 ISSN: 2473-3857
v3 n4365

Media versus Message: Choosing The ER Diagram To Teach ER Modeling

Peter Y. Wu
wu@rmu.edu

Diane A. Igoche
igoche@rmu.edu

Peter J. Drauss
drauss@rmu.edu

Department of Computer and Information Systems
Robert Morris University
Moon Township, PA

Abstract

It is essential to introduce database analysis and design in the undergraduate IS curriculum even though there may be different opinions about how much design should be included there. Due to the lack of experience, it is unrealistic to expect an outcome of skillful database design, but the fundamental concepts of Entity-Relationship (ER) modeling for relational database design should be covered. Our goal is to enable the student to interpret the ER Diagram and thus understand the ER model and the database. In search of the right ER Diagram convention to teach ER Modeling at the undergraduate level, we critically reviewed several prevailing ERD conventions presently in use: the traditional ERD, the Bachman notation and briefly about the Unified Modeling Language (UML) as well. Each of these ERD approaches has its strengths and weaknesses. We contend that the traditional ERD is good when introducing the ERM concepts, but for efficacy in professional use the Bachman ERD is much better, and suits itself to interactive graphical user interfaces for tools on the computer. UML is still primarily an object-oriented software design tool: appropriate for incorporating the database into software design and development but not for teaching ERM at the undergraduate level. In conclusion, we propose a staged teaching plan beginning with the traditional ERD gradually migrating into the Bachman notation.

Keywords: Entity-Relationship Diagram, ERD, Entity-Relationship Modeling, ERM, Database Design, Bachman Notation, Unified Modeling Language, UML.

1. INTRODUCTION

A database course is required in practically every undergraduate IS curriculum. The course most often covers the use of relational database, with SQL commands and sometimes also in application programming. There is however less consensus in how much database design should be covered. While there has been no shortage of research literature regarding the issue, including the on-going debate of whether it is better to teach use first and then design or design concepts first and then use of database, we would accept that the lack of experience in undergraduate students will make it unrealistic to expect skillful database design ability in the course or program outcome. However, it is still important to introduce database analysis and design bringing in the fundamental concepts of Entity-Relationship Modeling (ERM). Our goal is to enable the

ISSN: 2473-3857
v3 n4365

students to read and interpret the ER Diagram (ERD). These modeling concepts allow the students to relate the use of relational database to the business of real life examples and applications. Yet in the evolving history of ERM, various ERD conventions have developed and are now used in practice. The commonly used ones include the traditional ERD, the Bachman notation and the Unified Modeling Language (UML). Each of these may have its particular strengths and efficacy in use, but the variety may also become a source of confusion for some students.

This paper reports our study in the search of a good ERD to teach ERM. We critically reviewed two major ERD conventions to argue for or against the issues involved in bringing out the fundamental concepts in modeling for database design. Our hope is to guide the students to become reasonably familiar with prevailing ERD convections while introducing the ERM concepts.

We will begin with briefly recounting the history of ER modeling in the next section. We intend our brief description to high-light the few prevailing ERD conventions commonly in use. Section 3 will go into the various ERM conceptual details to analyze the strengths or the weaknesses of different aspects of the ERD conventions. Our critical review focuses on the expected course outcome to enable the students to read and interpret the diagram and understand the ER model. Based on our analysis, we sum up the discussion in Section 4 to present a staged plan about introducing the various ERD conventions while progressively covering the ERM concepts. Section 5 presents our concluding discussion and desirable further work for follow up.

2. BRIEF HISTORICAL ACCOUNT

Since the proliferation of using computers for data processing in the 60's, how to organize data in the database has been an active research topic. Data modeling research with a focus on real business application started with the discussion of "entities" and the "relationships" between them (Brown 1975). Today's prevalent practices of ER modeling often give credit to Chen who first used the term Entity-Relationship Modeling – ERM (Chen 1976). To illustrate the concepts involved, Chen made use of the Entity-Relationship Diagram - ERD. Even with many attempts of extensions and changes by researchers, Chen's ERD design has been established and generally known as the traditional ERD. Credit should also be given to Bachman who approached data modeling as set description (Bachman 1974) which has been highly influential to Chen's traditional ERD design. Martin and Bachman subsequently developed variants of a more concise diagram convention, known as crow's feet notation, or the Bachman ERD (Martin 1989; Bachman 1992). The Bachman ERD and the traditional ERD are the two most commonly used in practice today.

