Relational Database Migration: A Perspective

Abdelsalam Maatuk, Akhtar Ali, and Nick Rossiter

School of Computing, Engineering & Information Sciences, Northumbria University, Newcastle upon Tyne, UK

Abstract. This paper presents an investigation into approaches and techniques used for database conversion. Constructing object views on top of a Relational DataBase (RDB), simple database integration and database migration are among these approaches. We present a categorisation of selected works proposed in the literature and translation techniques used for the problem of database conversion, concentrating on migrating an RDB as source into object-based and XML databases as targets. Database migration from the source into each of the targets is discussed in detail including semantic enrichment, schema translation and data conversion. Based on a detailed analysis of the existing literature, we conclude that an existing RDB can be migrated into object-based/XML databases according to available database standards. We propose an integrated method for migrating an RDB into object-based/XML databases using an intermediate Canonical Data Model (CDM), which enriches the source database's semantics and captures characteristics of the target databases. A prototype has been implemented, which successfully translates CDM into object-oriented (ODMG 3.0 ODL), object-relational (Oracle 10g) and XML schemas.

1 Introduction

Object-oriented and Web technologies have become mainstream due to their productivity, flexibility and extensibility [9,10]. The dominance of traditional RDB and its limitation to support the benefits provided by these new technologies motivate its migration into Object-Oriented DataBase (OODB), Object-Relational DataBase (ORDB) and XML [1,2,11,12]. This paper aims to provide an investigation into the problem of DataBase Migration (DBM), to review various techniques and proposals, to identify their differences, and to assess the impact of existing literature and how it shapes current and future research in this area. We focus on the case where the input is an RDB and the outputs are OODB, ORDB and XML. Hence, we do not cover the inverse of the process (e.g., migrating OODB into RDB). Many proposals exist in the literature for handling data stored in RDBs through Object-Oriented (OO)/XML interfaces, i.e., Object-to/from-Relational (OR) and Xml-to/from-Relational (XR) mapping, connecting an existing RDB into non-RDB system that might be conceptually different, and migrating an RDB into other databases [9,15,8,16,19]. New requirements of database systems determine which technique is most suitable to adopt.

Database application migration is a process in which all components (i.e., schema, data, application programs, queries and update operations) of a source

S.S. Bhowmick, J. Küng, and R. Wagner (Eds.): DEXA 2008, LNCS 5181, pp. 676–683, 2008.

database application are converted into their equivalents in a target database environment. However, application programs and queries conversion is a software engineering job and is, therefore, out of the scope of this paper, i.e., we assume that DBM includes schema translation and data conversion. A schema of an existing data model can be translated into an equivalent target schema expressed in the target data model through applying a set of mapping rules. The translation of a source schema to a target schema consists of two subphases. The first one, called DataBase Reverse Engineering (DBRE), aims to recover the conceptual schema, e.g., Entity Relationship Model (ERM), which expresses explicit and implicit data semantics of the source schema. Explicit semantics involves relation, attributes, keys and data dependencies. It is necessary to extract extra semantics that are not expressed explicitly in RDBs (e.g., relationships). The second phase, called DataBase Forward Engineering (DBFE), aims to obtain the target physical schema from the conceptual schema obtained in the first phase. However, source schema can be translated directly to a target one without intermediate representation [8]. An expert user or a tool might be required to provide missing semantics or to refine the result to exploit the target database concepts [16,8]. Data Conversion is a process of converting data from the source into the target database. Data stored as tuples in an RDB are converted into complex objects/literals in object-based databases or elements in XML document. This involves unloading and restructuring relational data, and then reloading it into a target database in order to populate the schema generated earlier during the schema translation process [9].

The remainder of this paper is organised as follows. Section 2 surveys current approaches and techniques related to database conversion. In Section 3, DBM proposals, the focus of this paper, are discussed in detail. Section 4 concludes the paper.

2 Approaches and Techniques

2.1 Conversion Approaches

There are three approaches related to database conversion. The first approach is for handling data stored in RDBs through OO/XML interfaces. Connecting an existing RDB to a conceptually different database system is the second approach. The third approach is migrating an RDB into a target database. The first and second approaches deal with schema translation, whereas in the third approach, both schema and data are completely migrated into a target database.

