



ISSN: 1545-679X

Information Systems Education Journal

Volume 6, Number 10

<http://isedj.org/6/10/>

February 8, 2008

In this issue:

Self-paced Learning and On-line Teaching of Entity-Relationship Modeling

Peter Y. Wu

Robert Morris University
Moon Township, PA 15108-1189 USA

Jeanne M. Baugh

Robert Morris University
Moon Township, PA 15108-1189 USA

Valerie J. H. Powell

Robert Morris University
Moon Township, PA 15108-1189 USA

Abstract: We present a detailed plan to teach Entity-Relationship modeling in an undergraduate course in Database Systems. The course syllabus fulfills the requirements of a first database course in the IS 2002 model curriculum. It is designed to be offered on-line for self-paced learning within the 15 weeks of a regular term. Our goal is to ensure the scope of coverage and the quality of delivery, and at the same time provide sufficient flexibility for student learning. Our teaching plan comprises 19 modules: from the introduction to Extended ER modeling covering specialization and generalization, including two optional modules for alternative ER diagram notations. The teaching plan is complete and yet very compact. The dependency graph governs the flow of the learning modules. Students will then have the flexibility to work out their schedules for their effective use of time and materials on-line. The critical design factor in each module is in the selected ER modeling features to cover, and very specific learning objectives. We believe our design will engender effective self-paced learning. The plan also prepares very well for on-line delivery. We are eager to further develop the plan by gathering a very large collection of illustrative examples, together with exercises and test questions.

Keywords: Entity-Relationship modeling, self-paced learning, on-line teaching

Recommended Citation: Wu, Baugh, and Harvey (2008). Self-paced Learning and On-line Teaching of Entity-Relationship Modeling. *Information Systems Education Journal*, 6 (10). <http://isedj.org/6/10/>. ISSN: 1545-679X. (Also appears in *The Proceedings of ISECON 2006*: §2143. ISSN: 1542-7382.)

This issue is on the Internet at <http://isedj.org/6/10/>

The **Information Systems Education Journal** (ISEDJ) is a peer-reviewed academic journal published by the Education Special Interest Group (EDSIG) of the Association of Information Technology Professionals (AITP, Chicago, Illinois). • ISSN: 1545-679X. • First issue: 8 Sep 2003. • Title: Information Systems Education Journal. Variants: IS Education Journal; ISEDJ. • Physical format: online. • Publishing frequency: irregular; as each article is approved, it is published immediately and constitutes a complete separate issue of the current volume. • Single issue price: free. • Subscription address: subscribe@isedj.org. • Subscription price: free. • Electronic access: <http://isedj.org/> • Contact person: Don Colton (editor@isedj.org)

2006 AITP Education Special Interest Group Board of Directors

Stuart A. Varden Pace University EDSIG President 2004		Paul M. Leidig Grand Valley State University EDSIG President 2005-2006		Don Colton Brigham Young Univ Hawaii Vice President 2005-2006	
Wendy Ceccucci Quinnipiac Univ Director 2006-07	Ronald I. Frank Pace University Secretary 2005-06	Kenneth A. Grant Ryerson University Director 2005-06	Albert L. Harris Appalachian St JISE Editor	Thomas N. Janicki Univ NC Wilmington Director 2006-07	
Jens O. Liegle Georgia State Univ Member Svcs 2006	Patricia Sendall Merrimack College Director 2006	Marcos Sivitanides Texas St San Marcos Chair ISECON 2006	Robert B. Sweeney U South Alabama Treasurer 2004-06	Gary Ury NW Missouri St Director 2006-07	

Information Systems Education Journal 2005-2006 Editorial and Review Board

Don Colton Brigham Young Univ Hawaii Editor	Thomas N. Janicki Univ of North Carolina Wilmington Associate Editor
---	--

This paper is part of the group that was selected for inclusion in the journal based on preliminary ratings in the top 30% of papers submitted, and a second review placing it in the top 15% by persons unconnected with the conference or the journal, and whose names are withheld to preserve their anonymity.

