LNCS 1727, pp110-121, 1999.
(URL: http://www.springer.de/comp/lncs/index.html)

On the Ontological Expressiveness of Temporal Extensions to the Entity-Relationship Model*

Heidi Gregersen and Christian S. Jensen

Department of Computer Science, Aalborg University, Denmark

Abstract. It is widely recognized that temporal aspects of database schemas are prevalent, but also difficult to capture using the ER model. The database research community's response has been to develop temporally enhanced ER models. However, these models have not been subjected to systematic evaluation. In contrast, the evaluation of modeling methodologies for information systems development is a very active area of research in the information systems engineering community. Based on a framework from information systems engineering, this paper evaluates the ontological expressiveness of three different temporal enhancements to the ER model. The three temporal ER model extensions are well-documented, and together the models represent a substantial range of the design space for temporal ER extensions.

1 Introduction

Both the research community and the companies that design databases have recognized that temporal aspects of database schemas are both prominent and difficult to capture using the ER model. Intuitive and easy-to-comprehend diagrams become obscure and cluttered when modeling fully the temporal aspects. As a result, the research community has developed a number of temporally enhanced ER models [13, 5, 18, 4, 2, 23, 22, 16, 28, 8].

Both the standard and temporally enhanced ER models may be used for different, but related purposes, namely for analysis—i.e., for modeling a part of reality—and for design—i.e., for describing the database schema of a computer system. The typical use seems to be one where the model is used primarily for design and where the constructed diagrams are mapped to a relational platform. In step with the increasing diffusion and use of relational platforms in industry, ER modeling is growing in popularity.

In the database research community, the models that are offered for conceptual database design are rarely evaluated systematically. In contrast, in the area of information systems engineering, the evaluation of modeling methodologies for information systems development is a very active area of research. A substantial number of evaluations are reported in the literature [1, 14, 20, 6, 11, 19, 24–27, 15], and the IFIP Working Group 8.1 is co-sponsoring an annual workshop, EMMSAD, devoted solely to this topic.

* This work was supported in part by grants 9502695 and 9700780 from the Danish Technical Research Council and grant 9701406 from the Danish Natural Science Research Council, as well as grants from the Nykredit corporation and the Danish National Centre for IT Research.

Weber and Wand have developed a framework for evaluating the ontological expressiveness of information systems development methodologies [24–27]. This framework includes an ontological model of the real world that covers both structural and behavioral aspects. The framework has been used to evaluate the notations and associated semantics of three models for information systems development.

The present paper uses the approach of Weber and Wand for evaluating three different temporal extensions to the ER model. Specifically, the objective of this paper is to evaluate the ontological expressiveness of the temporal notational constructs of three selected temporal ER models [23, 28, 8]. Each of these temporal ER model extensions is well-documented, and together the models represent a substantial range of the designs space for temporal extensions. The evaluation considers the use of the models for design only, since this is the typical use of the model. As a result, it is evaluated how well the models capture the temporal aspects of a database design. This necessitates the introduction of a representational model for a database design. The three models were chosen based on their recency and quality. One was published in 1991, and the latter two were published within the last two years and may be considered second-generation models in that they attempt to build on the earlier models.

This work is related to four surveys and comparisons of methodologies for the analysis and design of information systems. Brandt [1] surveys and evaluates thirteen methodologies for system's specification. Kung [14] studies three conceptual models with a time perspective. Floyd has [6] evaluated and compared three different system's development methodologies. Jayaratna [12] has developed a framework, termed NIMSAD, for understanding and evaluating methodologies. The evaluations above focus on modeling properties, the usage of the models, and the user-friendliness of the models. Only one evaluation considers expressiveness as a criterion [14], but does not consider the models' abilities to express temporal aspects; in contrast, the evaluation in this paper focuses entirely on expressiveness in relation to temporal aspects.