Viewing Objects on Top of RDBs: Data may be required to be processed in object/XML form and stored in relational form based on the concept of object for programs and RDB for persistence. A single object might be represented by several tuples in several tables, therefore, joining these tables is required for queries. However, converting these objects to tabular forms to be stored in and retrieved from RDB systems leads to a semantic gap between two different paradigms, which is known as the OR impedance mismatch. To avoid this, developers have to write huge amount of code to map objects in programs into tuples in an

RDB, which might be time-consuming to write and execute. Another solution is via using OR mapping middleware, which is a software layer that links OO Programming Languages (OOPLs) concepts to data stored in RDBs through ODBC or JDBC drivers. Similarly, RDBs data can be published as XML documents using special declarative languages to be exchanged over the Web, with which users see views that can be queried using XML query languages. However, mapping using middleware requires schema mapping time.

Database Integration: A connection could be established between RDBs and other databases allowing the applications built on top of new DBMS accessing both relational and object/XML DBMSs giving an impression that all data are stored in one database. This presents a simple level of DataBase Integration (DBI). This is achieved using a special type of software called *Gateways*, which support connectivity between DBMSs and do not involve the user in SQL and RDB schema. Hence, queries and operations are converted into SQL and the results are translated into target objects. Most commercial DBMSs provide flexibility on gateways construction among heterogeneous databases. The difference between Gateways and mapping tools is that, in Gateways, objects are persistently stored in the new target database system, whereas in the mapping, objects are created and handled in the normal way but are stored in an RDB. However, in both approaches old data, stored in an RDB, is retained.

Database Migration: Migration of an RDB into its equivalents is accomplished in the literature for two databases. The first database is an RDB, the *source*, and the second database, the *target*, represents the result of DBM process. The process is performed with or without the help of an Intermediate Conceptual Representation (ICR), e.g., ERM. The input source schema is enriched semantically and translated into a target schema. Generally, relations and attributes are translated into equivalent targets. Foreign Keys (FKs) may be replaced by another domain or relationship attributes. Relationships can be extracted by analysing data dependencies or database instances. Data stored in the source database is converted into the target database. FKs realise relationships between tuples, which are converted into value-based or object identifier references. The challenge in this process is that data of one relation may be converted into a collection of literal/references rather than into one corresponding type. This is because of the heterogeneity between the concepts and structures of source and target data models.

2.2 Translation Techniques

Existing techniques can be classified into two types: Source-to-Target (ST) and Source-to-Conceptual-to-Target (SCT).

ST Technique: This type of technique translates a physical source code into an equivalent target. However, as the target schema is generated using one-step mapping with no ICR for enrichment, this technique usually results in an ill-designed database as some of the data semantics are ignored. This approach could take the following forms:

Flat Technique: This technique converts each relation into object class/XML element in target database [9,16]. FKs are mapped into references to connect objects. However, the flattened form of RDBs is preserved in the generated database, with which object-based model features and the hierarchical form of XML model are not exploited. This means that the target database is semantically weaker and of a poorer quality than the source. Moreover, creating too many references cause degraded performance during data retrieval.

Clustering Technique: This technique is performed recursively by grouping entities and relationships together starting from atomic entities to construct more complex entities until the desired cluster is achieved, which is labelled with the strong entity name [19]. However, this technique may lead to complex structures, data redundancy and is prone to error in translation.

Nesting Technique: This technique uses the iterated mechanism of *nest* operator to generate a nested target structure from relational inputs [7]. The target is extracted from the best possible nesting outcome. However, the technique has some limitations, e.g., mapping each table separately and ignoring integrity constraints. Besides, the process is quite expensive, as it needs all tuples of a table to be scanned repeatedly to get the best possible nesting.

SCT Technique: This type of technique enriches a source schema by semantics that might not have been clearly expressed in it and their interrelationships. Then, the schema is translated from logical into conceptual through recovering the domain semantics and making them explicit. Finally, the results are represented as a conceptual schema, which can be translated into the target effectively. In this way the technique results in a good well-designed target database. Inferring conceptual schema from a logical RDB schema has been extensively studied by many researchers based on analysing schema, data and queries. Chiang *et al.* presented a method for extracting an Extended ERM (EERM) from an RDB [5] through derivation and evolution of key-based inclusion dependencies. Alhajj developed algorithms for identifying candidate keys to locate FKs in an RDB using data analysis [1]. Andersson extracts a conceptual schema by investigating equi-join statements [3]. The approach uses a join condition and the *distinct* keyword for attribute elimination during key identification.

3 Migrating RBD into OODB/ORDB/XML

This section discusses proposals for migrating RDBs into OODB/ORDB/XML databases. Table 1 shows a comparison of some proposals showing input, output, technique used, data semantics, prerequisites and features of DBM process.

