A reverse engineering approach for migrating data-intensive web sites to the Semantic Web

Nenad Stojanovic, Ljiljana Stojanovic, Raphael Volz
AIFB Institute, Univ. of Karlsruhe,Germany, {nst,lst,volz}@aifb.uni-karlsruhe.de

The Semantic Web is intended to enable machine understandable web content and seems to be a solution for many drawbacks of the current Web. It is based on metadata that describe the formal semantics of Web contents. In this paper we present an integrated and semi-automatic approach for generating shared-understandable metadata of data-intensive Web applications. This approach is based on mapping the given relational schema into already existing ontology structure using a reverse engineering process. As a case study we present this style of a schema- and data-migration for our Institute web portal. The presented approach can be applied to a broad range of today's data-intensive Web sites.

1. INTRODUCTION

The Semantic Web is one of today's hot keywords. It is about bringing ``[...] structure to the meaningful content of Web pages, creating an environment where software agents, roaming from page to page, can readily carry out sophisticated tasks for users." [17]. In order to enable this, web sites are enhanced with metadata that provide formal semantics for Web content. The key technology involved here are the ontologies. The ontologies provide consensual domain models, which are understandable to both human beings and machines as a shared conceptualisation of a specific domain that is given. Using ontologies, a content is made suitable for machine consumption, opposing to the content found on the web today, which is primarily intended for human consumption.

Currently people are slowly starting to build the Semantic Web, thus ontology-based metadata is being provided. This process of generating such metadata - also called semantic annotation - is mostly done by hand and therefore cumbersome and expensive. Visual annotation tools [*] make this task much easier, but even with sophisticated tools it is laborious to provide semantic annotations. Additionally, a maintenance problem arises; annotations must be consistent, must make proper reference, redundancy must be avoided and of course be maintained that results in a need for constant synchronization with their corresponding web content. However, because ontologies aim to consensual community (domain) knowledge, it is not enough to provide only *formal* semantics for information, but also *real- world semantics* allowing to link a machine processable content with the meaning for humans, based on *consensual* terminologies [6].

In this paper we addressed these problems and gave a solution for an important class of those data-intensive web sites that draw their data from relational databases. These sites have moved away from static, fixed web pages to those that are dynamically generated at the time of users' requests from data in relational databases. In order to migrate these web sites into the Semantic Web we have developed an approach based on mapping the given relational schema into an existing ontological structure, using a reverse engineering process. Using this mapping the database content can be **directly** used to provide the intended semantic annotations. We also present a tool that supports this mapping in a semi-automatic manner.

The benefits of the proposed approach are manifold: The process of providing metadata is automated and thus inexpensive and fast. Consequently, the content of dynamic web pages is machine-understandable and therefore visible for specialized search engines. Moreover, the problem of dynamic updating metadata according to changes in corresponding web pages is also resolved. The most important benefit is that information from various community members could be exchanged on the semantic basis.

Our approach can be applied to a broad range of today's data-intensive Web sites. One of the most common applications for such data-intensive web sites are most e-commerce applications, many kinds of directories and ``reporting sites'' [1]. Such data-intensive web sites have numerous benefits, i.e. a simplified maintenance of the web design (due to complete separation between data and layout), the automated updating of web content, etc.

Moreover, the using of ontologies as mediation level for product data exchange is already proposed [13]. Our approach could be a mechanism for resolving semantic problems that arise in this integration process.

The paper is organized as follows: Section 2 details the mapping architecture, mapping process and rules used in it. In Section 3 we present our case study by concluding with some lessons we learned. Before we conclude we contrast our contribution with related work.

2. THE MAPPING ARCHITECTURE AND PROCESS

2.1 Design rationale

2.1.1 Source data

Our mapping architecture is grounded on the logical database model found in running database systems [1]. The reader may note that the logical database model does not specify formal semantics of the contained information and thus is not sufficient per se as a conceptual backbone for the Semantic Web.

