Entity-Relationship Modeling:

Historical Events, Future Trends, and Lessons Learned

Peter P. Chen

Computer Science Department
Louisiana State University
Baton Rouge, LA 70803, USA
E-mail: pchen@lsu.edu

Abstract. This paper describes the historial developments of the ER model from the 70's to recent years. It starts with a discussion of the motivations and the environmental factors in the early days. Then, the paper points out the role of the ER model in the Computer-Aided Software Engineering (CASE) movement in the late 80's and early 90's. It also describes the possibility of the role of author's Chinese culture heritage in the development of the ER model. In that context, the relationships between natural languages (including Ancient Egyptian hieroglyphs) and ER concepts are explored. Finally, the lessons learned and future directions are presented.

1 Introduction

Entity –Relationship (ER) modeling is an important step in information system design and software engineering. In this paper, we will describe not only the history of the development of the ER approach but also the reactions and new developments since then. In this perspective, this paper may be a little bit different from some other papers in this volume because we are not just talking about historical events that happened twenty or thirty years ago, we will also talk about the consequences and relevant developments in the past twenty-five years. At the end, we will talk about lessons learned during this time period. In particular, we intend to show that it is possible that one concept such as the ER concept can be applied to many different things across a long time horizon (for more than twenty-five years) in this fast-changing Information Technology area.

This paper is divided into 8 sections. Section 1 is the Introduction. In Section 2, the historical background and events happened around twenty-five years ago will be explained. For example, what happened at that time, what the competing forces were, and what triggered researchers like the author to work on this topic will be explained. Section 3 describes the initial reactions in the first five years from 1976 to 1981. For example, what the academic world and the industry viewed the ER model initally? Section 4 states the developments in the next tweny years from 1981 to 2001. In particular, the role of the ER model in the Computer-Aided Software Engineering (CASE) will be discussed. Section 5 describes a possible reason for the author to come up with the ER modeling idea , that is, the author's Chinese culture heritage. The author did not think about this particular reason until about fifteen years ago. Section 6 presents our view of the future of ER modeling. Section 7 states the lessons learned. For those of you who have similar experience in the past twenty-five years, you probably have recognized similar principles and lessons in this section. For those who just started their professional careers recently, we hope the lessons learned by the author will be helpful to those readers. Section 8 is the conclusion.

2 Historical Background

In this section, we will look at the competing forces, the needs of the computer industry at that time, how the ER model was developed, and the main differences between the ER model and the relational model.

2.1 Competing Forces

First, Let us look at the competing forces in the computer software area at that time. What are the competing forces then? What triggered people like the author to work on this area (data models) and this particular topic (ER modeling)? In the following, we will discuss the competing forces in the industry and in the academic world in the early 70's,

Competing Forces in the industry. There were several competing data models that had been implemented as commercial products in the early 70's: the file system model, the hierarchical model (such as IBM's IMS database system), and the Network model (such as Honeywell's IDS database system). The Network model, also known as the CODASYL model, was developed by Charles Bachman, who received the ACM Turing Award in 1973. Most organizations at that time used file systems, and not too many used database systems. Some people were working on developing better data or index structures for storing and retrieving data such as the B+-tree by Bayer and McGreight [1].

Competing Forces in the Academic World. In 1970, the relational model was proposed, and it generated considerable interest in the academic community. It is correct to say that in the early 70's, most people in the academic world worked on relational model instead of other models. One of the main reasons is that many professors had a difficult time to understand the long and dry manuals of commercial database management systems, and Codd's relational model paper [2] was written in a much more concise and scientific style. For his contributions in the development of the relational model, Codd received ACM Turing Award in 1981.