Conceptual models for database design have also been evaluated and compared. Schrefl et al. [20] develop a set of criteria for comparing conceptual models and evaluate seven conceptual models. Hull and King [11] discuss issues of conceptual modeling and survey sixteen conceptual models. Peckham and Maryanski [19] describe generic properties of conceptual models and survey a representative selection of models. Leander et al. [15] compare the modeling capabilities of the ER model and the NIAM methodology. The evaluations of the conceptual models for database design all focus on non-temporal properties. This paper's evaluation focuses exclusively on the modeling of temporal aspects.

Ten temporally extended ER models have been surveyed and evaluated by the authors [7, 10]. The focus of these evaluations are entirely on model properties, and criteria based on ontologies are not considered.

In summary, the focus of previous, related evaluations ranges from determining the environments in which methodologies where developed, over the usages of the methodologies, to the user-friendliness of the methodologies. Some studies examine the expressiveness of conceptual data models [14, 15]. However, previous work does not consider evaluation parameters that concern how well the models capture temporal aspects, which is the topic of this paper. We evaluate three different temporally extended

ER models' abilities in describing the temporal aspects of relational database schemas capturing temporal aspects. A substantially extended version of this paper considers also the use of the three models for analysis [9].

The paper is structured as follows. Section 2 states the objectives of the paper's evaluations, and Section 3 presents the evaluation framework. In Section 4, the three temporal ER extensions are evaluated. Finally, Section 5 summarizes the findings and outlines directions for future research.

2 Evaluation Objectives

This section first describes the context of the paper's evaluation, by outlining several aspects of a conceptual model that may be subjected to evaluation. It then proceeds to state the objective of the paper's evaluation, by positioning it within this context.

2.1 Types of Evaluation

Three different approaches to evaluating a conceptual model can be taken. First, the evaluation can be done by examining the diagrams that result from using the conceptual model, i.e., the *diagrams* will be the target of the evaluation. Second, the *notation*, i.e., the "building blocks" of the conceptual model can be evaluated by examining, e.g., which notational constructs the model offers for modeling specific aspects of the underlying implementation. Third, the *methods and guidelines* that describe how to use the model during the design of a database can be evaluated.

The difference between evaluating the resulting diagrams versus the notation of a temporal ER model may be explained as follows. A temporal ER model is a graphical model, that is, the notation of the model is graphical symbols, including rectangles, diamond, and lines. Each symbol has a specific interpretation (semantics) that gives the meaning of the symbol. In contrast, a diagram is a connected collection of symbols. Evaluating a conceptual model by considering specific diagrams produced using the model versus evaluating it by considering each of its modeling constructs yields different evaluations. We proceed to discuss the three types of evaluations in more detail.

One can evaluate a diagram with respect to analysis by observing how well the diagram describes the underlying database structure. This means that it should be possible for database designers to recognize the structure of the underlying database by examining the diagram. The notation of the model is irrelevant to the evaluation of a diagram, since the focus is entirely on how easy it is to recognize the database structure by examining the diagram.

Next, the notation of a conceptual model can be evaluated with respect to design by by examining whether or not the modeling constructs offered by the model can describe database structure with the desired accuracy. In order to do this, we have to examine which modeling constructs the underlying data model offers and and then to determine which modeling constructs the conceptual model offers for describing these constructs of the underlying model.

Finally, the methods and guidelines for how to use the model in the design phase can be evaluated with respect to how well they help and support the construction of the schema of the underlying database.

2.2 Choice of Evaluation Objectives

We have chosen not to base our evaluation on specific temporal ER diagrams, for several reasons. First, it is not clear who should create the diagrams. It is almost impossible to ensure that the corresponding diagrams for several different temporal ER models are created under similar conditions. Solving this problem by having the same persons create all diagrams introduces new problems: the sequence in which the diagrams are created is likely to matter, so that later diagrams are influenced by knowledge obtained during the creation of earlier diagrams. Second, the evaluation of how well a ER diagram documents the underlying database is quite subjective. Different database designers may have different answers to whether or not a digram documents the underlying database. We have also chosen not to evaluate the methods and guidelines that the designers of the temporal ER models may have given, for the simple reason that no temporal ER model is equipped with such methods and guidelines.