The reason for basing our approach on the logical data model is that most often no conceptual model (like an ER model) was created during the

---

[1] providing for example information about stock-quotes or the current weather

conception of the database or this information is lost by now. Naturally a mapping from ER models to ontologies preserves more information (like cardinalities).

Target data model for the mapping approach is F-Logic. More information about F-Logic can be found in [7]. Here only notice that it was developed to combine the rich data-modelling primitives of object-oriented databases with logical languages as developed for deductive databases.

2.1.2 Implementation constraints

The architecture should require only minor changes on the existing web application. Therefore the initial data is not converted. The newly implemented “Semantic Web” part of the web application should reference this source data and create the required information (in form of RDF files) on demand and dynamically by applying the generic mapping rules specified in this paper. Another reason not to convert the data is that constant synchronization would be needed.

2.1.3 Information preservation

We reinstate that the goal of this mapping is to preserve a maximum of information under the ontology framework. It is important to say that this process of the schema transformation cannot be loss less, as described in section 2.4. Needless to say almost all information is preserved.

2.2 Migration architecture

The general migration architecture is depicted in figure 1.

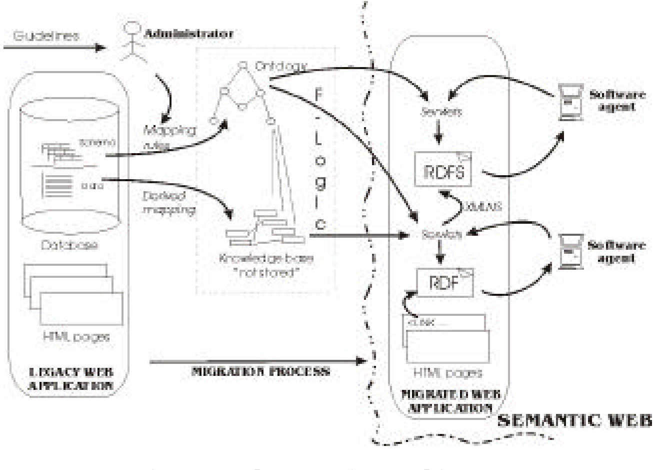


Figure 1. The mapping architecture

The input of the migration is a relational model that is derived from the SQL DDL. The database schema is mapped into the ontology using the mapping process described bellow, which applies the rules specified in the following sections. The same holds for database instances that are mapped into the knowledge base, based on the domain ontology.

The actual ontology is computed once (under the supervision and revision of the designer) and must be recomputed only if the database schema is changed. Knowledge base instances are computed on demand. Servlets are used to create the RDF output files from the ontology and the database data. Two files are produced: one file containing the ontology and another file containing instances that refers to the schema file using XML namespace mechanisms. To create the files F-Logic is mapped into RDF. This mapping is straightforward. The legacy HTML files must be changed minimally to contain a reference to their metadata descriptions[2]. This provides the semantic annotations for the legacy content.

2.3 Mapping Process

The mapping process enhances the semantics of the database by providing additional ontological entities. Our proposed mapping method consists of four steps:

1. Capture information from a relational schema through reverse engineering (consider relations, attributes, attributes types, primary keys, foreign keys / inclusion dependencies);

2. Analyse the obtained information to map database entities into ontological entities, by considering using a set of mapping rules specified in the following sections. Our rules are similar to [2], which describes a transformation into an object-oriented model. This phase is split into:

   2a. alignment of the top-level terms (How to decide which relation's name corresponds to which concept`s name);
   2b. using existing concept creation rules to determine set of relations in relational schema related to a concept;
   2c. using attribute creation rules to assign relation's attributes to the attributes of a concept;

Note: Using rules does not impose changes in the definition of the ontology, but only alignment between relational entities and ontological entities.

3. Make evaluation, validation and refinement of the mapping. Check whether all relational entities are mapped into corresponding ontological entities (whether the existing ontology is conceptually rich enough to represent relational schema completely). Also, the implicit semantic of the relational model must be mapped into explicit ontological structures (e.g. for each n:m relation in the relational model must exist a rule that defines that the corresponding ontological relations are inverse).