Object-oriented programming started in the late 70's and in the 80’s became a new approach to organize software, specifically program code along with data, in a way to better harness the flexibility of software (Goldberg 1983). Booch, Rumbaugh and Jacobson (1994) joined their efforts to formulate a diagram technique known as the Unified Modeling Language (UML) as an analysis, design and development tool for object-oriented software. Although UML was originally intended for object-oriented software design, it has also been applied to database design. UML becomes yet another ERD for Entity-Relationship modeling.

3. CRITICAL REVIEW

We consider first the fundamental concepts needed to introduce ER modeling, and then some of the special features in ER Diagrams to analyze the strengths and the weaknesses thereof. The basis of our review is pedagogy: our goal being the introduction of ER modeling, our focus is therefore on the features whether they may improve or hinder understanding.

Entities and Attributes

The most fundamental in ER modeling is to identify entities and model them with the relevant attributes. Definition of an entity needs to be *specific*. Simple examples may illustrate that. The lack of experience shows up when the student is confused about whether an idea ought to be an entity or an attribute. The student who thinks deeper may understand that the crucial principle of an entity being *specific* means that every instance of the entity has identity, but the value of an attribute does not. It is therefore important for the beginner to explicitly work out the relevant attributes, helpful to sort out whether or not the entity should be an attribute, validating the definition as *specific* for an entity. We found the traditional ERD of using an oval for each attribute connected to a box for the entity set more illuminating, leading the practitioner to explicitly name the attribute and illustrate it with proper data type for its values. A simple ERM example of bank accounts and account owners is illustrated in Figure 1, showing the traditional

ISSN: 2473-3857
v3 n4365

ERD as well as the Bachman ERD. The Bachman ERD permits naming of the attributes optional, to allow for efficiently proceeding on to working out the other aspects of ER modeling such as the relationships. Compared to Bachman ERD, the traditional ERD would be more inducive for beginners to think clearly when defining an entity, and less likely to erroneously treat an attribute as an entity. Consequently, it is not hard for students to see that the entity set captures the schema of a relational table while every instance of the entity translates to a row of specific data values in the table.

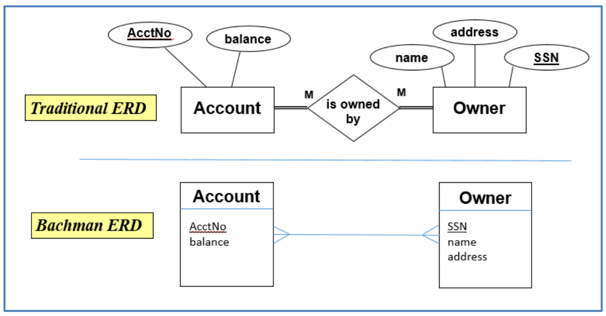

Figure 1: Bank Accounts and Account Owners

Relationships and Structural Constraints

More complicated, but still a very fundamental concept is the relationship in ER modeling. Students more astute in mathematical thinking can accurately understand the relationship as a set of ordered pairs, each partner identifying a specific entity instance from a participating entity set. The student learning ER modeling as a beginner however has to understand how the model properly captures the way business is done since that must be expressed in the ERD accordingly.

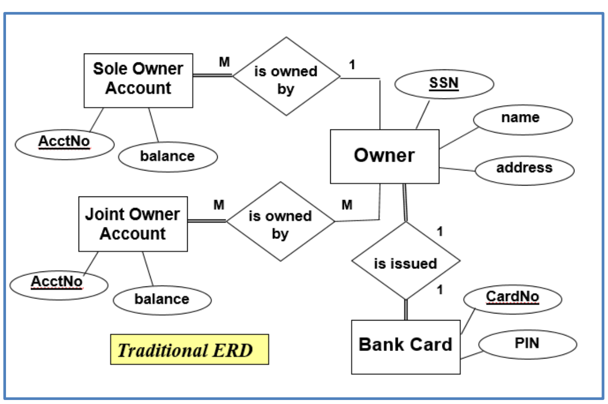

Figure 2: Relationships in the Traditional ERD

We take a simple banking business example to illustrate the relationships. There are two types of account — sole owner account and joint owner account and every owner is issued a bank card. Figure 2 depicts the traditional ERD.