Rather, we have chosen to evaluate the notations of three temporal ER models. The evaluation occurs within a framework originally developed for evaluating the notations of methodologies for information systems development. Therefore, a new ontology to be used for evaluating the temporal ER model is provided.

3 An Ontologically-Based Evaluation Framework

This section presents the overall evaluation framework and then the ontology to be used when evaluating the temporal ER models.

3.1 Overall Framework

One part of Wand and Weber's framework for evaluating the ontological expressiveness of the notations of methodologies for analysis and design of information systems (ISAD) [24–27] is a representational model of the real world. This model consists of all the real-world constructs, called ontological constructs, that an ISAD notation must be able to capture in order to model the real world. Since our evaluation focuses on temporally extended ER models' capabilities in modeling temporal aspects of a database design, we develop our own representational model.

The evaluation of whether or not a notation is ontologically expressive is based on the notion of mathematical mappings. The focus is on two sets: the set of ontological constructs and the set of notational constructs offered by the model to be evaluated. Two mappings exist between these, the *representation mapping*, which maps ontological constructs to corresponding notational constructs, and the *interpretation mapping*, which maps notational constructs to corresponding ontological constructs, see Fig. 1.

Although based on the precise mathematical notion of mappings between sets, the evaluation involves a certain degree of subjectiveness because the construction of the specific mappings necessitates some interpretation on the part of the evaluator.

Informally, the notation of a model is *ontologically complete* if the notation can represent the same information as the representational model (the ontological constructs); otherwise, it is ontologically incomplete and suffers from construct deficit [26]. That is,

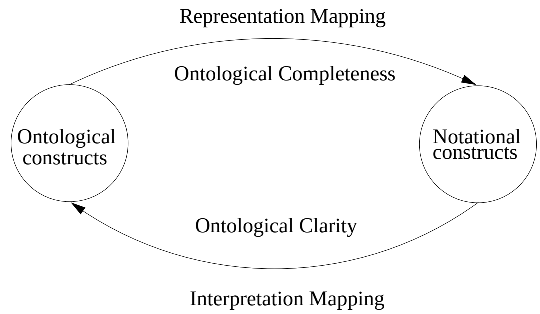

Fig. 1. Sets and Mappings in the Evaluation Framework [26]

a notation is ontologically complete if the representational mapping from the ontological constructs to the notational constructs is total.

Next, *ontological clarity* concerns the interpretation mapping from the notational constructs to the ontological constructs. Three situations can obscure the clarity of a notation. First, if one notational construct can be mapped into more than one ontological construct, this implies construct overload. Second, if more than one notational construct can be used to model the same ontological construct, the notation suffers from construct redundancy. Third, if there are notational constructs that do not represent any ontological constructs, the notation suffers from construct excess.

3.2 Ontology

When a conceptual model is used for database design, the predominant target model is the relational model, and we assume that the target model is this model or a temporal extension of it.

A database stores information about objects from a modeled reality, which consists of objects. The time during which an object exists in the reality, we call the *existence time* of the object. An object is characterized by its properties. The time during which it is true (in the modeled reality) that a property has a specific value is called the *valid time* of that particular value.

A relation may capture the existence or valid time of its tuples by using *time attributes* defined over an appropriate *time domain*. The time during which a tuple is current in the database is called the *transaction time* of the tuple, and a relation may also include time attributes that capture this aspect. Next, *user-defined constraints* can be a defined over the relations, e.g., how many tuples in one relation are allowed to have references to the same tuple in another relation. The constraints that must hold for all points in time, we will call *temporal*, while we will term the constraints that must hold at each point in time in isolation *snapshot* constraints.

4 Model Evaluation

This section examines the ontological completeness and clarity of three temporally extended ER models. The notations of the three models are presented by diagrams modeling a company database.

4.1 The ERT Model

The first model to be examined is the Entity-Relation-Time (ERT) model [23].