4. Form a knowledge base ("data migration").

The implemented system, presented in next section, provides assistance in all phases. Actually, the reverse engineering process cannot be completely automated as some situations can arise where several rules could be applied. User interaction is then necessary when this kind of ambiguities occurs and domain semantics cannot be inferred.

---

[2] HTML handles this situation with the following tag: `<LINK rel="meta" href="mydocMetadata.DC.RDF">`

2.4 Mapping rules

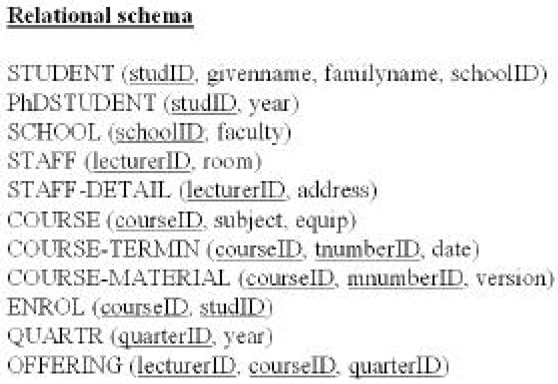
The mapping rules are applied in the presented order, an ontology is created incrementally. In the following, we refer to an example schema that models some university. The example is depicted in figure 3. Table 1 shows the translated schema in F Logic.

Concept	Relation	Predicate & Axiom

Object[].
Student::Object.
PhDStudent::Student.
School::Object.
Staff::Object.
Course::Object.
CourseTermin::Object.
CourseMaterial::Object.
Quarter::Object.
Offering::Object.
	Student[
	studID=>>Number;
	givenname=>>String;
	familyname=>>String;
	schoolID->>School;
	courseID=>>Course].
	PhDStudent[
	year=>>String].
	School[
	faculty=>>String;
	studID=>>Student
	…
	].
	…
		Key(Student, studID).
		NotNull(Student, familyname).
		Inverse(Course, studID, Student, courseID).
		MustExists(Offering, lecturerID).
		…
		Forall C1,R1,C2,R2 Inverse(C1,R1,C2,R2) <-
		Forall IR1, IR2 EXISTS IC1 IC2 IR1:C1 and IR2:C2 and IC1:C1 and IC2:C2 and
		(Forall IC1[R1->>IR2]<-IC2[R2->>IR1]) and
		(Forall IC1[R2->>IR1]<-IC1[R1->>IR2]).

Table 1. Created ontology

2.4.1 Rules for concepts



Relational schema

STUDENT (studID, givenname, familyname, schoolID)
PhDSTUDENT (studID, year)
SCHOOL (schoolID, faculty)
STAFF (lecturerID, room)
STAFF-DETAIL (lecturerID, address)
COURSE (courseID, subject, equip)
COURSE-TERMIN (courseID, tnumberID, date)
COURSE-MATERIAL (courseID, mnumberID, version)
ENROL (courseID, studID)
QUARTR (quarterID, year)
OFFERING (lecturerID, courseID, quarterID)

Figure 3. Relational example schema

The general assumption is that each relation is converted into a concept. There are two exceptions described in the rules C1 and C2. Rule C1 treats relations that are only created to express (n:m) relationships between some other relations. In this case the relation contains only a pair of foreign-key attributes (relation ENROL in Fig. 3). Rule C2 integrates information about a particular entity (STAFF) that is spread across several relations (STAFF and STAFF-DETAILS) and can be integrated into one concept. Rule C3 is the default rule for concept creation. If the other rules can not be applied a relation is converted into a concept.

2.4.2 Rules for inheritance

Extracting inheritance relationship from a relation schema usually requires behavioural information. Thus, only one rule can be applied to the schema. This rule creates an inheritance relationship if an inclusion dependency between two relations exists and concepts for both relations. This rule is a specialization of C2. In general the user has to decide whether C2 or I1 is applied.

2.4.3 Rules for relations

There are several possibilities to create ontological relations from a relational schema. Again we have a default rule (A6) that is applied if no other rules for attributes can be applied. In this case all attributes are converted to ontological relations.