Most People were working on DBMS Prototypes. Many people at that time in the academic world or in the industry worked on the implementation of database management system prototypes. Most of them were based on the relational model.
Most Academic People were investigating the definitions and algorithms for the Normal Forms of Relations. A lot of academic people worked on normalization of relations because only mathematical skills were needed to work on this subject. They could work on the improvement of existing algorithms for well-defined normal forms. Or, they could work on new normal forms. The speed of research moved very fast in the development of normal forms and can be illustrated by the following scenario. Let us say that several people were ready to publish their results on normal forms. Assuming that one person published a paper on 4th normal form and another person who had written a paper on 4th normal form but had not published it yet, the 2nd person would have changed the title of the paper from 4th normal form to 5th normal form. Then, the rest would work on the 6th normal form. This became an endless game till one day somebody wrote a paper claiming that he had an infinity-th normal form and arguing that it did not make any sense to continue this game. Most practitioners also said loudly that any relational normal form higher than 3rd or 4th won't have practical significance. As a result, the game of pursuing the next normal form finally ran out of steams.

2.2 Needs of the System Software in the Early 70's

The Needs of the Hardware/Software Vendors. In terms of software vendors at that time, there were urgent needs for (1) integration of various file and database formats and (2) incorporating more "data semantics" into the data models.

The Needs of the User Organizations. For user organizations such as General Motors and Citibank, there were urgent needs for (1) a unified methodology for file and database design for various file and database system available in the commercial market and (2) incorporation of more data semantics including business rules into the requirements and design specifications.

2.3 How the ERM was Developed

Here, we will give some personal history of the development of the ER model: where the author was and what the author did in the early 70's, particularly on how the author developed the ER model.

Harvard (Sept. '69 to June '73). After the author got a B.S. in Electrical Engineering from National Taiwan University in 1968, the author received a fellowship to study Computer Science (at that time, it was a part of Applied Mathematics) at Harvard graduate school. The author received the Ph.D. degree in 1973. The thesis was very mathematically oriented – focusing on the file allocation problems in a storage hierarchy using the queuing theory and mathematical programming techniques. The knowledge the author learned in EE, CS and applied math was crucial in the development of the ER model in subsequent years.

Honeywell and Digital (June '73 to August '74). The author joined Honeywell Information Systems in Waltham, MA in June '73. He participated in the "next-generation computer system" project to develop a computer system based on distributed system architecture. There were about ten people in the team, and most of them were at least twenty years senior than the author. The team consisted of several well-known computer experts including Charles Bachman. One of the requirements of such a "distributed system" was to make the files and databases in different nodes of the network compatible with each other. The ER model was motivated by this requirement. Even though the author started to crystallize the concepts in his mind when he worked for Honeywell, he did not write or speak to anyone about this concept then. Around June of 1994, Honeywell abandoned the "next-generation computer system" project, and all the project team members went different ways. The author then spent three months at Digital Equipment Corporation in Maynard, MA to develop a computer performance model for the PDP-10 system.

MIT Sloan School of Management (1974 – 1978). In September 1974, the author joined MIT Sloan School of Management as an Assistant Professor. This was the place that he put the ER ideas down into an article. Being a professor in a business/management school provided the author many opportunities to interact with the user organizations. In particular, he was particularly impressed by a common need of many organization to have a unified methodology for file structure and database design. This observation certainly influenced the development of the ER model. As a result, the first ER paper was first presented at 1st International Conference on Very Large Databases in 1975 and subsequently published in the first issue of ACM Transactions on Database Systems [3] in March of 1976.

2.4 Fulfilling the Needs

How did the ER model fulfill the needs of the vendor and user organizations at that time? We will first start with the graphical representation and theoretical foundations of the ER model. Then, we will explain the significant differences between the ER model and the relational model.

The Concepts of Entity, Relationship, Types, and Roles. In Fig. 1, there are two entities; both of them are of the "Person" type. There is a relationship called, "is-married-to," between these two persons. In this relationship, each of these two Person entities has a role. One person plays the role of "husband," and another person plays the role of "wife."