EDSIG activities include the publication of ISEDJ, the organization and execution of the annual ISECON conference held each fall, the publication of the Journal of Information Systems Education (JISE), and the designation and honoring of an IS Educator of the Year. • The Foundation for Information Technology Education has been the key sponsor of ISECON over the years. • The Association for Information Technology Professionals (AITP) provides the corporate umbrella under which EDSIG operates.

© Copyright 2008 EDSIG. In the spirit of academic freedom, permission is granted to make and distribute unlimited copies of this issue in its PDF or printed form, so long as the entire document is presented, and it is not modified in any substantial way.

Self-paced Learning and On-line Teaching of Entity-Relationship Modeling

Peter Y. Wu
wu@mail.rmu.edu

Jeanne M. Baugh
baugh@mail.rmu.edu

Valerie J. Harvey
harvey@mail.rmu.edu

Department of Computer & Information Systems
Robert Morris University
Moon Township, Pennsylvania, USA

ABSTRACT

We present a detailed plan to teach Entity-Relationship modeling in an undergraduate course in Database Systems. The course syllabus fulfills the requirements of a first database course in the IS 2002 model curriculum. It is designed to be offered on-line for self-paced learning within the 15 weeks of a regular term. Our goal is to ensure the scope of coverage and the quality of delivery, and at the same time provide sufficient flexibility for student learning. Our teaching plan comprises 19 modules: from the introduction to Extended ER modeling covering specialization and generalization, including two optional modules for alternative ER diagram notations. The teaching plan is complete and yet very compact. The dependency graph governs the flow of the learning modules. Students will then have the flexibility to work out their schedules for their effective use of time and materials on-line. The critical design factor in each module is in the selected ER modeling features to cover, and very specific learning objectives. We believe our design will engender effective self-paced learning. The plan also prepares very well for on-line delivery. We are eager to further develop the plan by gathering a very large collection of illustrative examples, together with exercises and test questions.

Keywords: Entity-Relationship modeling, self-paced learning, on-line teaching

1. INTRODUCTION

In the era of much needed on-going IS/IT training, curriculum design must attempt to accommodate the schedules of working adults. While the need for training is particularly acute for working adults already in the IS/IT industry (Barker 1994), it is also very desirable in curriculum and program design to allow greater flexibility for the students (Learmonth 2001). Granted the wide accessibility of the internet, it becomes imperative to re-design teaching plans and delivery methods to work with the internet and to provide the greatest flexibility for students in

self-paced learning. Not all subject areas will lend themselves nicely to be offered on-line with self-paced learning. This project is motivated by the intention to bring much of the courses in the IS curriculum to be offered on-line.

To begin working on the curriculum and program re-design, it is essential to address the core area of database topics (Robbert 2000, Springsteel 2000). The prevailing standards and model curricula evidently see the database systems course as part of the core in the IS curriculum (SIGCSE 2001, IS2002 2002). There has been much considerable

effort in the design and review pertaining to the teaching of database systems (Baugh 2004, Lenox and Woratschek 2005), primarily in the traditional format of course offering through lectures and lab sessions. This paper represents our attempt to re-design a practicable teaching plan to offer the course on-line for self-paced learning. Our previous paper focused on teaching SQL (Wu, Baugh and Harvey 2005). This paper focuses on teaching Entity-Relationship modeling, which is not only an important topic in the course, but it is also the essential tool to support information system analysis and design, a skill known to be highly desirable in IS/IT professionals today (Caputo et al 2004).

We first present the syllabus for the coverage of topics in the next section, and then we will present the teaching plan, discussing each teaching module in further details. We also present the dependency graph to illustrate the prerequisite requirements between each module in the teaching plan. In further discussion, we describe our development effort toward offering the database systems course on-line.

2. COURSE SYLLABUS

The context of our ER Modeling teaching plan is a first course in Database Systems for undergraduates. The course fulfills the requirements of coverage for database topics in the IS 2002 Model Curriculum (IS2002 2002). The following states the primary goal and lists the detailed topics under the objectives of the course syllabus. There are two textbooks currently adopted: Riccardi (2003) and Bordoloi and Bock (2004).

Primary Goal:

The primary goal is to introduce the concepts of relational database systems to the students, and to enable them in the use, management, and the design and implementation of databases.