First relation creation rule corresponds to the rule C1 and treats many-to-many relationship. It is about a relation where all attributes are a concatenation of primary keys of two other relations. These attributes are called foreign keys and they are converted to multi-valued concept references. The relation ENROL has only key attributes where one subset *studID* references a relation STUDENT and the other subset *courseID* references a relation COURSE. Multi-valued relations *studID* and *courseID,* which are mutually inverse (Inverse(Course, studID, Student, courseID)) are attached to their corresponding concept.

Second rule treats one to many relationships. Again key references are converted to concept references. In distinction to A1 on the "one" side only an ontological relation with a single-valued concept reference is created. For example, a relation STUDENT has a foreign key *schoolID* referencing the primary key *schoolID* of the relation SCHOOL. A single-valued relation *schoolID* of type School, is attached to the Student concept. A multi-valued relation *studID* is attached to the concept School.

Next rule corresponds to C2 and groups the attributes that are distributed in several relations. We also introduce the rule that handles one-to-one relationships. In this case both foreign keys are converted to single-valued conceptual relations.

Non-binary relationships representing relationships of higher degrees (non-binary relation) must be transformed into further concepts and a set of the binary relations. RDF does not provide means for n-ary relations, therefore complex relations are decomposed into a group of several binary relations. For example, the relation OFFERING denotes a relationship between lecturers, courses, and quarters. For each involved database relation a new relation is added to the concept Offering connecting the corresponding concepts, which are also connect with the concept Offering. We need further semantics for this kind of relations, that ensure that the group of binary relations only exists as a whole.

2.5 Data Migration

Once the ontology is created, the process of *data migration* can start. The objective of this task is the creation of ontological instances (that form a knowledge base) based on the tuples of the relational database.

The data migration process has to be performed in two phases:

1. In the first phase the instances are created. To each instance is assigned a unique identifier. This translates all attributes, except for foreign-key attributes, which are not needed in the metadata.

2. In the second phase, relations between instances are established using the information contained in the foreign keys in the database tuples. This is accomplished using a mapping function, that maps keys to ontological identifiers. Figure 5 illustrates an example result of the data migration process.

In our particular implementation the Ontobroker F-logic inference engine [4] is used. Ontobroker's API function *dbaccess* enables the generation of the instances from given relational database on the fly.

In the next section we describe a tool-support for the (semi-)automation of the presented mapping process and its application in a real-world case-study.

3. AIFBNET CASE STUDY

Case study concerns Intranet of our Institute AIFB (www.aifb.uni-karlsruhe.de), that in the first version was a database application. In order to enable processing of the content of our Intranet by machine agent we decided to describe semantic of the content explicitly and formally. To make the new application more sharable (for the research Institute community) we decided to create a domain ontology – the benefits of a such approach for information integration in the community are obvious.

This ontology is developed independently from the relational model of our database application, while it should sublimate community demands and not only our particular instantiation. This version of the ontology is still the first one and our particular view is still dominant, as careful reader might note. The ontology is developed using OTK methodology and the process and accumulated experiences are described in [15].

Our DB-based application contains real data about researchers, projects and the institute. In order to avoid redundant work we need to map the content of the database in the knowledge base of the ontology and that was the primary reason to develop some mechanism, which will support this mapping. The mapping process is described in previous section and we present here a tool-support for that process.

For the automation of the mapping process we used KAON-REVERSE, a tool for semi-automatically connecting relational database to ontologies, implemented as a plug-in for KAON ontology environment (http://kaon.semanticweb.org/).

KAON–REVERSE supports some phases in the adapted mapping process, particularly capturing information from the relational schema, validation of the mapping process and data migration. For step 2 KAON-REVERSE generates some recommendations based on the mapping rules defined in the section 2, as will be presented in the following text.

Alignment of the top level terms is very difficult and can not be completely automated. Some guidelines could be proposed comparing entities' names on the syntax level [10]. In the case that ontology definition contains lexical layer [11] (labels, synonyms, stamps) this string matching process can be performed more efficiently.