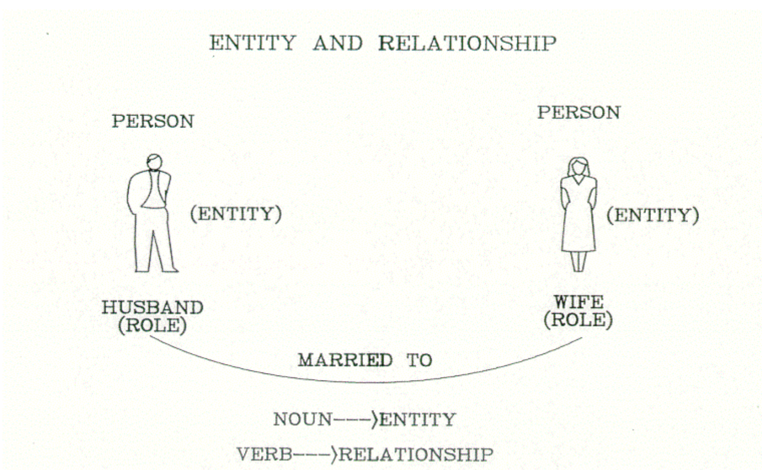


Fig. 1. The Concept of Entity and Relationship

The Entity-Relationship (ER) Diagram. One of the key techniques in ER modeling is to document the entity and relationship types in a graphical form called, Entity-Relationship (ER) diagram. Figure 2 is a typical ER diagram. The entity types such as EMP and PROJ are depicted as rectangular boxes, and the relationship types such as WORK-FOR are depicted as a diamond-shaped box. The value sets (domains) such as EMP#, NAME, and PHONE are depicted as circles, while attributes are the "mappings" from entity and relationships types to the value sets. The cardinality information of relationship is also expressed. For example, the "1" or "N" on the lines between the entity types and relationship types indicated the upper limit of the entities of that entity type participating in that relationships.

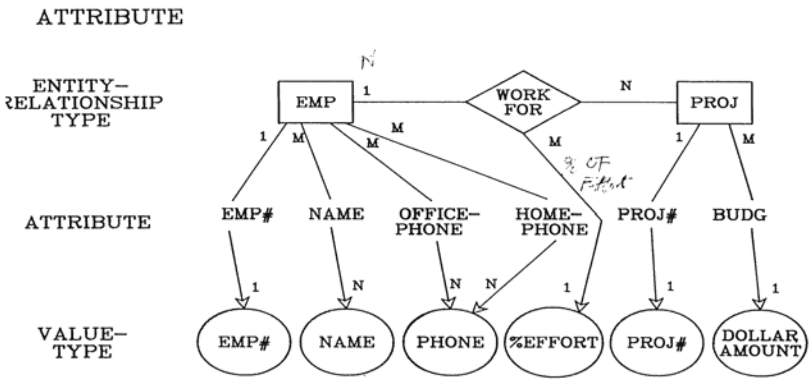


Fig. 2. An Entity-Relationship (ER) Diagram

ER Model is based on Strong Mathematical Foundations. The ER model is based on (1) Set Theory, (2) Mathematical Relations, (3) Modern Algebra, (4) Logic, and (5) Lattice Theory. A formal definition of the entity and relationship concepts can be found in Fig. 3.

SET THEORY (DEFINITIONS)

ENTITY	e
ENTITY SET	$E;\ e \in E$
VALUE	v
VALUE SET	$V;\ v \in V$
RELATIONSHIP	r
RELATIONSHIP SET	$R;\ r \in R$

A RELATIONSHIP SET IS DEFINED AS A "MATHAMATICAL RELATION" ON ENTITY SETS

$$R = \{r_1, r_2, \ldots, r_n\}$$

$$r_i = [e_{i1}, e_{i2}, \ldots, e_{in}] \mid e_{i1} \in E_1, \ldots, e_{in} \in E_n$$

Fig. 3. Formal Definitions of Entity and Relationship Concepts

Significant Differences between the ER model and the Relational Model. There are several differences between the ER model and the Relational Model:

ER Model uses the Mathematical Relation Construct to Express the Relationships between Entities. The relational model and the ER model both use the mathematical structure called Cartesian product. In some way, both models look the same – both use the mathematical structure that utilizes the Cartesian product of something. As can be seen in Figure 3, a relationship in the ER model is defined as an ordered tuple of "entities." In the relational model, a Cartesian product of data "domains" is a "relation," while in the ER model a Cartesian product of "entities" is a "relationships." In other words, in the relational model the mathematical relation construct is used to express the "structure of data values," while in the ER model the same construct is used to express the "structure of entities."