The tradition ERD using the diamond to represent a relationship will help beginners in practice to see it as defining a set of relationship instances, a pair of two specific entity instances. It is crucial to understand the structural constraints to see the schema of relational tables involved in the analysis and design process. In the traditional ERD, the structural constraint consists two parts: the participation constraint and the cardinality ratio.

The participation constraint indicates the level of participation of the entity set in the relationship: partial participation — some entity (at least one) does not participate, having no part in the relationship, and total participation — every entity participates, having at least one partner. Figure 2 illustrates the simple banking scenario: every account must have at least one owner but an owner may or may not have either type of account. Every account is issued a bank card and every bank card in use must have an owner. The participation constraint does not impose much difficulty here.

The cardinality ratio classifies the relationship between the two entity sets as one-to-one, one-to-many, many-to-one or many-to-many according to the applicable business rules. It is then significant for the students to understand how the cardinality ratio affects the translation of the ERM into the logical schema of the tables to implement the database. Only the many-to-many relationship needs an extra table for the relationship; in the other cases, the tables for the entity sets in the relationship can be extended to capture the information of the relationship, not needing an extra table.

Figure 2 shows that the ownership relationship from the account owner to the sole owner account is one-to-many but that to the joint owner account is many-to-many. Inclusion of account owner information in the table for accounts works only for the sole owner accounts. A separate relational table becomes necessary to capture ownership information for joint owner accounts. The relationship between the account owner and the bank card is one-to-one: the relationship can be captured in the relational table for entity set on either side.

The traditional ERD, showing the relationship prominently, is helpful for the students learning

ISSN: 2473-3857
v3 n4365

as beginners to see the significance of the cardinality ratio in the design process.

The Bachman ERD is much more subtle here since it does away with the diamond: a connecting line between two entity sets implies a relationship there. The "chicken feet" notation at the ends of the line indicating cardinality ratio is actually easier to read for most students. The little circle indicating participation constraint, while applied to the opposite side, can also be comfortable to get used to. Figure 3 below depicts the same simple banking example using the Bachman ERD.

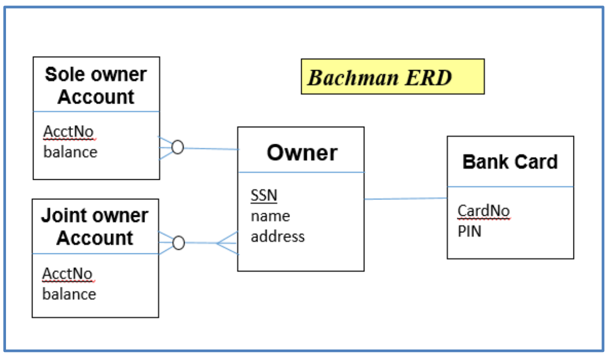

Figure 3: Relationships in the Bachman ERD

Clearly the Bachman ERD approach is much more efficient when compared to the traditional ERD in terms of effort in drawing and/or laying out the diagram. Our opinion is for the student to begin first with the traditional ERD and move on to the Bachman ERD when they become more familiar with structural constraints in the ER model.

An interesting but somewhat trivial issue is about coming up with an appropriate name for the relationship. Students learning as beginners sometimes find it a challenge to their command of the language when dealing with more sophisticated case studies. The Bachman ERD, doing away with the diamond, tends to promote giving the relationship no name at all, although one may also leave the diamond in the traditional ERD empty with no name.

The (min,max) adornment

An alternative feature to indicate the structural constraint in the ERD is the (min,max) notation. The notation adorns the diagram on each side of a relationship diamond. Hence, it defines the constraint applied to the entities participating in the relationship on each side. The (min,max) numbers are the lower and upper limits to the number of partners for any entity instance. The (min,max) notation is commonly used in the traditional ERD in place of the traditional structural constraint. In other ERD conventions, it may appear as the min..max notation to indicate the countable number of partners per instance. In either case, the notation sums up for the structural constraint and actually in some cases may give more detailed information about the partner count than the cardinality ratio. However, that often becomes more confusing for the student. The (min,max) notation is common in professional practice of ER modeling. We feel that student does not need to start with it but should learn to use and interpret it in the expected course outcome.