KAON-REVERSE supports the alignment of top level terms using lexical layer of the AIFB ontology. For example relation “Projekt” is mapped into the concept “Project”, while one of the defined synonyms for the concept “Project” is “Projekt” (in German). The output of this phase (Figure 4) is a set of the mapping pairs in the form {(name_of_relation, name_of_concept)}.

As already mentioned, KAON-REVERSE supports validation step, in which is discovered, for example, that the relation “Beschreibung” is not mapped into any concept. After the analysis of the structure and the content of this relation, it is determined that this relation has neither foreign keys nor other relations have a foreign key to this relation. However, the connection to the other relations is hard-coded and can be discovered only by analysis of the content of this relation. That is the reason why we have made new heuristic rules for concept and attribute creation, which are based on comparing the content of a relation with the given ontological structure. The result of this phase is an extension of the set of mapping pairs. For example, in this phase is discovered that the relations “Projekt_FG”, “Projekt_Person”, “Projekt_Gruppe” correspond to the concept “Projekt”.

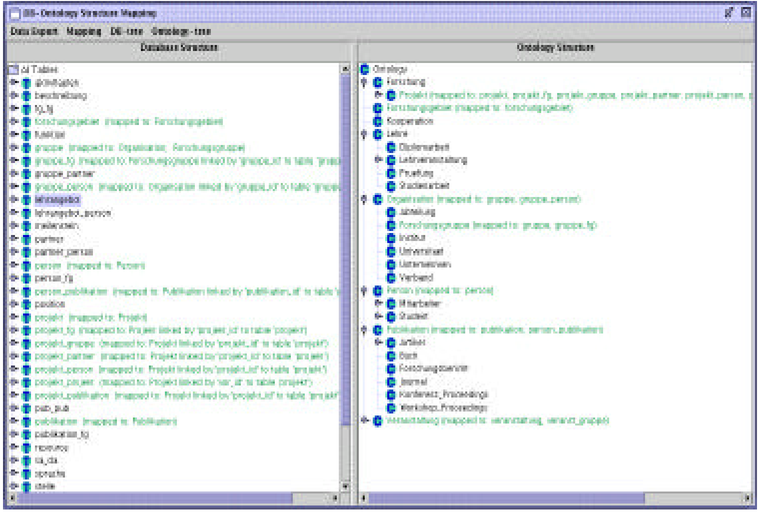


Figure 4. Mapping relations to the concepts using KAON-REVERSE

Figure 5 shows how KAON-REVERSE makes recommendations for assigning relation’s attributes to the attributes of a concept. On the left side of the figure is depicted structure of the relational schema. The right side shows the AIFB ontology. Highlighted entities are mapped to each other. Recommendations for mapping attributes are listed in the middle window.

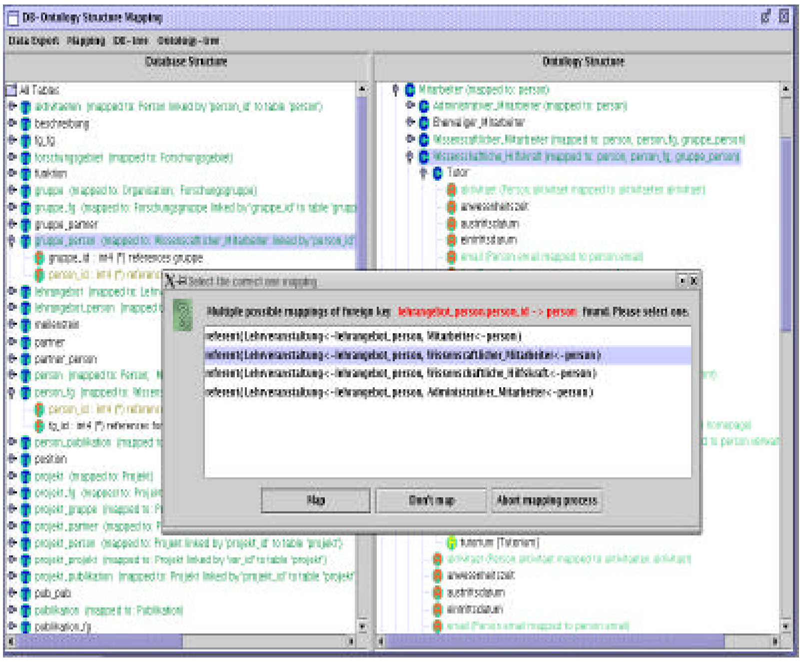
The data migration process is performed according to guidelines described in the section 2.5. In the first phase the object identifiers for instances are created and all the attributes with range primitive data types are instantiated. In the second phase relations between instances are instantiated, using mapping function which maps tuple-identifier from database into instance identifier. An illustration of the data migration process and the mentioned phases is presented in the Figure 6.

