Preface

Over the last three decades, tremendous investment has been made in information systems. Such systems have evolved from file systems, through database systems, and we are now seeing the emergence of management information systems (MIS) and executive information systems (EIS). With the advent of each new form of technology, there has been a need to redesign and reimplement existing information systems.

In recent years, a great deal of resources has been put into the area of reengineering. Reengineering involves the redesign of existing information systems, while using as much of the existing systems as possible. That is, the approach taken is to transform the existing information systems into the format needed for the new technology, rather than to throw away the old systems. Such an approach has obvious benefits, particularly if it can be automated and/or supported by methods and tools.

Very often, a large company has multiple heterogeneous databases for MIS operations. The company needs to integrate them into a corporate database for its decision support systems. Subsequently, schemas integration must be performed to resolve the conflicts between two databases with respect to data name, data type, and data semantics. Schemas integration must be done before data integration, which is mainly concerned with the automation of loading data from source databases into an integrated database. Furthermore, in reality, user demands are changing daily. It is essential for companies to enhance and evolve the existing database schemas to meet the new data requirements.

Database normalization aims to remove irregularity and abnormality of update. The un-normal database is difficult to maintain the correctness of database after update. Relational data normalization is to eliminate data redundancy and update irregularity of relational schema. Irregularity of database design must be normalized in order to ensure user friendliness of updating data. Similarly, we need to normalize nonrelational database such as XML document to get rid of their repeating redundant data, and also to normalize their XML schema design to ensure good performance in accessing data.
This book will focus upon practical approaches to information systems reengineering, integration, and normalization including:

- The conversion of hierarchical or network database systems into relational database technology, or from a relational to an object-oriented database and XML database
- The schemas integration includes the integration of relational schemas, and the integration of XML schemas according to the relevance of the schemas
- The integration of multiple databases and also between a database system and an expert system to produce MIS (management information systems) and EIS (executive information systems)

The book will summarize the concepts, the approach to be taken, and the benefits to be gained in the above crucial technological areas. It will focus upon proven methods and tools for:

- Converting hierarchical and network databases to relational technology, or from relational to object-oriented databases, or from relational to XML databases
- Reengineering existing systems to produce MIS and EIS
- Normalizing relational database and XML schema design

The book will describe in detail:

- Database conversion techniques
- Reverse engineering and forward engineering for data modeling
- A reengineering methodology for information systems
- Techniques of schemas and data integration
- Normalize database to eliminate redundant data

From a professional point of view, this book proposes a general solution for the problem of system migration to new database technology. It offers a systematic software engineering approach for reusing existing database systems built with “old” technology by converting them into the “new” database technology. As a result, investment in the existing information systems can be protected by upgrading database systems and expert systems, rather than phasing them out.

This book focuses on methodologies for information systems reengineering, integration, and normalization for the interoperability of their database systems. It applies many examples, illustrations, and case studies of procedures for reusing existing database systems and information systems. The objective is to make the methodologies very practical for readers to follow. Even though there are many technical terminologies used in the book, the techniques proposed are simple enough for students or computer professionals to follow. The content of the book is divided into ten chapters.

Chapter 1 gives an overview of information systems, and deals with its past history, its evolution to management information systems, its problems encountered in file systems, its solution found in database systems and expert systems, and the need for the reengineering of existing database systems and information systems. It also describes database conversion, the merge of multiple databases, and the in-
Integration of the expert systems and the database systems into an expert database system. It shows how to apply data transformation for electronic data interchange on the Internet.

Chapter 2 describes basic theories and data structures of various data models, including hierarchical, network, relational, object-oriented, and XML. Their pros and cons are discussed. Expert systems technology is explained. The advanced expert database systems are introduced. The basic concepts discussed include data definition language, data manipulation language, forward chaining, backward chaining, procedural language and nonprocedural language, data type definition, and XML schema definition.

Chapter 3 covers various techniques in schema translation from nonrelational to relational, and from relational to object-oriented and XML databases. Reverse engineering is adopted to recover original schema’s semantics into the conceptual model of the Extended Entity Relationship (EER) model. Forward engineering is used to map the EER model into relational or Unified Model Language (UML), a conceptual model for an object-oriented database.

Chapter 4 shows a methodology of converting data from nonrelational database to relational database, and from relational database to object-oriented database, and also from relational database into XML database. Download and upload processing in a logical level approach is adopted to do the task.

Chapter 5 explains a methodology of emulating SQL by using a hierarchical or network database data manipulation language. The methodology can be used in program translation from relational database programs to nonrelational database programs. The objective is to provide a relational interface to the nonrelational database so that the users can use SQL to access a hierarchical or network database. It also presents a methodology of translating SQL query into OSQ (Object SQL or Object Query Language) and XQL (XML Query Language).

Chapter 6 summarizes the database conversion methodology for converting hierarchical or network databases to relational databases. The methodology is in three phases: schema translation, transaction translation, and data conversion. The first and second phases provide a relational interface to a nonrelational database as a temporary solution in the database conversion (migration) process. The third phase provides a permanent solution to convert data from nonrelational database to relational database after nonrelational database programs are phased out or rewritten. A case study of constructing an XML view of a relational database involving schema and data transformation from relational into XML is presented. A section covers the interoperability of relational databases and XML databases such that their database languages SQL and XQL can access both databases through database gateways.

Chapter 7 offers a state-of-the-art methodology for integrating two relational or XML database schemas by resolving their name, data type, and data semantics conflicts with user supervision. The relational, object-relational or XML data integration can only be done after relational, object-relational or XML schemas integration for the loading of data into the integrated databases is performed. The rational of schemas integration is based on the application relevance of the schemas. A Frame
model metadata is introduced to store data operation for encapsulation in the object-oriented database.

Chapter 8 lays out the rules in integrating expert systems and database systems for the purpose of reengineering. The technique is to transform both expert systems rules and database systems relations into a common Frame model metadata. This Frame model metadata offers object-oriented-like database functions by treating each frame as an object and a collection of objects as a class. Coupling classes, active classes, static classes, and integrated classes are introduced to implement an expert database system (EDS). The users can then apply EDS to develop new applications.

Chapter 9 shows how to normalize un-normal form of relational schema into first, second, third, Boyce Codd, fourth, and fifth normal form. On the other hand, the denormalization is the inverse of normalization. It transforms the normalized database design into un-normal form for better performance. As XML document has become a data standard on the Internet, XSD Graph is a conceptual schema to represent the data semantics of XML document. We need to detect and normalize XML schema and document by factorizing their redundant elements into a single unique element in XML Schema Definition.

Chapter 10 summaries the methodologies proposed by the book. The main theme is that knowledge engineering is a requirement for information systems re-engineering, integration, and normalization. We need users’ knowledge to assist system developers in reusing existing database systems and expert systems in order to develop new applications. The final result is database systems upgrade, multiple databases integration and expert systems enhancement to knowledge-based systems. As knowledge engineering becomes important in data processing, the resultant knowledge-based system, that is, the expert database system, will become a very important asset to companies. The resultant information system must be normalized to eliminate data redundancy.