Currently, a great deal of effort is being directed toward the development of heterogeneous database concepts in order to achieve system interoperability. This paper presents a conceptual model that was used in a study for developing an automatic means of translating one database structure into another with minimal human intervention. Our prototype—developed from this model—demonstrates the usefulness of an approach that would allow many of today’s systems to be more accessible when platform, syntax or semantic changes occur. This preliminary study is limited to the automatic conversion of one specific relational database into a different, yet still relational, database format. The paper concludes with a summary of findings in order to provide a foundation for future work on the use of the proposed model as a generic, extensible framework.

Intuitively, the process of converting an existing database system from one relational database management implementation to another entails two, well understood tasks. First, the schema model for the data stored on the existing system is evaluated and recreated on the target system, abiding by any prescribed mappings derived during the first task, such as type conversions. Various researchers (e.g., Batini et al., 1986; Bright et al., 1992; Thomas et al., 1990) have addressed these tasks and identified several problems worthy of future study, which served as the impetus for this study.

Initially the existing database must be represented in a format that lends itself well to analysis and translation. The original system and detailed design specifications could be used to derive the database structure and translate it to the target system. However, these documents are not always updated to reflect changes made during the implementation and maintenance phases. Further, such documentation often lacks adequate details about the implementation needed for an accurate translation. For example, the exact length of an attribute or its value constraints may be missing or incorrect.

Certainly, developers have the flexibility and comprehension needed for the required semantic conversion, but they also precipitate inconsistencies, incompleteness, and errors due to insufficient knowledge of the source and target database structures. The increase in use of powerful software such as ORACLE® and SYBASE®, which increase the current use of SQL in organizations, has resulted in a corresponding increase in the need for database conversion—not only from other software, but from older, disparate versions as well.

What is needed is an automatic means of translating one database structure into another with minimal human intervention. This would allow many of today’s systems to be accessible when platforms, syntax or semantics changes occur. One small area of this large problem that we have studied focuses on the automatic conversion of one relational database into another relational database format. All that is required is a physical schema that is in a relational format.

Previous research on multi-database management systems (e.g., Breitbart, 1990; Bright et al., 1992; Lu and Lan-xiz, 1993; Ram, 1991) has identified many of the issues in developing a conversion system. It is noteworthy that Kim and Seo
focused on the complexity issues alone. In the context of the database conversion system, functions defined by implementors of the source database processes could be linked to a translator process and executed whenever a target mapping process requests data from the source database. The functions would also provide conversion services, perhaps from the local representation to some standard representation defined by the conversion system. Sheth and Larson (1990) describe auxiliary schema, comprised of tables that can be used for translation tables and data format conversion information. Litwin et al., (1990) call standard data conversion methods, such as listing all possible conversions between data formats, the “brute force approach.” They explore methods that employ a self-describing data type in their discussion of multiple database interoperability.

The system that we propose is a straightforward database conversion model that translates existing software-dependent relational schemas into other software-dependent schemas. This is accomplished by developing a common, schematic representation that is a simple type of non-embedded data dictionary. We have chosen this approach because non-embedded data dictionaries have been shown to be successful in enhancing the integration of various applications (Appleton, 1987). The common schema developed in our work is independent of software specific data structures, integrity checks, and non-standard SQL.

In the following section, we present a Database Conversion Model that is used as an automated means of database conversion in our study. The model was developed as a means of providing a relatively transparent process that involves several phases for conversion of one system to another with minimal human intervention.

**Database Conversion Model**

Our study identified the need for a two-phase process for the effective management of database conversion from a source to a target system. The two phases are:

- **Phase 1**—Evaluation of the schema model for the source system database to be converted to facilitate translation into a Common Schema Representation allowable by the target system database.
- **Phase 2**—Movement of the existing data from the source to an target system, abiding by any prescribed mappings, such as type conversions, derived during the first phase.

The first phase contains the most difficult activities, which include: defining tables, views and constraints in the target system; mapping data types from the source to target system; defining a physical specification of segments and extents for the target system; and setting privileges or access rules. Once the first phase is successfully completed, the second phase becomes a straightforward process of moving data into a library of tables.

Figure 1 presents a high-level view of our database conversion model. The model shows the translation process from a source system into a Common Schema Representation which is then translated to a target system. The use of this conversion process would result in the mapping of any n database systems into 2*n mapping definitions. For example, Figure 1 shows four databases (db1—db4) and the required eight mapping definitions. Specifically, the conversion process is comprised of the following activities:

- **Map-to-Common** - a mapping is defined whereby a source database system is translated into a common schema using ANSI SQL.
- **Map-from-Common** - a translator process is used to access views created in the common schema representation in order to construct the target database. The translator process is also used to select table data from the source system for populating the new target database. Any data conversion necessary for successful translation is also performed.
- **SQL Translate Map** - supports a small subset of SQL used to retrieve the database information from the common schema representation and the actual data values from the source database system. Translator processes that support standardized SQL are common to a number of multidatabase systems (for examples, see Ahmed et al., 1991; Chung, 1990; Sheth et al., 1990; Thomas et al., 1990).

The next section describes the architecture of a database conversion system based on the conceptual model presented in Figure 1. The prototype SQL scripts used to instantiate the common schema have been omitted but can be found in their entirety in Leist and Becker (1994).

**Implementation of the Database Conversion Model**

The ISO/IEC 9075 standard (see ISO, 1992) serves as the basis for the conversion process in our model because it specifies a generic schemata describing the structure and semantics for given SQL applications. Such a generic schemata provides the means of representing and managing the data being translated from a source to a target database system.

Unfortunately, typical database systems are not in the model format as expressed in the ISO/IEC 9075 standard. What is needed is a definition of the tables and views in the common schema in accordance with the standard. These Common Schema tables and views would be stored in the data dictionary or catalogue for the relational database management system that is to be converted.

The Common Schema Representation in the conversion model shown in Figure 1 is defined at a more granular level, and expressed in a slightly different form in Figure 2 as being
Related Content

**Metric Databases**
[www.igi-global.com/chapter/metric-databases/11174?camid=4v1a](www.igi-global.com/chapter/metric-databases/11174?camid=4v1a)

**Using Business Rules within a Design Process of Active Databases**
[www.igi-global.com/article/using-business-rules-within-design/3251?camid=4v1a](www.igi-global.com/article/using-business-rules-within-design/3251?camid=4v1a)

**Categorizing Post-Deployment IT Changes: An Empirical Investigation**
[www.igi-global.com/article/categorizing-post-deployment-changes/3368?camid=4v1a](www.igi-global.com/article/categorizing-post-deployment-changes/3368?camid=4v1a)

**A Disk-Based Algorithm for Fast Outlier Detection in Large Datasets**
[www.igi-global.com/chapter/disk-based-algorithm-fast-outlier/24228?camid=4v1a](www.igi-global.com/chapter/disk-based-algorithm-fast-outlier/24228?camid=4v1a)