The process of migrating data from database to knowledge base is straightforward, but there are situations where some heuristics should be applied in order to map semantic of the content of the relations into the knowledge base.

3.1 Lessons learned

These remarks are the result of applying our mapping process in the real-world case studies and help in forming heuristic mapping rules, which will be implemented in the next version of software.

- The order of applying the transformation rules is very important. At first, for every relation in the relational schema we tried to apply rules for the concept creation. It results in an ontology consisting of concepts and is-a hierarchy. These rules are applied in the established order: first C1, then C2 and at the end C3. Moreover, the order in which we analyse relations is also important. First and foremost we test relations with only one primary key. In the next step we check relations with one key more until there are no remained relations.



Multiple possible mappings of foreign key lehrangebot_person.person_id -> person found. Please select one.

referent(Lehrveranstaltung<-lehrangebot_person, Mitarbeiter<-person)
referent(Lehrveranstaltung<-lehrangebot_person, Wissenschaftlicher_Mitarbeiter<-person)
referent(Lehrveranstaltung<-lehrangebot_person, Wissenschaftliche_Hilfskraft<-person)
referent(Lehrveranstaltung<-lehrangebot_person, Administrativer_Mitarbeiter<-person)

Figure 5. A screenshot of KAON-REVERSE

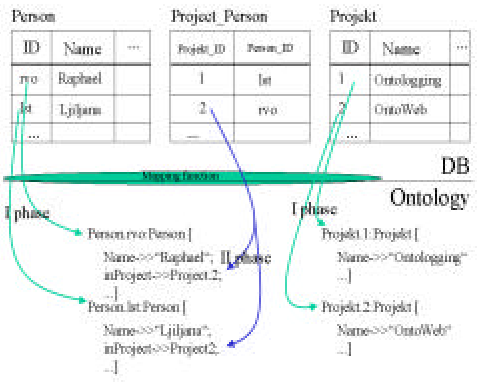


Figure 6. The data migration process – An example

- As already emphasised is the section 2, the reverse engineering process cannot be completely automated. User interaction is necessary while ambiguities occur and domain semantics cannot be inferred. For example, the rule I2 is a specialisation of the rule C2.

- After applying the concept creation rules, the proposed procedure tries to transform attributes of relations into the corresponding ontological entities.

For example, certain attributes of relations are candidates for choosing the attributes of a concept in the ontology or foreign key dependency can be transformed to the relation between two concepts in the ontology. These rules also create axioms that introduce some constraints into a concept description. It is not necessary to complete the set of concepts and concept hierarchy before creating attributes, but the corresponding concept (for an attribute) has to exist.

- One exception for ontological relation creation can be the existence of a local key in the relation schema that is used to maintain internal consistency of the data across various tables. It is possible that the local keys are abstract out and not represented in the domain ontology in the case that users don't want to create queries involving them. If the user decides to retain the keys, than for these attributes the key predicate has to be added into the ontology.

- The case that an attribute of a relation has value NULL is treated as a special case: mapping procedure does not assign any value to the corresponding attribute of the ontological instance.

