to automatically generate an ontology representing a DB schema (by means of automatic reverse engineering of the relational schema toward the original conceptual ER diagram), similar approaches are [13] and the D2R Server[5]. The *Semantic History Metadata Manager*, presented here, solves (ii) by automatically deriving, from the SMO-based representation of the schema evolution designed by the DBA, an ontological specification of the schema evolution. This task is made possible thanks to the translation of SMOs into as subset of the Disjunctive Embedded Dependencies provided by PRISM [6] and thus their correspondence to Description Logic [9]. The overall result is a fully ontological representation of the schema history on top of which both semantic querying and automatic reasoning are enabled. Due to space limitations we illustrate the potential of this approach only by means of an example query: *Which was the previous representation of the column* `revision.text` *in the MediaWiki schema?* that can be expressed in SPARQL as follows:

```
SELECT  ?tab, ?col
WHERE { ?col columnOf ?tab  .
      ?col becomes "text" .
      ?tab becomes "revision" }
```

This query will return two tuples containing the column `text` as represented in previous versions of the MediaWiki schema by the `old` and `cur` relations, which were used to store respectively old and current revisions of an article page. By explicitly capturing the semantics of the evolution this framework enables, together with a better browsing of the evolution documentation, the automation of complex auditing tasks, and the possibility of exploiting the history of the schema for automatic schema integration as discussed in the following Section 4.

4. History-aware Schema Integration

The problem of automatic schema matching and mapping has been addressed in [4, 3, 8] where we described X-SOM, an ontology mapper that combines a suite of heuristics and state-of-the-art consistency-checking, which can be extended by the user thanks to a modular architectural design. We can now exploit such flexibility to define and test a novel history-based schema matching technique, operating representation of the schema history introduced in Section 3.

[5]See: `http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/`.

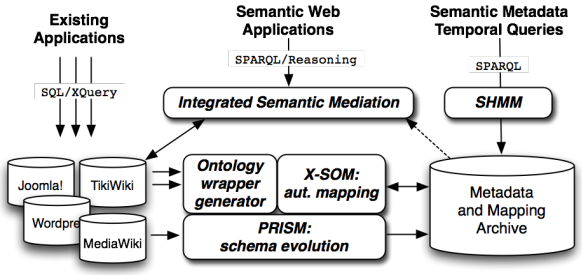

Figure 2. *ADAM* **general architecture.**

The proposed heuristics receives in input two schema histories and produces as output a matching of the two most recent versions of the schemas. This is, in particular, a meta-heuristics which operates by applying some of the lightweight heuristics available in X-SOM on pairs of schemas from the two histories – assuming that the histories' cardinalities be n and m we have $n \times m$ schema matchings to be performed. As illustrated in Figure 1, in many cases some of the historical versions of the schemas result easier to match thanks to more evident similarities. In the example of Figure 1 the two relations describing (internal) links between pages in MediaWiki and TikiWiki are almost identical when observed in version 9116 and 9882 [6] respectively. It is thus possible to exploit high-confidence results of our basic matching heuristics, applied to historical versions of the schema to: (i) increase the accuracy of matchings discovered in the current schema versions, and (ii) derive complex mappings by composing [12] simple mappings and complex schema changes. The key idea is to apply our automatic matcher on the simpler cases (where the two schemas result more similar) and exploit the schema changes — known because provided by the user at the time of evolution — to derive equivalent knowledge between less similar schema versions. This is well illustrated by the real-life example of MediaWiki and TikiWiki schema histories of Figure 1 where we derived the complex mapping: $page(Y,T,N) \cap pagelinks(X,T,N) \leftrightarrow links(X,Y)$ from the more obvious: $links(X,Y) \leftrightarrow links(X,Y)$, discovered under previous schema versions. To improve performance, together with selecting efficient basic heuristics, we can effectively prune the number of comparisons to be computed. In fact, most of the schema changes are circumscribed within limited areas of the schema, and thus big portions of it result unchanged throughout several evolution steps; e.g., the portion of the TikiWiki schema shown in Figure 1 remains unchanged throughout the analyzed history.

[6]The revision numbers refer to the following SVN repositories: `http://svn.wikimedia.org/viewvc/mediawiki/trunk/phase3/maintenance/tables.sql?view=log` and `http://tikiwiki.svn.sourceforge.net/viewvc/tikiwiki/branches/1.10/db/tiki.sql?view=log`.