ER Model Contains More Semantic Information than the Relational Model. By the original definition of relation by Codd, any table is a relation. There is very little in the semantics of what a relation is or should be. The ER model adds the semantics of data to a data structure. Several years later, Codd developed a data model called RM/T, which incorporated some of the concepts of the ER model.

ER Model has Explicit Linkage between Entities. As can be seen in Figures 2 and 4, the linkage between entities is explicit in the ER model while in the relational model is implicit. In addition, the cardinality information is explicit in the ER model, and some of the cardinality information is not captured in the relational model.

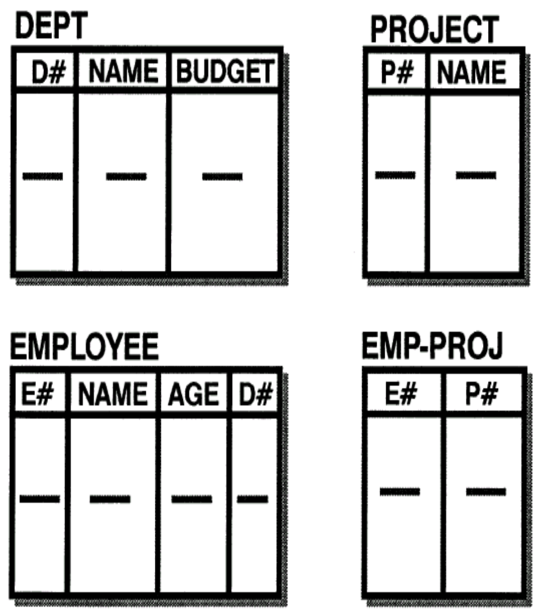


Fig. 4. Relational Model of Data

3. Initial Reactions & Reactions in the First Five Years (1976 – 1981)

3.1 First Paper Published & Codd's Reactions

As stated before, the first ER model paper was published in 1976. Codd wrote a long letter to the editor of ACM Transaction on Database Systems criticizing the author's paper. The author was not privileged to see the letter. The editor of the Journal told the author that the letter was very long and single-spacing. In any case, Dr. Codd was not pleased with the ER model paper. Ironically, several years later, Codd proposed a new version of the relational data model called RM/T, which incorporated some concepts of the ER model. Perhaps, the first paper on the ER model was not as bad as Codd initially thought. Furthermore, in the 90's, the Codd and Date consulting group invited the author to serve as a keynote speaker (together with Codd) several times in their database symposia in London. This indicates that the acceptance of ER model was so wide spread so that initial unbelievers either became convinced or found it difficult to ignore.

3.2 Other Initial Reactions and Advices

During that time, there was a "religious war" between different camps of data models. In particular, there was a big debate between the supporters of the Relational model and that of the Network model. Suddenly, a young assistant professor wrote a paper talking about a "unified data model." In some sense, the author was a “new kid on the block” being thrown into the middle of a battle between two giants. The advice the author got at that time was: “why don’t you do the research on the n-th normal form like most other researchers do? It would be much easier to get your normal form papers published.” That was an example of the type of advices the author got at that time. Even though those advices were based on good intensions and wisdom, the author did not follow that type of advices because he believed that he could make a more significant contribution to the field by continuing working on this topic (for example, [4-13]). It was a tough choice for a person just starting the career. You can imagine how much problems or attacks the author had received in the first few years after publishing the first ER paper. It was a very dangerous but a very rewarding decision the author made that not only had a significant impact on the author’s career but also the daily practices of many information-modeling professionals.

