The entity relationship (ER) data model has become the most popular conceptual data modeling tool. Most CASE and expert tools that perform data modeling are based on the ER model. Further, many ER-based methodologies have been proposed. No ER-based methodology, however, has presented a systematic approach to model user views. This paper suggests an ER-based methodology for data modeling of user views even if they are complex and involve many layers of nested data. The methodology can not only handle binary relationships, but also ternary and higher degree relationships. In addition, it includes a methodical approach to detect derived relationships. It should be possible to implement the methodology in a CASE or expert tool.

In recent years, the entity relationship (ER) data model (Chen, 1976) has gained prominence as a conceptual data modeling tool (Batini, Ceri, and Navathe, 1992). Methodologies like logical relational design methodology (Teorey, Yang, and Fry, 1986), CASE tools like Excelerator (Index Technology, 1987), and expert tools like View Creation System (Storey and Goldstein, 1988) are all based on the entity relationship (ER) data model. Usability studies (e.g., Batra, Hoffer, and Bostrom, 1990; Jarvenpaa and Machesky, 1989; Juhn and Naumann, 1985) have found that the ER model (compared to other alternatives) leads to higher user performance in data modeling.

It is well known that methodologies based on the entity relationship (ER) model are well suited when the input is a high level text based description of the application (Batra, Hoffer and Bostrom, 1990). A systematic methodology, however, does not exist for applications that have input in the form of user views, which are usually forms or reports required in a business application. User views are easily available and are reliable sources of known user requirements. The ER data model should be effective in modeling user views since it allows abstraction of data elements into entities: attributes which pertain to the same object can be clustered into an entity. Thus, instead of having to deal with numerous attributes and their interrelationships especially in a complex user view, a designer can reduce the complexity of the task by first clustering the attributes into entities, and then considering the relationships between the entities.

The ER data model, thus, seems quite suitable for modeling user views. Yet, text books and research papers have generally refrained from recommending an ER-based approach when the input is user views. Recently, however, some texts (e.g., Batini, Ceri and Navathe, 1992; McFadden and Hoffer, 1990) have adopted the ER data model for applications involving user views as input. But these texts have not presented a systematic methodology for using the ER model. The texts indicate broad guidelines for identifying entities and relationships in a user view, and integrating the ER diagrams resulting from different user views. Consequently, there is a need for a precise ER-based methodology that goes beyond broad procedural steps. The methodology should detail a specific approach for representing requirements expressed in a user view into a conceptual data model. In addition, it should be possible to incorporate the methodology in a CASE (Computer-Aided Software Engineering) tool.
Determining the entities and relationships in an application is only one aspect of the overall conceptual design. An important aspect of conceptual database design is that it should be minimal, that is, there should not be any redundancy in the design (Batini, Ceri, and Navathe, 1992). Since different user views can map to overlapping entities and relationships, the methodology should address the minimal aspect of conceptual database design. Although one can integrate the ER diagrams by a union of entities and relationships (after reconciling homonyms and synonyms), this does not completely eliminate the minimalism problem. Some relationships may be completely derived from other relationships, and in most cases, should be eliminated from conceptual design. A relationship is derived if it can be inferred by other relationships. It may be difficult, at times, to detect such relationships. It would be useful, therefore, to have a methodology to guide the designer discover derived relationships, which in most cases need to be deleted. Deleting a derived relationship does not lead to loss of semantics of an application.

This paper presents an ER-based methodology for modeling user views and detecting derived relationships. In the next section, a brief description of relevant literature is presented. This is followed by a listing of the phases involved in designing a conceptual data model. We then present the general basis of the methodology pertaining to modeling of user views. The next two sections describe the design and rules used by the methodology to use the ER model to model user views and detect derived relationships. The paper concludes with a discussion on how a CASE tool can incorporate such a methodology.

Literature Survey

Conceptual and logical database design are important but difficult tasks, especially for large databases. Conceptual design cannot totally be performed by expert tools, and the designer has full responsibility for the process of understanding and transforming requirements into conceptual schemas (Batini, Ceri, and Navathe, 1992). To accomplish this, the designer needs a methodology that will facilitate the design process. A CASE or an expert tool can incorporate such a methodology and guide the designer as well as check for inconsistencies.

Many ER-based methodologies have been proposed to develop a conceptual database. Teorey, Yang and Fry (1986) proposed LRDM (Logical Relational Database Methodology), a methodology for relational databases using the extended entity relationship (EER) model, which extends the ER to include the generalization concept. The LRDM methodology involves developing an EER diagram which can then be translated into a relational representation. The translation can be performed by a designer or a CASE tool. Blaha, Premerlani, and Rumbaugh (1988) suggested an object-oriented model for relational database design which was similar in its approach to LRDM except for its additional emphasis on methods, that is, programs that capture the behavior of objects. Nilson (1985) illustrates a methodology for translating conventional files into the ER model. Batini, Demo, and DiLeva (1984) describe a methodology that assumes forms as input documents. Batini, Ceri and Navathe (1992) have introduced four strategies - top-down, bottom-up, mixed and inside-out - using the ER model. However, these methodologies do not provide detailed procedural steps in using these strategies to develop a conceptual data model from user views.

Holsapple, Shen and Whinston (1982) proposed a methodology for automated database design based on user views. However, their methodology did not consider relationships more complex than binary, and they did not suggest any technique to detect derived relationships.

In summary, past literature has not addressed two issues. First, except for the Holsapple, Shen and Whinston (1982) paper, there are no specific rules to model and integrate user views. General guidelines are of little use when user views become complex. User views become more complex as the number of nests increase. A nest is ‘set’ data, that is, data items which are multivalued with respect to other data items. For example, if there is a report that shows a list of orders along with invoices corresponding to each order, the information pertaining to invoices will be ‘set’ with respect to each order. Second, there is no methodology that helps a user eliminate derived relationships. Consider the following example: a customer can place many orders and an order can have many invoices; thus, the relationship between customer and invoice can be derived from the other two relationships. If the relationship between customer and invoice is explicitly shown, it will lead to redundancy. While the example shown is simple, in a large database, the detection of such relationships could become tricky since it would be difficult to get an overall picture of the database.

This paper presents an ER-based methodology to address these issues, namely, modeling user views, and detecting derived relationships. Before presenting the methodology, however, the general approach used in designing a conceptual database from user views is described.

Conceptual Modeling Approach

A user view contains data items which may belong to different entities and relationships. Creating an ER diagram for a user view requires identifying the relevant entities and relationships. Further, different user views may embed same entities and relationships. Duplicate entities and relationships need to be eliminated. Thus, the ER diagrams obtained from the various user views need to be integrated into an overall ER diagram which represents the database.

An ER-based approach for conceptual design of user views can, thus, be considered as comprised of the following two phases:
Related Content

Managing Uncertainties in Image Databases
[www.igi-global.com/chapter/managing-uncertainties-image-databases/8036?camid=4v1a](www.igi-global.com/chapter/managing-uncertainties-image-databases/8036?camid=4v1a)

Representation and Storage of Motion Data
[www.igi-global.com/article/representation-storage-motion-data/3283?camid=4v1a](www.igi-global.com/article/representation-storage-motion-data/3283?camid=4v1a)

Component-Based Generalized Database Index Model
[www.igi-global.com/chapter/component-based-generalized-database-index/11127?camid=4v1a](www.igi-global.com/chapter/component-based-generalized-database-index/11127?camid=4v1a)

Fuzzy Database Approaches
[www.igi-global.com/chapter/fuzzy-database-approaches/18759?camid=4v1a](www.igi-global.com/chapter/fuzzy-database-approaches/18759?camid=4v1a)