Migrating RBD into OODB: Several methods have been proposed for migrating RDBs into OODBs without using an ICR [16,4,8,9]. Premerlani and Blaha propose a procedure for mapping an RDB schema into an OMT schema [16]. They produce an initial schema and determine Primary Keys (PKs) and FKs by resolving synonyms and homonyms. Then, horizontally partitioned classes are refined, and relationships are identified using keys evaluation. Fahrner

Table 1. RDB migration (prerequisites, features, input and output databases)

Proposal	ST	DC	Tec	Data Semantics					Input	Prerequisites	Features		Output		
				AS	AG	IN	RI	OP			SA	UI	OODB	ORDB	XML
[9]	√	√	ST	√	×	√	×	×	RDB	FD,ID,ED	×	H	√	×	×
[8]	√	×	ST	√	√	√	√	√	RDB	keys,FD,ID,3NF	√	H	√	×	×
[2]	√	√	SCT	√	√	√	×	×	RDB	keys,DD,Ins	×	L	√	×	×
[15]	√	×	ST	√	√	×	×	×	ERM	ERM	×	H	√	×	×
[16]	√	×	ST	√	√	√	×	√	RDB	keys,non-3NF	×	H	√	×	×
[4]	√	×	ST	√	√	√	×	×	RDB	FD,ID,ED,non-3NF	×	H	√	×	×
[18]	√	×	ST	√	√	×	√	√	UML	UML class diagram	×	-	×	√	×
[14]	√	×	ST	√	√	√	√	√	UML	UML class diagram	√	-	×	√	×
[10]	√	√	SCT	√	√	√	√	×	RDB	PKs,FKs	√	L	×	×	√
[12]	√	√	ST	√	×	×	×	×	EERM	FD,ID	√	H	×	×	√
[6]	√	×	SCT	√	√	√	√	×	RDB	3NF	√	H	×	×	√
[11]	√	√	SCT	√	√	×	×	√	RDB	FD,MVD,JD,TD	√	L	×	×	√
[7]	√	×	ST	√	√	×	×	√	RDB	PKs,FKs	√	H	×	×	√

ST: Schema Translation DC: Data Conversion MVD: Multi-valued Dependency TD: Transitive Dependency Ins: Data instances FD: Functional Dependency ED: Exclusion Dependency ID: Inclusion Dependency JD: Join Dependency UI: User Interaction SA: Standard Adoption L: Low consideration H: High consideration Tec: Technique AS: Association AG: Aggregation IN: Inheritance RI: Referential Integrity OP: Optimization √: Yes ×: No

and Vossen propose a method in which an RDB schema is normalised to 3NF, enriched by semantics using data dependencies and translated into an ODMG-93 OODB schema [8]. Moreover, the resulting schema is then restructured (by the user) with respect to OO paradigm options, e.g., binary relationship relations are eliminated and integrity constraints are mapped into class methods. Castellanos *et al.* present a method that improves an RDB schema semantically (by analysing the schema and data) and converts it into an object-based schema, called BLOOM schema [4]. Narasimhan *et al.* propose a procedure for mapping an ERM into an OO schema [15]. The approach suggests creating a separate constraint class as a subclass for each of OODB classes. Yan and Ling present a method that produces an OODB schema from an RDB using clustering technique [19]. A cluster of relations is identified from a main relation and its component/subclass relations, which are not participating in relationships with other relations. Besides, the method proposes generating OIDs for identified objects by concatenating the key of each tuple with the relation name. Alhajj and Polat re-engineer an RDB into an OODB using an RID graph as an ICR [2]. The graph, which is similar to EERM is derived and optimised for identifying relationships. Finally, RDB tuples are migrated into objects in OODB.

Migrating RDB into ORDB: A number of researchers have considered exploiting user-defined types in Oracle $8_i/9_i$ and SQL3 from conceptual models [18,14]. The logical structure of an ORDB schema is achieved by creating object-types using UML, based on which tables are created to store data. Multi-valued attributes are defined using arrays. An association relationship is mapped using REF/collection of REFs. An inheritance is defined using FKs or REF types in Oracle 8_i and using the UNDER clause in Oracle 9_i/SQL3 [14]. Although most ORDB concepts are presented in these proposals, they are aimed at producing an ORDB schema from conceptual models rather than DBM. However, if a DBM

process uses a conceptual model as an ICR then these proposals could be useful in schema translations.