3.3 IDEF, ICAM, and Other Believers

There were a small but growing number of believers of the ER or similar data models. For example, Mike Hammer, who was an Assistant Professor at the EECS department of MIT, developed the Semantic Data Model with his student, Dennis McCleod. Later on, Hammer applied the idea in reverse engineering in the IT field to organization restructuring and became a management guru. Outside of the academic world, the industry and government agencies began to see the potential benefits of ER modeling. In the late 70’s, the author served as a consultant in a team that developed the data modeling methodology for the ICAM (Integrated Computer-Aided Manufacturing) project sponsored by the U.S. Air Force. One of the objectives was to develop at least two modeling methodologies for modeling the aircraft manufacturing processes and data: one methodology for process modeling and one for data modeling. The data modeling methodology was called IDEF1 methodology and has been used widely in US military projects.

3.4 Starting a Series of ER Conferences

The first ER conference was held in UCLA in 1979. We were expecting 50 people, but 250 to 300 people showed up. That was a big surprise. Initially, the ER conference was a bi-annual event, but now it is an annual event being held in different parts of the world [14]. In November of this year (Year 2001), it will be held in Japan [15], and next year (Year 2002) it will be held in Finland. This series of conferences has become a major annual forum for exchanging ideas between researchers and practitioners in conceptual modeling.

4 The Next Twenty Years (’81 –’01)

4.1 ER Model Adopted as a Standard for Repository Systems and ANSI IRDS.

In the 80’s, many vendors and user organizations recognized the need for a repository system to keep track of information resources in an organization and to serve as the focal point for planning, tracking, and monitoring the changes of hardware and software in various information systems in an organization. It turned out that the ER model was a good data model for repository systems. Around 1987, ANSI adopted the ER model as the data model for Information Resource Directory Systems (IRDS) standards. Several repository systems were implemented based on the ER model including IBM’s Repository Manager for DB2 and DEC’s CDD+ system.

4.2 ER Model as a Driving Force for Computer-Aided Software Engineering (CASE) tools and Industry

Software development has been a nightmare for many years since the 50’s. In the late 80’s, IBM and others recognized the needs for methodologies and tools for Computer-Aided Software Engineering (CASE). IBM proposed a software development framework and repository system called, AD Cycle and the Repository Manager that used the ER model as the data model. The author was one of the leaders who actively preached the technical approach and practical applications of CASE. In 1987, Digital Consulting Inc. (DCI) in Andover, Mass., founded by Dr. George Schussel, organized the 1st Symposium on CASE in Atlanta and invited the author to be one of the two keynote speakers. To everybody’s surprise, the symposium was a huge commercial success, and DCI grew from a small company to a major force in the symposium and trade show business.

4.3 Object-Oriented (OO) Analysis Techniques are Partically Based on the ER Concepts
It is commonly acknowledged that one major component of the object-oriented (OO) analysis techniques are based on the ER concepts. However, the "relationship" concept in the OO analysis techniques are still hierarchy-oriented and not yet equal to the general relationship concept advocated in the ER model. It is noticeable in the past few years that the OO analysis techniques are moving toward the direction of adopting a more general relationship concept.

4.4 Data Mining is a Way to Discover Hidden Relationships

Many of you have heard about data mining. If you think deeply about what the data mining actually does, you will see the linkage between data mining and the ER model. What is data mining? What does the data mining really is doing? In our view, it is a discovery of "hidden relationships" between data entities. The relationships exist already, and we need to discover them and then take advantage of them. This is different from conventional database design in which the database designers identify the relationships. In data mining, algorithms instead of humans are used to discover the hidden relationships.
.

5 In Retrospect: Another Important Factor – Chinese Culture Heritage

5.1 Chinese Culture Heritage
Many people asked the author how he got the idea of the Entity-Relationship model. After he kept on getting that kind of questions, the author thought it might be related to something that many people in Western culture may not have. After some soul searching, the author thought it could be related to his Chinese culture heritage. There are some concepts in Chinese character development and evolution that are closely related to modeling of the things in the real world.