- There are also various "irregular situations" in the structure of the relational schema. For example, a relation that corresponds to a concept, has no primary key. This induces a lot of problems in the mapping process (how to generate Ids, the problem of the consistency of data). An example is depicted in the Figure 7. Such kind of the problems is not treated in the presented mapping approach and is resolved per hand, but that will be challenge for the future work.

The presented user scenario could be generalized for each community. The only prerequisite is that the common-shared community ontology exists. In the absence of the community ontology, our mapping process can be slightly modified and used for generating such one from the existing relational schema. But, what is really needed is the shared agreement about that ontology.

4. RELATED WORK

As known to the authors, there is no approach that integrates aspects of reverse engineering and integration platforms for the metadata on the (Semantic) Web, so those topics will be discussed separately.

Database reverse engineering

There are very few approaches investigating the transformation of a relational model into an ontological model. The most similar approach to our approach is the project Infosleuth [9]. In this project an ontology is built based on the database schemas of the sources that should be accessed and thereafter it is refined based on user queries. However, there are no techniques for creating axioms, which are a very important part of an ontology. Moreover, the semantic characteristics of the database schema are not always analysed.

More work has been addressed on the issue of explicitly defining semantics in database schemas [3], [14], extracting semantics out of database schema [3], [8] and transforming a relational model into an object-oriented model [2], which is close to an ontological theory. Rishe [14] introduces semantics of the database as a “means” to closely capture the meaning of user information and to provide a concise, high-level description of that information. In [3] an interactive schema migration environment that provides a set of alternative schema mapping rules is proposed. In this approach, which is similar to our approach on the conceptual level, the re-engineer repeatedly chooses an adequate mapping rule for each schema artefact. However, this stepwise process creates an object-oriented schema, therefore axioms are not discussed.

Information integration

In the broader sense our approach could be treated as an information integration approach [16], [12], while we provide a platform for resolving semantic problems that arise in this integration process. The information mediation architecture contains intermediate layer that mediates between human user and information sources. That layer has dual structure: the mediator deals with the human user and the wrappers deal with the information sources [6]. In our approach information sources have uniform structure, so that the role of wrapping is dedicated to the mapping process. In [5] is proposed an approach for reconciling XML data, based on intermediate conceptual models. In this case, a human expert is needed to reverse-engineer the underlying conceptual model for a XML schema, and to specify formally how the original schema maps onto the corresponding conceptual model. Our approach provides the guidelines how to make such kind of reverse engineering, so that some phases can be automated and user intervention avoided.

5. CONCLUSION

The Semantic Web is the new WWW infrastructure that will enable machine processibilty of the web content and seems to be a solution for many drawbacks of the current Web. As formal and common-shared semantics for information must be created, the migration of the existing Web applications into the Semantic Web is a research challenge. This (formal and real-world) semantic is provided by means of domain ontologies.

We have developed a novel, integrated and semi-automated approach for migrating data-intensive Web applications into the Semantic Web that can be applied to a broad range of today's business Web sites.

The approach starts with transforming the relational database model into corresponding ontological structures, which is then used for mapping the content of the database into an ontology-based knowledge base. The facts can be transformed in RDF statements and published on the web.

Publishing such statements on the web makes content of the web pages (in the last instance also the content of a given database) public and machine understandable, what is the prerequisite to achieve the Semantic Web. As a side effect, the issue of making the semantics of a database more explicit is also addressed and our approach leads to more formal semantics that could be used, for example, for the maintenance of data-driven applications and also in the simplified migration to other databases.

Moreover, the presented scenario can be very popular in the e-commerce domain, where the using of ontologies as mediation level for product data exchange is already proposed [13] and our approach could be a mechanism for resolving semantic problems that arise in this integration process.

The benefits of the proposed approach are manifold: The process of providing metadata is automated and thus inexpensive and fast. The problem of dynamic updating metadata according to changes in corresponding web pages is also resolved. Moreover, since, from the database generated, metadata can be too long, additional semantic annotation using tools like Onto-O-Mat can be very useful for ontology envolvement. For example, new annotated text can be candidate for the new ontological entities. The most important benefit is that information from various community members could be exchanged on the semantic bases. This paves the way to unleash the full-power of the Semantic Web.