Migrating RDB into XML: Fong and Cheung introduce a method, in which data semantics are extracted from an RDB into an EERM, which is then mapped into an XSD graph. An XML logical schema is extracted from the XSD graph [10]. The authors suggest mapping FKs into element hierarchy, which may cause redundancy when an element has a relationship with more than one element. Kleiner and Lipeck translate an ERM to DTD [12]. However, some data semantics cannot be represented, e.g., the limitation of DTD in specifying composite keys. Vela and Marcos propose an approach for extending UML to represent an XML Schema in graphical notation, which has a unique equivalence with XML Schema [20]. Du *et al.* propose translation rules for converting an enriched RDB schema into a semi-structured model, called ORA-SS, which is then translated into XML Schema [6]. However, they adopt an exceptionally deep clustering technique, which is prone to errors. Fong *et al.* propose a procedure to translate RDB views into XML documents [11]. The approach de-normalises an RDB into joined tables and translates them Document Object Models (DOMs), which are integrated into one DOM, which is then mapped into an DTD schema. Based on the generated DTD schema and data dependencies, each tuple of the joined tables is loaded into an instance in DOM and then transformed into an DTD document. Lee *et al.* present two algorithms, called NeT and CoT, to translate an RDB schema to DTD using a language named XSchema [7]. The CoT algorithm is proposed to remedy the drawbacks of NeT, e.g., the mapping of each table separately and not taking into account integrity constraints.

4 Discussion

In this paper, we have presented a survey of existing approaches and techniques used for database conversion. Our investigation into DBM problem shows that different proposals have different focuses. Each proposal has some assumptions to facilitate the process, which might be a point of limitations or a drawback. While existing works for migrating into OODBs focus on schema translation using ST techniques, we note that most works for migrating to XML are following SCT techniques, focusing on generating a DTD schema and data. Moreover, all researches on the generation of ORDBs are focused on design rather than migration. It could be concluded, based on our analysis of the literature, that there are several areas in need of more attention for migrating RDBs to object-based/XML databases.

Due to focusing on schema rather than data, proposals either ignore data loading or assume working on virtual target databases and data remain stored in RDBs. Moreover, there are still shortcomings in implementation of loading an RDB data to more than one environment. Using middleware may lead to slow performance making the process expensive at run-time because of dynamic mapping of tuples to complex objects. However, using object-based DBMSs and

native-XML, objects can be stored and retrieved directly without any need for translation layers, hence saving development time and improving performance.

Some semantics (e.g., inheritance) have not been considered during DBM. ERM and DTD do not support inheritance. Despite UML's ability to model data semantics such as aggregation and inheritance, UML is still weak to handle the hierarchical structure of the XML data model [10]. UML should be extended by adding new stereotypes to specify ORDB and XML models features [14,20]. Although generalization/specialzation and categorization could be realized in an RDB, they have been either ignored or briefly mentioned without delving into its different types, e.g., union and multiple inheritance, and its constraints. Translating inheritance relationships from RDBs to object-based/XML databases needs more attention. Standard adoption is essential for more portability and flexibility. In the ODMG 3.0 model, referential integrity is maintained via inverse references. SQL4 has an ability to address complex objects in ORDBs. Compared to DTD, XML Schema offers a much more extensive set of data types, and provides a powerful referencing, nesting and inheritance mechanisms of attributes and elements.

Most of the existing proposals and techniques generate a database that is either flat relational or has a deep level of clustering/nestng. It would be desirable to avoid the flattened form and reduce clustering levels of objects structure to the lowest in order to increase utilisations of advantages that target models provide and to avoid undesirable redundancy. This requires preservation of semantics of the source database and relocating them into an ICR, which takes into account the relatively richer data model of the target database.

The Way Forward: The existing work does not provide a solution for more than one target database or for either schema or data conversion. Besides, none of the existing proposals can be considered as a method for migrating an RDB into an ORDB. Several challenges could arise when a DBM process aims at several target databases, which are fundamentally different and have different design characteristics. An integrated method, which deals with migration from RDB to OODB/ORDB/XML covering both schema and data is not yet in existence. We propose a complete method [13], which is able to preserve the structure and semantics of an existing RDB in a CDM, to generate OODB/ORDB/XML schemas, and to find an effective way to load data into target databases without lose or unnecessary redundancies. The method is superior to the existing proposals as it can produce three different output databases. Besides, the method exploits the range of powerful features that target data models provide such as ODMG 3.0, SQL4, and XML Schema. A system architecture is designed and a prototype has been implemented, which resulted successfully in target databases.