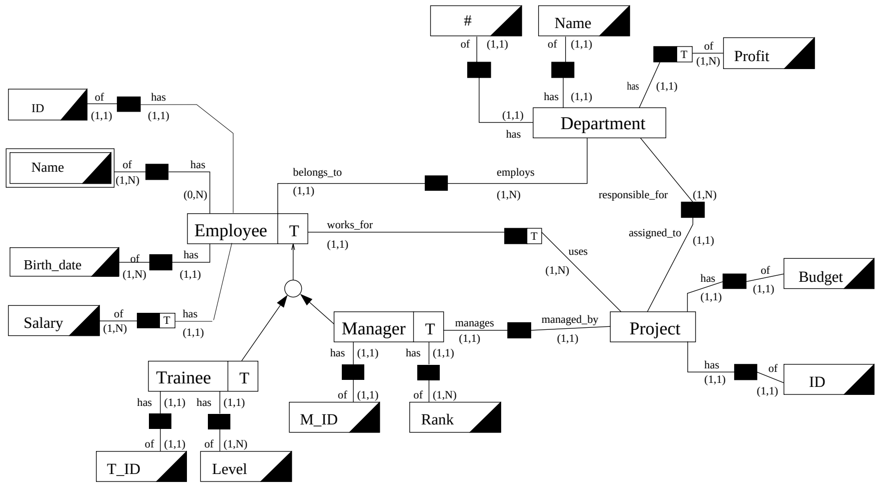

Fig. 2. An ERT Diagram Describing a Company Database

The model supports lifespans of entities and the valid time of relationships. Fig. 2 is an ERT diagram modeling a company database. A rectangle represents an entity class or a value class (black triangle placed in the bottom-right corner). Entity and value classes can be specified as complex by using a double rectangle. An entity class expanded with a “timebox” containing the symbol T is specified as temporal. Relationship classes are denoted by small filled rectangles, and only binary relationships are available. These can be specified as temporal by expanding the filled rectangle with a timebox. For each entity (or value) class participating in a relationship class, an involvement role and a cardinality constraint must be specified. An ISA relationship class is denoted by a circle with arrow(s) flowing from the subclass(es) to the circle and an arrow flowing from the circle to the superclass.

The result of the examination of ERT with respect to ontological completeness is presented in Table 1. It can be seen that ERT suffers from three, possibly four, cases of construct deficit. First, the model has no notation for specifying time domains. Second, the model does not support transaction time. Third, no notation is offered for specifying temporal constraints. Fourth, since the semantics of the construct offered for specifying cardinality constraints is unclear, it cannot be determined with certainty if the cardinality constraints is a snapshot constraint.

The result of the examination of ERT with respect to ontological clarity is presented in Table 2. The model does not suffer from construct redundancy. It is unclear if ERT suffers from construct excess due to the unclear semantics of the constructs offered for specifying constraints. The model suffers from one cases of construct overload: the timebox models both lifespan for entity types and valid time for attributes.

Table 1. Evaluating ERT With Respect to Ontological Completeness

Ontological Construct	ERT Representation
Time domains	Not represented.
Lifespan attributes	Represented by the timebox extension of entity classes.
Valid time attributes	Represented by the timebox extension of user-defined relationship classes.
Transaction-time attributes	Not represented.
Temporal user-defined constraints	Not represented.
Snapshot user-defined constraints	Represented by the placing the min-max constraint near the entity type (or value class) participating in the relationship class.

Table 2. Examining ERT With Respect to Ontological Clarity

ERT Construct	Ontological Construct
Timebox	Models lifespan attributes and valid-time attributes.
Cardinality constraint	Might model a snapshot user-defined cardinality constraint.
Superclass/subclass completeness constraint	Might model a snapshot user-defined generalization completeness constraint.
Superclass/subclass disjointness constraint	Might model a snapshot user-defined constraint.

4.2 The TERC+ Model

This section evaluates the TERC+ model [28], which supports lifespans of entity types and valid time of relationship and attribute types. Fig. 3 presents a TERC+ diagram modeling a company database.