Weak Entities and Dependencies

The notion of weak entities provides convenience of expression in the ER model, but is not really essential in the practice of ERM. It is often easier for the students to identify the key attributes first for the entity and only discover the weak entity subsequently upon removing key attributes which are duplicated in another related entity set. This also requires the student to understand entity uniqueness and how it is relevant to the concept of keys and dependency relationships. The traditional ERD and the Bachman ERD are about the same here, except that the double diamond in traditional ERD indicates the dependency relationship much more prominently than the Bachman ERD.

Subclass and Superclass

The idea of subtypes in ER modeling originally gave rise to the concepts of specialization and generalization (Worboys, Hearnshaw & Maguire 1990). The popularity of object-oriented software design has prompted more for the adoption of the names subclass and superclass. Among the various attempted extensions to the practice of ER modeling, subclass and superclass have steadily become accepted in what is now often called Extended ER modeling (EERM). For a while various different symbols were used to denote subclass/superclass between two entity sets in EERM. The symbol more popular in use now is the triangle symbol originally adopted in object-oriented software design (Booch 1994). Both the traditional ERD and the Bachman ERD can be conveniently extended by adopting the same symbol. It is an advanced concept for students not yet exposed to object-oriented programming, but should be included in the course outcome.

UML for OO software design

Among the myriad varieties of object-oriented software design symbology and tools in the 1990's, Unified Modeling Language (Booch et al 2005) has emerged to become the bona fide standard in practice today. Although it is primarily

ISSN: 2473-3857
v3 n4365

used in the OO approach to software design, some has shown that it is also effective in ER modeling for database design. The UML class diagram can be suitable as ERD, similar to the Bachman ERD. The many artifacts in OO analysis and design such as association and containment, do add to the wealth of symbology in OO software design, but they quickly become confusing distractions for the student beginning to learn ER modeling. It is our opinion to better just allude to the use of UML and not to include its coverage.

The Relationships pane in MS SQL Server

Originally developed to include the implementation of Query By Example (Whang et al, 1987) for interactive SQL construction, MS Access (and now also in MS SQL Server) includes the Relationships Pane which is a tool to present graphically the relationships between the tables. The Relationships Pane allows the user to create and edit referential relationships between tables interactively using the graphical user interface. Students are sometimes confused to consider that as the ER diagram even though it is not. But it is useful to demonstrate the logical schema of tables as the implementation of a conceptual schema captured in the ERD.

4. A STAGED TEACHING PLAN

Originally intended for a self-paced online course, Wu, Baugh and Harvey (2006) presented a detailed plan for teaching ER modeling. The plan consists of 19 teaching modules, from M0 to M18, each with very specific objectives. We have followed and continued to make improving changes in practice. Figure 4 below shows the list of all the 19 modules, each of which we will have subsequent description.

M0	Introduction to ER modeling
M1	Entities and Attributes
M2	Types of Attributes
M3	Key and Key Attributes
M4	Tables for Entity Sets
M5	The Index Card Analogy
M6	Relationship and Relationship Instances
M7	Participation Constraint
M8	Cardinality Constraint
M9	Design Rules and Tips

M10	Tables for Relationship Sets
M11	Weak Entities and Dependency Relationship
M12	Tables for Weak Entity Sets
M13	Creating Entity Sets in Design
M14	Specialization and Generalization
M15	Tables for Extended ER Model
M16	(min,max) Structural Constraint
M17	Bachman Notation review
M18	ER Diagram using UML (optional)

Figure 4: 19 Modules in the Teaching Plan

The original plan primarily taught the traditional ERD but added the Bachman ERD along with UML as options. Based on our review, our plan now uses both: starting with the traditional ERD and gradually moving toward the Bachman ERD. UML remains optional. But the Bachman ERD is now required in a module (M17) for review. Our expected outcome now includes reasonably proficiency in interpreting both prevailing ERDs. The following further describes these teaching modules.

M0: Introduction to ER Modeling

We present a general overview of ER modeling and how it derives from the way business is done and captures the relational schema as the unchanging abstraction of the procedures in the processing of business data. Our illustration uses the traditional ERD.