Here is an example. Figure 5 shows the Chinese characters of "sun", "moon, and "person". As you can see, these characters are a close resemblance of the real world entities. Initially, many of the lines in the characters are made of curves. Because it was easier to cut straight lines on oracle bones, the curves became straight lines. Therefore, the current forms of the Chinese characters are of different shapes.

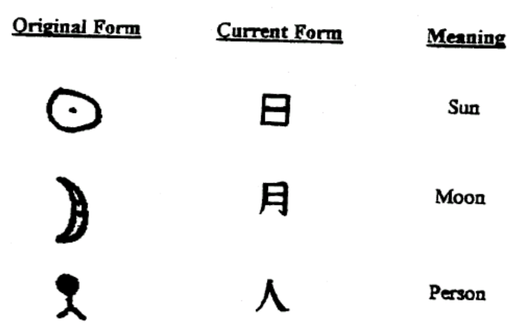


Fig. 5. Chinese Characters that Represent the Real-World Entities

Chinese characters also have several principles for "composition." For example, Figure 6 shows how two characters, SUN and MOON, are composed into a new character. How do we know the meaning of the new character? Let us first think: what does sun and moon have in common? If your answer is: both reflect lights, it is not difficult to guess the meaning of the new character is "brightness." There are other principles of composing Chinese characters [10].

日 (sun) + 月 (moon) = 明 (Bright/ Brightness by light)

Fig. 6. Composition of Two Chinese Characters into a New Chinese Character

What does the Chinese character construction principles have to do with ER modeling? The answer is: both Chinese characters and the ER model are trying to model the world – trying to use graphics to represent the entities in the real world. Therefore, there should be some similarities in their constructs.

5.2 Ancient Egyptian Hieroglyphs

Besides Chinese characters, there are other languages have graphic characters. Ancient Egyptian language is one of them. It turns out that there are several characters in ancient Egyptian characters are virtually the same as the Chinese characters. One is "sun", another is "mouth, and the third one is "water." It is amazing that both the Egyptian people and the Chinese people developed very similar characters even though they were thousands of miles away and had virtually no communication at that time. Ancient Egyptian Hieroglyphs also have the concept of composition. Interested readers should refer to [11].