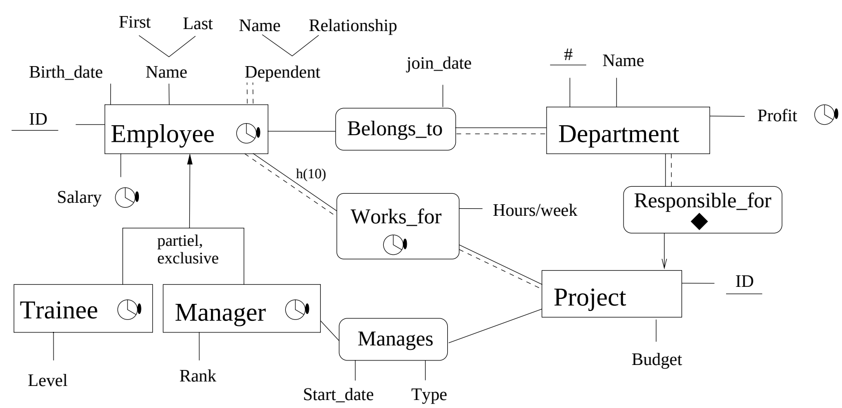

Fig. 3. A TERC+ Diagram Describing a Company Database

An entity type is represented by a rectangle, and a relationship type is denoted by a rectangle with rounded corners. Entity types and relationship types can be annotated with a clock symbol to indicate that their life cycles are to be captured. Attributes are represented by plain text linked to entity or relationship types and can be annotated with a clock symbol to indicate that valid time is to be captured. Cardinality constraints are expressed using the lines connecting the entity types to the relationship types. A historical cardinality constraint, h(*max*), can be expressed for relationship types. A part_of/component_of relationship type is denoted as a relationship type annotated with a filled diamond and an arrow pointing to the component.

Table 3 characterizes the ontological completeness of TERC+. The model suffers from three cases of construct deficit. First, there is no notation available for specifying time domains. Second, there is no notation offered for specifying the transaction time of attributes. Third, temporal user-defined constraints cannot be specified.

Table 3. Evaluating TERC+ With Respect to Ontological Completeness

Ontological Construct	TERC+ Representation
Time domains	Not represented.
Lifespan attributes	Represented by annotated entity types with a clock symbol.
Valid-time attributes	Represented by attributes and relationship types annotated with a clock symbol.
Transaction-time attributes	Not represented.
Temporal user-defined constraints	Not represented.
Snapshot user-defined constraints	Represented by constraint (min, max) that has to be specified for each entity type participating in a relationship type.

The result of examining the ontological clarity of TERC+ is presented in Table 4. The model does not suffer from construct redundancy, but from construct overload, since the clock symbol models both valid-time and lifespan attributes. Construct excess occurs because the historical cardinality constraint cannot be mapped to any ontological construct.

Table 4. Examining TERC+ With Respect to Ontological Clarity

TERC+ Construct	Ontological Construct
Clock symbol	Model valid-time or lifespan time attributes.
Cardinality constraint	Models a snapshot user-defined cardinality constraint.
Historical cardinality constraint	No corresponding ontological constructs.
Total generalization constraint	Models a snapshot user-defined generalization completeness constraints.
Exclusive generalization constraint	Models a snapshot user-defined generalization disjointness constraints.

4.3 The TIMEER Model

The last model to be evaluated is the Time Extended ER (TIMEER) model [8]. This model offers support for lifespans of entity types; valid time for attributes and relationship types; and transaction time for entity types, relationship types, and attributes. Figure 4 presents a TIMEER diagram modeling a company database. The model extends the notation of the ER model with annotations to indicate which temporal aspects are to be captured. The annotations are LS, indicating lifespan support, VT indicating valid-time support, TT indicating transaction-time support, LT indicating lifespan and transaction-time support, and BT indicates valid- and transaction-time support.

Table 5 shows that the TIMEER model is ontologically complete with respect to the relational ontology.