5. Unified Evolutionary Framework

Several solutions to problems related to the Information System evolution and maintenance, such as data integration, data exchange, and schema evolution exploit the notion of mapping for the purpose of supporting query rewriting and providing documentation of the evolution itself. Thus, designing a framework to tackle these problems in a unified way we need to select an appropriate mapping language and efficient rewriting techniques. The natural choice falls on the class of logical formal languages. These languages provide great expressive power and strong theoretical guarantees but lack the usability of operator based languages such as our SMO. To this purpose we mask the logical core of our system to the final user, who is offered operational interfaces targeting each sub-problem, e.g., X-SOM and PRISM. Thanks to the correspondences between the formalisms of Disjunctive Embedded Dependencies (DED) exploited by the PRISM rewriting engine [9], and OWL, the Semantic Web esperanto, our framework is capable of exploiting a unified representation of the mappings for schema evolution and data integration, published accordingly to the specific problem considered, e.g., as an OWL ontology to support SPARQL queries. Figure 2 shows the overall architecture of *ADAM*. This framework assists the users in the challenging activity of schema mapping, by extracting an ontological representation of the schema and exploiting the automatic mapping and flexibility of X-SOM [4], and in the process of schema evolution by offering the functionalities of the PRISM workbench [6]. As a results *ADAM* is capable of archiving in an ontology-based format data and metadata evolution and integration histories. To the purpose of documenting and automating Information Systems evolution and maintenance, this archive represents an invaluable piece of information that tracks in a machine-readable format the histories of multiple Information Systems and the evolution of their mappings. On top of this *ADAM* enables: (i) semantic querying of the metadata histories by means of the Semantic History Metadata Manager of Section 3, (ii) the support of history-aware heuristics such as the one we described in Section 4, (iii) graceful (i.e., assisted and seamless) schema evolution support by means of automatic query rewriting, and (iv) semi-automatic data integration, allowing semantic queries (expressed in SPARQL) on top of the ontological representation of the mapped schemas.

6. Conclusion

In this paper, we presented an ongoing research effort to develop *ADAM*, a methodology and workbench to address the data-centric challenges of IS evolution and integration. At its logical core, *ADAM* integrates and unifies an assortment of powerful schema mapping and query rewriting primitives. But *ADAM* also provides an operator-oriented interface that encapsulates and streamlines this powerful and complex primitives to facilitate their actual usage in managing the evolution and integration process of an actual IS. *ADAM*'s mapping repository is fully compatible with the latest standards of the Semantic Web and thus delivers immediate advantages in terms of automatic, machine-readable documentation of the evolution and integration process, and history-aware automatic schema matching. Immediate benefits of this integrated approach are: (i) a history-aware schema matching heuristics, which is currently being validated against real-life examples, such as MediaWiki and TikiWiki, and (ii) the Semantic Historical Metadata Manager, a tool capable of supporting reasoning and semantic querying on the schema and mapping repository that is at the core of our framework.

References

[1] S. W. Ambler and P. J. Sadalage. *Refactoring Databases: Evolutionary Database Design*. Addison-Wesley Professional, 2006.

[2] P. A. Bernstein, T. J. Green, S. Melnik, and A. Nash. Implementing mapping composition. *VLDB J.*, 17(2), 2008.

[3] C. Curino, G. Orsi, and L. Tanca. X-som results for oaei 2007. *The Second International Workshop on Ontology Matching colocated with ISWC 2007*, 2007.

[4] C. Curino, G. Orsi, and L. Tanca. X-som: A flexible ontology mapper. *Database and Expert Systems Applications, 2007. DEXA '07. 18th International Conference on*, pages 424–428, 3-7 Sept. 2007.

[5] C. A. Curino, H. J. Moon, L. Tanca, and C. Zaniolo. Schema Evolution in Wikipedia: toward a Web Information System Benchmark. *ICEIS*, To appear: 2008.

[6] C. A. Curino, H. J. Moon, and C. Zaniolo. Graceful database schema evolution: the prism workbench. In *UCLA/CSD Tech. Rep., March 2008. Submitted for publication*, 2008.

[7] C. A. Curino, H. J. Moon, and C. Zaniolo. Managing the history of metadata in support for db archiving and schema evolution. In *UCLA/CSD Tech. Rep., April 2008. Submitted for publication*, 2008.

[8] C. A. Curino, G. Orsi, and L. Tanca. X-som: Ontology mapping and inconsistency resolution. In *Proceedings of the 4th European Semantic Web Conference (ESWC '07), poster presentation*, 2007.

[9] A. Deutsch and V. Tannen. Mars: A system for publishing XML from mixed and redundant storage. In *VLDB*, 2003.

[10] R. Fagin. Inverting schema mappings. *ACM Trans. Database Syst.*, 32(4):25, 2007.

[11] L. Macagnino. Ontology extraction from relational databases. Master's thesis, Politecnico di Milano, 2006.

[12] A. Nash, P. A. Bernstein, and S. Melnik. Composition of mappings given by embedded dependencies. In *PODS*, pages 172–183, 2005.

[13] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Linking data to ontologies. *Journal on Data Semantics X*, pages 133–173, 2008.