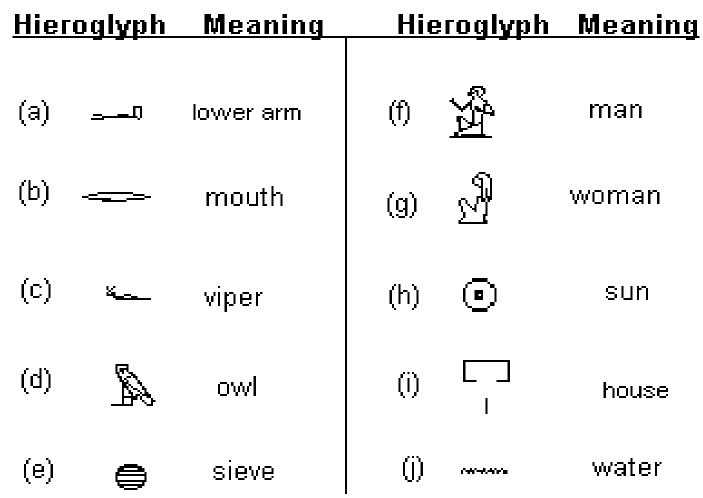


Fig. 7. Ancient Egyptian Hieroglyphs

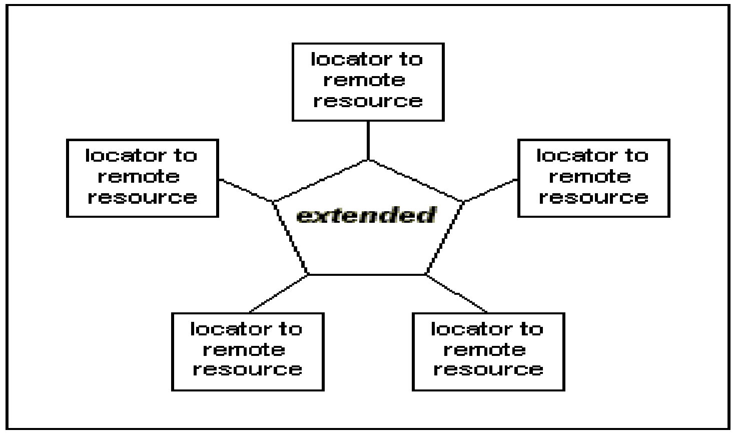
6 The Future

6.1. XML and ER Model.

In the past few years, the author has been involved in the developing the "standards" for XML. He has participated in two XML Working Groups of the World Wide Web Consortium (W3C) as an invited expert. During this involvement, some similarities between XML and the ER model were discovered including the following:

RDF and the ER Model. There are several components in the XML family. One of them is RDF, which stands for Resource definition Framework. This is a technology that Tim Berners-Lee, the Director of W3C, pushes very hard as a tool for describing the meta-data in the web. There are some similarities and differences between RDF and the ER model, and Mr. Berners-Lee has written several articles discussing this issue. In a joint meeting of the RDF and Schema Working Groups over one year ago, they issued the Cambridge Communiqué [16] that states: "…RDF can be viewed as a member of the Entity-Relationship model family…"

XLink and the ER model. Most of us are familiar with the hyperlink in HTML. The XLink Working Group of W3C has been trying to do is to develop a new kind of hyperlink for XML. In HTML, the hyperlink is basically a "physical pointer" because it specifies the exact URL of the target. In XLink, the new link is one step closer to a "logical pointer." In the evolution of operating systems, we have been moving from physical pointers to logical pointers. The XLink Working Group proposed a new structure called, "extended link." For example, Fig. 8 is an extended link for five remote resources. The extended link concept in XML is very similar to the n-ary relationship concept in the ER model. Figure 8 can be viewed as a relationship type defined on 5 entity types.



.

Fig. 8. "Extended Link" in XML is Similar to the N-ary Relationship Concept in the ER Model

6.2. Theory of the Web

One thing that is still missing today is the theory of the web. The ER model could be one of the foundations for the theory of the Web. The author plans to work on that topic and would encourage the readers to work on the subject, too.

7 Lesson Learned

7.1 Reflections on Career choices

In the past twenty-five years, the author made some tough career choices as some of the other authors in this volume did. It is the hope of the author that our experience will be useful to some other people who just started their professional careers and are making their career choices. Here are some reflections based on the author's own experience:

Right idea, right place, right time, and belief in yourself. In order to have your idea be accepted by other people, you need not only to have the right idea but also to present them at the right place and right time. You also need "persistence." In other words, you need to believe in yourself. This is probably the most difficult part because you have to endure some unnecessary pressures and criticisms when you are persistent on your idea and try to push it forward. Hopefully, some days in the future, you will be proved to be right. At that time, you will be happy that you have persisted.

Getting Fresh Ideas from Unconventional Places. After working on a particular area for a while, you may run out of "big" ideas. You may still have some "good" ideas to get you going, but those ideas are not "earth-breaking." At that time, you need to look for ideas in different subject areas and to talk to new people. For example, most of us are immersed in Western culture, and learning another culture may trigger new ways of thinking. Similarly, you may look into some fields outside of information technology such as Physics, Chemistry, Biology, or Architecture to find fresh ideas. By looking at the theories, techniques, and approaches used in other fields, you may get very innovative ideas to make a breakthrough in the IT field.