Considering the ontological clarity of the TIMEER model with respect to the relational ontology, it follows from Table 6 that the model does not suffer construct

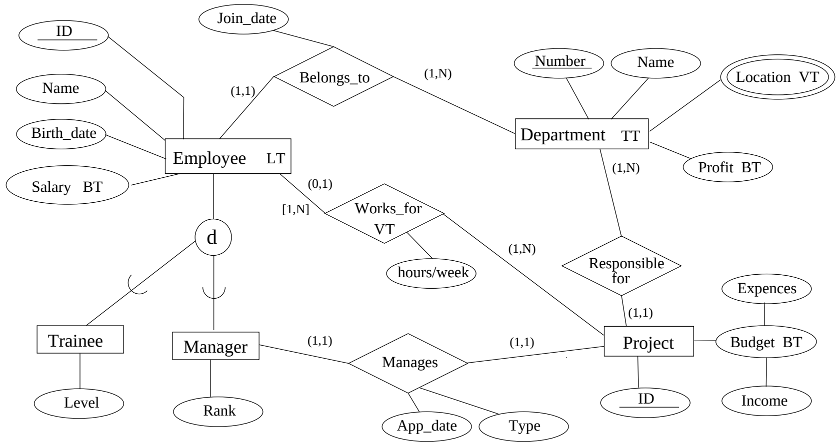

Fig. 4. A TIMEER Diagram Describing a Company Database

Table 5. Evaluating TIMEER with Respect to Ontological Completeness

Ontological Construct	TIMEER Representation
Time domains	The domain of a time attribute is indicated by the annotation used. If the annotation is an LS then the time domain is the lifespan time domain.
Lifespan attributes	Entity types annotated with an LS (LT) model the presence of lifespan attributes.
Valid-time attributes	The annotations VT and BT indicate the presence of valid-time attributes.
Transaction-time attributes	The annotations TT, LT, and BT indicate the presence of valid-time attributes.
Temporal user-defined constraints	These are represented by the lifespan participation constraint [min, max] for relationship types.
Snapshot user-defined constraints	These are represented by the snapshot participation constraint (min, max) for relationship types.
	Represented by the superclass/subclass completeness constraint.
	Represented by the superclass/subclass disjointness constraint.

redundancy. However, it suffers from one case of construct overload since the superclass/subclass completeness constraint models both temporal and snapshot user-defined generalization completeness constraints. The model does not suffer from construct excess.

5 Summary and Research Directions

At the outset, the paper presents an overall framework for examining the ontological expressiveness and clarity of temporal ER models. The framework includes an ontology which is used as a so-called representational model. This ontology describes the constructs of a relational database schema, which serve a role in capturing the temporal aspects of data.

Table 6. Examining the TIMEER With Respect to Ontological Clarity

TIMEER Construct	Ontological Construct
LS	Models lifespan attributes.
VT	Models valid-time attributes.
TT	Models transaction-time attributes.
LT	Models lifespan and transaction-time attributes.
BT	Models valid-time and transaction-time attributes.
Snapshot participation constraints	Model snapshot user-defined constraints.
Lifespan participation constraints	Model temporal user-defined constraints.
Superclass/subclass completeness constraints.	Model temporal and snapshot user-defined generalization completeness constraints.
Superclass/subclass disjointness constraints.	Model snapshot user-defined generalization disjointness constraints.

The framework concerns the mappings between the ontological constructs and those of temporal ER models. The framework is used for evaluating three temporal extensions of the ER model, namely the ERT, the TERC+, and the TIMEER model, with respect to their use for design of relational databases managing time-varying data.

All three models offer support for capturing lifespan and valid-time attributes and snapshot constraints (although the semantics of ERT's constraints are unclear). Only TIMEER is able to model transaction-time attributes and temporal constraints. The overall result is that no model is ontologically expressive with respect to the ontology. In addition, the ERT and TERC+ models are ontologically incomplete and ontologically unclear. The TIMEER model is ontologically complete, but also ontological unclear. The model supports all the three temporal aspects considered. As a result, TIMEER makes it possible to model the temporal aspects, as covered by the ontology, of a relational database schema. With the ERT and TERC+ models, it is not possible to model all the temporal aspects a relational database schema. This leads to the conclusion that these two models do not fully support the conceptual design of databases managing time-varying data, since transaction-time support is frequently necessary in applications.