6. ACKNOWLEDGEMENTS

The research presented in this paper would not have been possible without our colleagues and students at the Institute AIFB, University of Karlsruhe, and FZI, Karlsruhe. We thank out student Gert Pache who implemented KAON-REVERSE. Research for this paper was partially financed by EU in the IST-funded projects “OntoLogging” and “OntoWeb”.

7. REFERENCES

[1] Abiteboul S., Hull R., Vianu V. (1995) *Foundation of databases*, Addison-Wesley Publishing Company.

[2] Behm A., Geppert A., Dittrich, K. (1997) On the Migration of Relational Schemas and Data to Object-Oriented Database Systems. *In Proceeding of the 5th Int. Conference on Re-Technologies for Information Systems (Klagenfurt, December 1997), pp. 13-33, 1997.*

[3] Chiang R., Barron T., Storey V. (1994) Reverse engineering of relational databases: Extraction of an EER model from a relational database. *Journ. of Data and Knowledge Engineering, Vol. 12, No. 2, March 1994, pp. 107–142., 1994.*

[4] Decker S., Erdmann M., Fensel D., Studer R. (1999) Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. *In R. Meersman et al., editors, Database Semantics: Semantic Issues in Multimedia Systems, pages 351–369. Kluwer Academic Publisher, 1999.*

[5] Decker S., van Harmelen F., Broekstra J., Erdmann M., Fensel D., Horrocks I., Klein M., Melnik S. (2000) The Semantic Web - on the Roles of XML and RDF. *In IEEE Internet Computing. September/October, 2000.*

[6] Fensel D. (2001) Ontologies: Dynamic Networks of Meaning. *SWWS-1, Stanford, July 2001.*

[7] Kifer M., Lausen G., Wu J. (1995) Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal ACM, 42:741–843, 1995.*

[*] Handschuh, S., Staab, S., Maedche, A. CREAM — Creating relational metadata with a component-based, ontology-driven annotation framework. ACM K-CAP 2001. October, Vancouver.

[8] Hainaut J.L. (1991) Database Reverse Engineering, Models, Techniques and Strategies. *10th Conf. on ER Approach, San Mateo (CA), 1991.*

[9] Kashyap V. (1999) Design and creation of ontologies for environmental information retrieval. *12th Workshop on Knowledge Acquisition, Modeling and Management (KAW'99), Banff, Canada, October 1999.*

[10] Levenshtein I. V. (1966) Binary Codes capable of correcting deletions, insertions, and reversals. *Cybernetics and Control Theory, 10(8):707–710, 1966.*

[11] Maedche A. (2002) Ontology Learning Supporting Emergent Semantics. *To appear in IEEE Intelligent Systems - Trends & Controversies, 2002.*

[12] Melnik S., Decker S. (2000) A Layered Approach to Information Modeling and Interoperability on the Web. *In Proceedings of the Workshop "ECDL 2000 Workshop on the Semantic Web", 2000.*

[13] Omelayenko B. Fensel D. (2001) An Analysis of B2B Catalogue Integration problems: Content and Document Integration. *In Proceedings of the International Conference on Enterprise Information Systems (ICEIS-2001), 2001.*

[14] Rishe N. *Database Design: The Semantic Modeling Approach.* McGraw-Hill, 1992.

[15] Stojanovic N., Maedche A., Staab S., Studer S., Sure Y. (2001) SEAL — A Framework for Developing SEmantic PortALs. *ACM K-CAP-2001, 2001.*

[16] Wache H., Vögele T., Visser U., Stuckenschmidt H., Schuster G., Neumann H., Hübner C. (2001) Ontology-Based Integration of Information - A Survey of Existing Approaches, *In Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing*, pp 108-118, Seattle, USA, August 4-5, 2001.

[17] Berners-Lee T. (2000) XML 2000 – Semantic Web talk, *http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html, 2000.*