7.2 Implications of the Similarity and differences between the Chinese Charactors and Ancient Egyptian Hieroglyphs on Software Engineering and Systems Development Methodologies

As we pointed out earlier, there are several Chinese characters that are almost the same as their counterparts in ancient Egyptian hieroglyphs. What does this mean? One possible answer is that human beings think alike even though there was virtually no communication between ancient Chinese people and ancient Egyptian people. It is very likely that the way to conceptualize basic things in the real world is common to most of the races and cultures. As was discussed earlier, the construction and developments of other characters are different in Chinese and in Ancient Egyptian Hieroglyphs. It is valid to say that the language developments were dependent on the local environment and culture. What is the implication of the similarities and differences in character developments on the development of software engineering and information system development methodologies? The answer could be: some basic concepts and guidelines of software engineering and system development methodologies can be uniformly applied to all people in the world while some other parts of the methodologies may need to be adapted to local cultures and customs.

8. Conclusions

The author was very fortunate to have the opportunity to meet the right people and to be given the opportunity to develop the Entity-Relationship (ER) model at the time and environment such a model was needed. The author is very grateful to many other researchers who have continued to advance the theory of the ER approach and to many software professionals who have practiced ER modeling in their daily jobs in the past twenty-five years. We believe that the concepts of entity and relationship are very fundamental concepts in software engineering and information system development. In the future, we will see new applications of these concepts in the Web and other new frontiers of the software world.

References

1. Bayer, R. and McGreight, E., “Organization and Maintenance of Large Ordered Indexes,” Acta Informatica, Vol. 1, Fasc. 3, 1972, pp. 173-189.
2. Codd, E. F. “The Relational Model of Data for Large Shared Data Banks,” Comm. of the ACM, Vol. 13 (6), 1970, pp. 377-387.
3. Chen, P.P., "The Entity-Relationship Model: Toward a Unified View of Data," ACM Trans. on Database Systems, Vol.1, No.1, March 1976, pp. 1-36.
4. Chen, P. P., "An Algebra for a Directional Binary Entity-Relationship Model," IEEE First International Conference on Data Engineering, Los Angeles, April 1984, pp. 37-40.
5. Chen, P. P., "Database Design Using Entities and Relationships," in: S. B. Yao (ed.), Principles of Data Base Design, Prentice-Hall, NJ, 1985, pp. 174-210.
6. Chen, P. P., "The Time-Dimension in the Entity-Relationship Model," in: Information Processing '86, H. -J. Kugler (ed.), North-Holland, Amsterdam, 1986, pp. 387-390.
7. Chen, P. P. and Zvieli, A., "Entity-Relationship Modeling of Fuzzy Data," Proceedings of 2nd International Conference on Data Engineering, Los Angeles, February 1986, pp. 320-327.
8. Chen, P. P. and Li, M., "The Lattice Concept in Entity Set," in: Entity-Relationship Approach, S. Spaccapietra (ed.), North-Holland, Amsterdam, 1987, pp. 311-326.
9. Chandrasekaran, N., Iyengar, S.S., and Chen, P. P., "The Denotational Semantics of the Entity-Relationship Model," International Journal of Computer Mathematics, 1988, pp. 1-15.
10. Chen, P. P., “English, Chinese and ER Diagrams,” Data & Knowledge Engineering, Vol. 23, No. 1, June 1997, pp. 5-16.
11. Chen, P. P., “From Ancient Egyptian Language to Future Conceptual Modeling,” in: Conceptual Modeling: Current Issues and Future Directions, Chen, P.P., et al. (eds), Springer-Verlag, Berlin, Lecturing Notes in Computer Sciences, No. 1565, 1998, pp. 57-66.
12. Yang, A. and Chen, P. P., "Efficient Data Retrieval and Manipulation using Boolean Entity Lattice," Data & Knowledge Engineering, Vol. 20, 1996, pp.211-226.
13. http://www.csc.lsu.edu/~chen/
14. http://www.er2000.byu.edu/
15. http://www.arislab.dnj.ynu.ac.jp/ER2001
16. http://www.w3.org/TR/schema-arch