The research reported in this paper points to several directions for future research that deserve further attention.

It is recommended that the ERT and TERC+ models be enhanced with support for transaction time. Next, it appears relevant to consider extending the ERT and TIMEER models with support for modeling dynamic aspects of reality; TERC+ already offers some such support. In addition, new notation supporting the capture of database behavior might be introduced. Indeed, the idea of being able to capture in the conceptual model the evolution of a database schema appears to be very appealing. That is, it should be studied how to extend these and other conceptual models with notational constructs that conveniently capture the evolution of the database schema over time.

As another topic, the evaluation framework itself might be enhanced. The outcome of an evaluation is quite sensitive to the ontology employed, making it an interesting direction to expand the ontology to capture better the structural aspects not related to time, and perhaps also to capture dynamic aspects. It might also be of interest to develop a consensus ontology or to have an ontology be developed independently, by users of conceptual models.

Also, the mappings between notational constructs and ontological constructs could be generalized to include a degree of satisfaction rather than always assuming 100% satisfaction. The historical participation constraint of TERC+—which limits the maximum number of relations, but not the minimum number—offers an example of a partial satisfaction of a temporal constraint. Evaluations based on these more sophisticated mappings might yield a richer picture of the models. On the other hand, the assignment of degrees may prove very subjective.

The type of evaluation conducted indicates that the models evaluated are very similar, in that if only a few constructs are added to each of the three models, they would be isomorphic to each other. We feel that this apparent similarity is to some extent an artifact of the evaluation framework, which is unable to, e.g., discern significant linguistic differences. To obtain a more complete understanding of the models and their similarities and differences, more evaluations based on other criteria are recommended.

More generally, it is felt that there is a need for more methods that systematically evaluate and compare extended ER models. Such methods are likely to prove useful to both the designers of new conceptual models and the users of such models.

Acknowledgments

The authors would like to thank the anonymous reviewers for their particularly insightful comments.

References

1. I. Brandt. A Comparative Study of Information Systems Design Methodologies. In T. W. Olle, H. G. Sol, and C. J. Tully, editors, *Information Systems Design Methodologies: A Feature Analysis*, pp. 9–36. Elsevier Science Publishers, 1983.
2. R. Elmasri, I. El-Assal, and V. Kouramajian. Semantics of Temporal Data in an Extended ER Model. In *9th International Conference on the Entity-Relationship Approach*, pp. 239–254, 1990.
3. R. Elmasri and S. B. Navathe. *Fundamentals of Database Systems*. Benjamin/Cummings, 2. edition, 1994.
4. R. Elmasri, G. Wuu, and V. Kouramajian. A Temporal Model and Query Language for EER Databases. In A. Tansel et al., editor, *Temporal Databases: Theory, Design, and Implementation*, Chapter 9, pp. 212–229. Benjamin/Cummings Publishers, Database Systems and Applications Series, 1993.
5. S. Ferg. Modeling the Time Dimension in an Entity-Relationship Diagram. In *4th International Conference on the Entity-Relationship Approach*, pp. 280–286, 1985.
6. C. Floyd. A Comparative Evaluation of Systems Development Methods. In T. W. Olle, H. G. Sol, and A. A. Verrijn-Stuart, editors, *Information Systems Design Methodologies: Improving the Practice*, pp. 19–54. Elsevier Science Publishers, 1986.
7. H. Gregersen and C. S. Jensen. Temporal Entity-Relationship Models—a Survey. *IEEE Transactions on Knowledge an Data Engineering*, 11(3):464–497, 1999.
8. H. Gregersen and C. S. Jensen. Conceptual Modeling of Time-Varying Information. Technical Report TR-35, TimeCenter, 1998.