References

1. Alhajj, R.: Extracting the Extended Entity-Relationship Model from a Legacy Relationals Database. Info. Syst. 28, 597–618 (2003)
2. Alhajj, R., Polat, F.: Reengineering Relational Databases to Object-Oriented: Constructing the Class Hierarchy and Migrating the Data. In: WCRE 2001, pp. 335–344 (2001)

3. Andersson, M.: Extracting an Entity Relationship Schema from a Relational Database through Reverse Engineering. In: 13th Int. Conf. on the ER Approach, pp. 403–419 (1994)
4. Castellanos, M., Saltor, F., García-Solaco, M.: Semantically Enriching Relational Databases into an Object Oriented Semantic Model. In: Karagiannis, D. (ed.) DEXA 1994. LNCS, vol. 856, pp. 125–134. Springer, Heidelberg (1994)
5. Chiang, R.H., Barron, T.M., Storey, V.C.: Reverse Engineering of Relational Databases: Extraction of an EER Model from a Relational Database. Data Knowl. Eng. 12, 107–142 (1994)
6. Du, W., Lee, M., Ling, T.W.: XML Structures for Relational Data. In: WISE (1), pp. 151–160 (2001)
7. Lee, D., Mani, M., Chiu, F., Chu, W.W.: NeT and CoT: Translating Relational Schemas to XML Schemas using Semantic Constraints. In: CIKM, pp. 282–291 (2002)
8. Fahrner, C., Vossen, G.: Transforming Relational Database Schemas into Object-Oriented Schemas according to ODMG 1993. In: Ling, T.-W., Vieille, L., Mendelzon, A.O. (eds.) DOOD 1995. LNCS, vol. 1013, pp. 429–446. Springer, Heidelberg (1995)
9. Fong, J.: Converting Relational to Object-Oriented Databases. SIGMOD Record 26, 53–58 (1997)
10. Fong, J., Cheung, S.K.: Translating Relational Schema into XML Schema Definition with Data Semantic Preservation and XSD Graph. Info. & Soft. Tech. 47, 437–462 (2005)
11. Fong, J., Wong, H.K., Cheng, Z.: Converting Relational Database into XML Documents with DOM. Info. & Soft. Tech. 45, 335–355 (2003)
12. Kleiner, C., Lipeck, U.W.: Automatic Generation of XML DTDs from Conceptual Database Schemas. GI Jahrestagung (1), 396–405 (2001)
13. Maatuk, A., Ali, A., Rossiter, N.: A Framework for Relational Database Migration. TR (2008), `http://computing.unn.ac.uk/staff/cgma2/papers/RDBM.pdf`
14. Marcos, E., Vela, B., Cavero, J.M.: A Methodological Approach for Object-Relational Database Design using UML. Soft. and Syst. Modeling 2, 59–75 (2003)
15. Narasimhan, B., Navathe, S.B., Jayaraman, S.: On Mapping ER Models into OO Schemas. In: 12th int. Conf. on the Entity-Relationship Approach, vol. 823, pp. 402–413 (1993)
16. Premerlani, W.J., Blaha, M.R.: An Approach for Reverse Engineering of Relational Databases. Communications of the ACM 37, 42–49 (1994)
17. Soutou, C.: Inference of Aggregate Relationships through Database Reverse Engineering. In: Ling, T.-W., Ram, S., Li Lee, M. (eds.) ER 1998. LNCS, vol. 1507, pp. 135–149. Springer, Heidelberg (1998)
18. Urban, S.D., Dietrich, S.W., Tapia, P.: Succeeding with Object Databases: Mapping UML Diagrams to Object-Relational Schemas in Oracle 8, pp. 29–51. John Wiley and Sons, Ltd, Chichester (2001)
19. Yan, L., Ling, T.W.: Translating Relational Schema with Constraints into OODB Schema. In: The IFIP WG 2.6 Database Semantics Conf. on Interoperable Database Systems, vol. A-25, pp. 69–85 (1993)
20. Vela, B., Marcos, E.: Extending UML to Represent XML Schemas. In: CAiSE Short Paper Proceedings (2003)
21. Zhang, X., Zhang, Y., Fong, J., Jia, X.: Transforming RDB Schema to Well-structured OODB Schema. Info. & Soft. Tech. 41, 275–281 (1999)