M1: Entities and Attributes

We introduce the fundamental concepts in ER modeling, using the traditional ERD. The student must focus on defining an entity to be specific so that every instance can be uniquely identified. That is contrasted to the value of an attribute having no identity.

M2: Types of Attributes

We discuss different types of attributes, along with their data types. While we primarily use the traditional ERD, we may begin to introduce the Bachman ERD as an alternative, and perhaps more convenient when listed without the ovals in the traditional ERD.

M3: Key and Key Attributes

We describe the uniqueness constraint regarding every entity instance and how the unique identity is expressed when we capture that in the collection of key attributes. The particular ERD is not an issue here, but we may use either or both.

M4: Tables for Entity Sets

With only entity sets, the implementation of the ERM in the relational schema is relatively simple, with only the need to deal with data types and the key, each key attribute becoming a column in the primary key of the table. It is also appropriate to illustrate with both the traditional and Bachman ERDs.

M5: The Index Card Analogy

The idea here is a simple way to help the student visualize the ER model in use: each entity set is a deck of cards and each entity instance is a card. The uniqueness constraint therefore requires a unique set of attributes values presented on each card to model the entity with its identity properly expressed in the key attributes.

M6: Relationship and Relationship Instances

ISSN: 2473-3857
v3 n4365

The Index Card analogy is now extended to help the student visualize the concept of relationship as a collection of relationship instances. Each is a card denoted as an ordered pair identifying the two related entities. It is now important to start with the traditional ERD, with the diamond illustrating the relationship set. The Bachman ERD is not a good alternative at this point.

M7: Participation Constraint
We introduce the participation constraint in the traditional ERD with partial and total participation adornments.

M8: Cardinality Ratio
We present the cardinality ratios of one-to-one, one-to-many, many-to-one and many-to-many in the traditional ERD but discuss how they capture the essential properties of the business rules in formulating the relational schema of the database. While we focus on using the traditional ERD, the student will also take time to absorb the detailed interpretation of the cardinality ratios in practice.

M9: Design Rules and Tips
The student having completed Module 8 should then practice interpretation of the traditional ERD to understand the ERM in real applications. We may begin to guide the student with rules and tips about how to construct the ERD in the process of modeling. The exercises are mostly done in the traditional ERD.

M10: Tables for Relationship Sets
We may now definitively explain how the structure constraints, specifically the cardinality ratios, capture the unchanging properties of the business rules into the relational schema of the tables. We also begin to introduce using the Bachman ERD to show how the ERD can be more efficient in use. To familiarize ourselves with the Bachman ERD, we also go back to the exercises in M9 and re-do them using the Bachman ERD.

M11: Weak Entity and Dependence Relationship
We introduce the concept of the weak entity, and the need for the dependence relationship so that weak entities are still unique as required. We illustrate the weak entity with both the traditional ERD and the Bachman ERD, briefly showing that the traditional ERD more prominently indicates the dependence relationship.

M12: Tables for Weak Entity Sets
This module now becomes a good revision of M10 for the student to review the procedures of working out the relational schema from the ERM. We also begin to show both the traditional ERD and the Bachman ERD so that students can become familiar with both. The student can learn that the weak entity set is a convenient additional feature but it should not complicate the modeling.

M13: Creating Entity Sets in Design
We discuss the fine tuning of the ERD while reviewing all the different aspects about ER modeling. Using both the traditional ERD and the Bachman ERD. We also show how the Bachman ERD may indicate the creation of an associative entity set or an attributive entity set as a way to preserve the history of ERD development.

M14: Specialization and Generalization
We introduce the concept of subtype in ER modeling, forming the subclass or superclass of an entity set. The new symbol applies to both ERD approaches and should be illustrated easily in both, including the symbol adornments for total and disjoint specialization as well.

M15: Tables for Extended ER Model
We describe how to formulate the relational schema from the ERD with subclass/superclass. The two ERD approaches should make no difference to the student.

M16: (min,max) Notation
We introduce the (min,max) notation in place of the structural constraint as an alternative in the traditional ERD, and how we may need to include additional constraints in the relational schema when required, because of additional information in the notation. The students needs to learn it because it is also commonly used in professional practice. Since the (min,max) notation is an alternative to specify the structural constraint, the student can use the exercises of M10 again and re-do them using the new notation.