9. H. Gregersen. Temporally Enhanced Database Design. Ph.D. Thesis, Department of Computer Science, Aalborg University, 1999.
10. H. Gregersen, C. S. Jensen, and L. Mark. Evaluating Temporally Extended ER Models. In K. Siau, Y. Wand, and J. Parsons, editors, *Proceedings of the Second CAiSE/IFIP8.1 International Workshop on Evaluation of Modeling Methods in Systems Analysis and Design*, 12 pages, 1997.
11. R. Hull and R. King. Semantic Database Modeling: Survey, Applications, and Research Issues. *ACM Computing Surveys*, 19(3):201–260, 1987.
12. N. Jayaratna. *Understanding and Evaluating Methodologies: NIMSAD – A Systematic Framework*. Information Systems, Management and Strategy Series. McGraw-Hill, 1994.
13. M. R. Klopprogge and P. C. Lockeman. Modeling Information Preserving Databases: Consequences of the Concept of Time. In *9th International Conference on Very Large Data Bases*, pp. 399–416, 1983.
14. C. H. Kung. An Analysis of Three Conceptual Models with Time Perspective. In T. W. Olle, H. G. Sol, and C. J. Tully, editors, *Information Systems Design Methodologies: A Feature Analysis*, pp. 141–167. Elsevier Science Publishers, 1983.
15. A. H. F. Laender and D. J. Flynn. A Semantic Comparison of the Modeling Capabilities of the ER and NIAM Models. In R. Elmasri, V. Kouramajian, and B. Thalheim, editors, *Entity-Relationship Approach — ER'93*, Volume 823 of *Lecture Notes in Computer Science*, pp. 242–256. Springer Verlag, 1993.
16. V. S. Lai, J.-P. Kuilboer, and J. L. Guynes. Temporal Databases: Model Design and Commercialization Prospects. *DATA BASE*, 25(3):6–18, 1994.
17. P. McBrien, A. H. Seltveit, and B. Wangler. An Entity-Relationship Model Extended to Describe Historical Information. In *International Conference on Information Systems and Management of Data*, pp. 244–260, 1992.
18. A. Narasimhalu. A Data Model for Object-Oriented Databases with Temporal Attributes and Relationships. Technical report, National University of Singapore, 1988.
19. J. Peckham and F. Maryanski. Semantic Data Models. *ACM Computing Surveys*, 20(3):153–189, 1988.
20. M. Schrefl, A. M. Tjoa, and R. R. Wagner. Comparison-Criteria for Semantic Data Models. In J. K. Aggarwal, editor, *First International Conference on Data Engineering*, pp. 120–125, 1984.
21. R. T. Snodgrass. The Temporal Query Language TQuel. *ACM Transactions on Database Systems*, 12(2):247–298, 1987.
22. B. Tauzovich. Toward Temporal Extensions to the Entity-Relationship Model. In *10th International Conference on the Entity Relationship Approach*, pp. 163–179, 1991.
23. C. I. Theodoulidis, P. Loucopoulos, and B. Wangler. A Conceptual Modelling Formalism for Temporal Database Applications. *Information Systems*, 16(4):401–416, 1991.
24. Y. Wand and R. Weber. An Ontological Evaluation of Systems Analysis and Design Methods. In E. D. Falkenberg and P. Lindgreen, editors, *Information Systems Concepts: An In-depth Analysis*, pp. 79–107, 1989.
25. Y. Wand and R. Weber. Toward a Theory of The Deep Structure of Information Systems. In J. I. DeGross, M. Alavi, and H. Oppelland, editors, *International Conference on Information Systems*, pp. 61–71, 1990.
26. Y. Wand and R. Weber. On the Ontological Expressiveness of Information Systems Analysis and Design Grammars. *Journal of Information Systems*, 3(4):217–237, 1993.
27. R. Weber and Y. Zhang. An Analytical Evaluation of NIAM's Grammar for Conceptual Schema Diagrams. *Information Systems Journal*, 6(2):147–170, 1996.
28. E. Zimanyi, C. Parent, S. Spaccapietra, and A. Pirotte. TERC+: A Temporal Conceptual Model. In *International Symposium on Digital Media Information Base*, 1997.