M17: The Bachman ERD
Since we start ER modeling with the traditional ERD and inject the Bachman ERD along the way, this module becomes an appropriate place to sum up and review the Bachman ERD as the modeling tool. Its use is also more prevalent in industry practice, and most of the automated computer aided design tools. The student is expected to understand ER modeling and also be reasonably proficient in interpreting the Bachman ERD.

M18: Using UML
For the sake of completeness, we introduce the use of UML for ER modeling, as an aside extended from object-oriented software design. This is therefore an optional module in our scheme.

Although we have presented these 19 modules sequentially, the pre-requisite knowledge for each module does not necessarily depend on the ones before it. Consequently, teachers as well as students in a self-paced learning situation do not need to follow the modules in sequence. Figure 5 in the Appendix depicts in a flow graph the pre-requisite dependencies between these modules. It is also our intention to make it helpful in the design of illustrative examples needed for teaching in each case. Good examples need to illustrate the main learning objectives for the module but do not involve issues on topics not yet covered. The flow graph of dependencies between these teaching modules would be helpful.

5. CONCLUSION AND FURTHER WORK

We are concerned about teaching ER modeling at the undergraduate level of the IS curriculum. Granted that students have very little or no experience, our goal as course outcome is not skillful database analysis and design, but reasonable familiarity with the process and the tools, as well as the ability to read and interpret the ERD. We reviewed two prevailing ERD approaches, namely the traditional ERD and the Bachman ERD, in view of our teaching goals and our students. We also examined more detailed features such as the options for structural constraints, weak entity sets and subclass and superclass. Our conclusion in general is that the traditional ERD is good for beginners learning ER modeling because the students are guided to work more discreetly in the process of constructing the ERD. But after achieving some familiarity, the Bachman ERD will help to bring the students to focus more on what is essential in ER modeling, with efficacy in a more concise ERD. Our analysis led us to propose the staged teaching plan, using a mixture of ERD approaches along the way.

We have presented our teaching plan mostly in abstract description. Effective teaching requires good illustrative examples and creative discussion questions in class, as well as in homework assignments. We analyzed the pre-requisite dependencies between the teaching modules and illustrated that in a flow graph (Figure 5), which we believe will help to identify good illustrative examples for use in each case. We plan to organize our teaching examples with class exercises and homework assignments, while we also collect more of the same kind into a library repository of active learning tools for the purpose. To argue for our case, it is also imperative to gather assessment results from direct assessment. Based on our critical review, we should formulate statements of hypotheses, design experiments to gather data from direct assessment as well as student survey. That is our plan to follow up our work reported here in this paper.

6. REFERENCES

Bachman, C.W. (1974). The Data Structure Set Model. *Proceedings of ACM SIGFIDET* (now SIGMOD) *workshop on Data Description, Access and Control: Data models: Data-Structure-Set vs Relational*, pp.1-10.

Bachman, C.W. (1992). Bachman Analyst, published by Bachman Information Systems, Inc.

Booch, G. (1994). Object-Oriented Analysis and Design, Addison-Wesley.

Booch, G., J. Rumbaugh & I. Jacobson. (2005). The Unified Modeling Language User Guide (2*nd* edition), Addison-Wesley.

Brown, A.P.G. (1975). Modelling a Real-World System and Designing a Schema to Represent it. Douque and Nijssen (*eds*), *Data Base Description*, North-Holland.

Chen, P.P.S. (1976). The Entity-Relationship Model - Toward a Unified View of Data, *ACM Transactions on Database Systems*, 1(1): 9–36.

Goldberg, A. (1983). Smalltalk 80: The Interactive Programming Environment, published by Addison-Wesley.

Martin, J. (1989). Information Engineering: Book II: Planning And Analysis, published by Prentice Hall.

Whang, K.Y., A. Ammann, A. Bolmarcich & M. Zloof. (1987). Office-by-Example: An Integrated Office System and Database Manager, *ACM Transactions on Information Systems*, 5(4), pp.393-427.

Williams, P. (2012). A Short History of the ER Diagram and Information Modeling, article published at *Dataversity* by Shannon Kempe, http://www.dataversity.net/a-short-history-of-the-er-diagram-and-information-modeling/.

ISSN: 2473-3857
v3 n4365

Worboys, M.F., H.M. Hearnshaw & D.J. Maguire (1990). Object-Oriented Data Modelling for Spatial Databases. International Journal of Geographical Information Systems, 4(4), pp.369-383.

Wu, P.Y., J. Baugh & V. Harvey. (2006). Self-paced Learning and Online Teaching of Entity-Relationship Modeling, *Information Systems Education Journal*, 6(10). ISSN: 1545-679X.

ISSN: 2473-3857
v3 n4365

7. APPENDIX

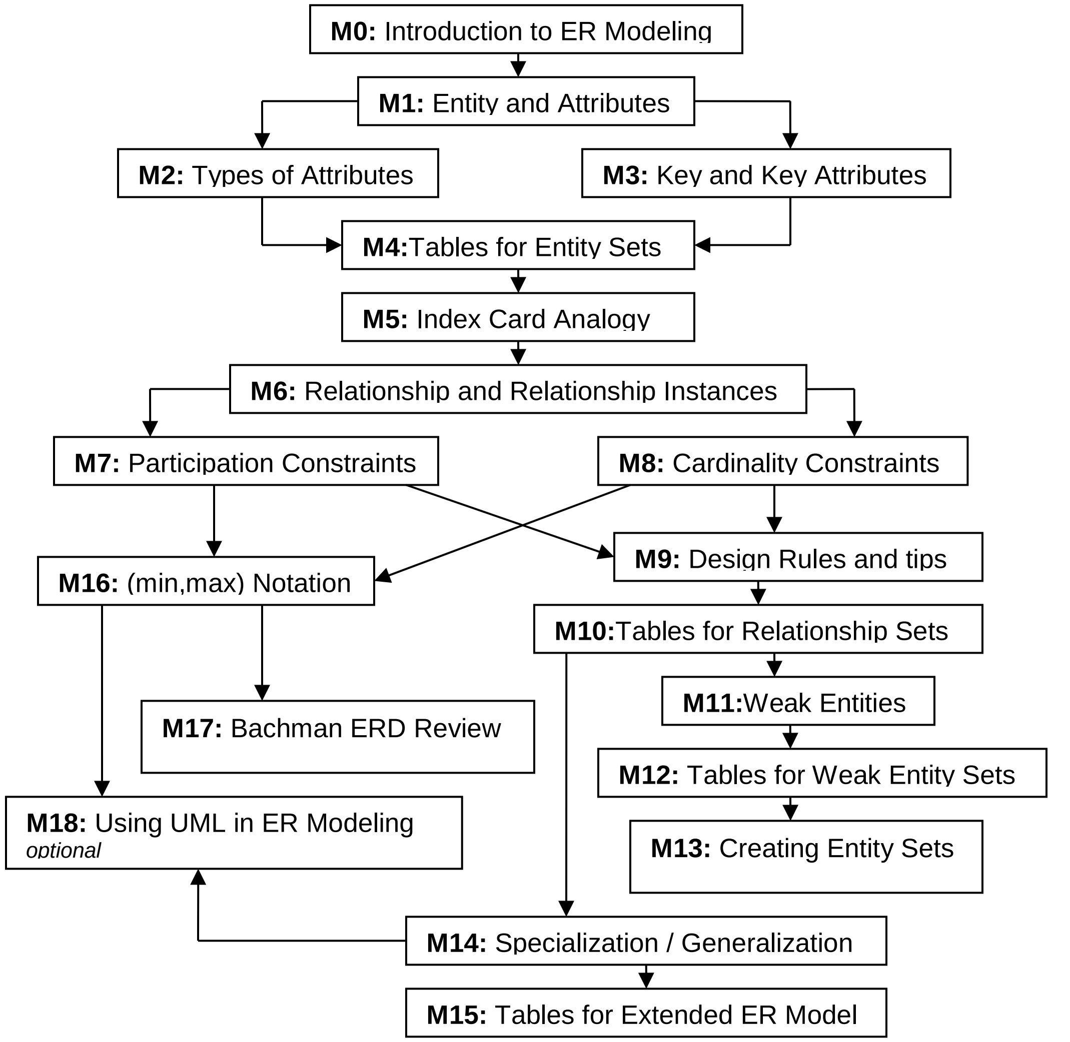

Figure 5: Dependency Flow Graph of the 19