Objectives:

At the completion of the course, the student is expected to be able to:

Topic 1. Database Terminology

- Define the components and terms used in database management system
- Define and discuss the types and roles of the users of a DBMS

- Define and discuss logical vs physical views of the data

Topic 2. Relational Database

- Create a relational database and its components
- Define the tables
- Define attributes and their domains
- Define keys, primary keys and foreign keys
- Create relationships among the data in the database implementation
- Define referential integrity constraints and resolution upon violation.
- Create reports from the database implementation
- Create forms for interactive use of database

Topic 3. SQL Language

- Create database schema using Data Definition Language (DDL) with SQL.
- Manipulate data using Data Manipulation Language (DML) with SQL
- Query the database using multiple tables and criteria with SQL: such as select with join operations and various selection criteria.
- Use sub-queries in SQL

Topic 4. Normalization

- Understand anomalies and redundancy based on the concept of functional dependencies
- Recognize normal forms (up to 3rd) and how they are relevant to good design
- Use the process of normalization to improve schema design

Topic 5. Database Design

- Define and discuss the problems of data redundancy, and data inconsistency
- Define the 4 main data models used in DBMS and describe their main features
- Describe the major concepts in the analysis and design of a database.
- Utilize software to create a database design.
- Practice Entity-Relationship (ER) modeling for analysis and design
- Understand entities and the concepts of attributes, uniqueness, and key
- Identify relationships, and the concepts of structural constraints
- Use the concept of weak entities, specialization and generalization, in design
- Define constraints on the data, including semantic constraints

- Evaluate a given data base design for a specific application.
- Use of Bachman Diagram for ER modeling and database design
- Translate of a conceptual schema in ER model to a relational schema for data-base implementation

Topic 6. Database Systems Administration

- Define and discuss responsibilities of the database manager and administrator.
- Define and discuss relevance of the data dictionary.
- Define and discuss critical elements for implementing and maintaining a data-base.
- Define and discuss laws governing information practices in the United States and their application to database systems.
- Discuss security issues in the implementation of a database.
- Define and discuss the terms, programming aspects, hardware and software requirements, and the various views for each of the major models of databases.
- Define and discuss the architecture and the components of a database system, and the various interfaces.

To summarize, the course covers the relational data model, use of SQL, normal forms and normalization, Entity-Relationship modeling (ER modeling) for the design and implementation of databases, and then the administration and management of database systems. Evidently, the requirements for database topics in the IS 2002 model curriculum (IS2002 2002) are sufficiently fulfilled. The following table illustrates the time frame to cover these topics in the full 15 weeks term.

Week(s)	Topic
1 to 6	Relational Data Model and Use of SQL
7 and 8	Normal Forms and Normalization (for design evaluation)
9 to 14	Entity-Relationship Modeling and database design
15	Database Systems: administration and management

The aim of our project does not dictate the order one must follow to cover the topics. Traditional approach favors beginning with

ER modeling, followed by Relational Model and use of SQL. McMaster (2005) surveyed a number of current database textbooks to compare and contrast different approaches. In conclusion after discussing the pros and cons, they favored their "reverse approach". Our teaching plan presented in this paper follows their approach. One benefit is that we can introduce translation into relational schema in ER modeling to demonstrate its application in database design.

To offer the course on-line for self-paced learning, we need a teaching plan with small learning modules each of which has a set of highly focused objectives. Each module has its own on-line presentation examples, along with a large set of exercises and a discussion forum. This will facilitate for self-paced learning, requiring as well as enticing students to exercise discipline in their own learning. The following section will iterate through the details in each of the modules comprising the teaching plan.

3. THE LEARNING MODULES

ER modeling is an important topic in the course. In the regular pace, students should plan to spend about 7 weeks on the topic. Students may or may not be adept in the use of SQL, but should have been exposed to the relational model and the use of tables in relational DBMS for an introduction.

On the topic of ER Modeling, we have 19 learning modules altogether: starting with Module 0 serving as an introduction to ER model, to Modules 17 and 18 which are optional, covering alternative notations for ER diagrams, namely, the Bachman notation, and the Unified Modeling Language (UML). In the following, we briefly describe each module and note the specific teaching points in each.

Module 0: Introduction to ER Modeling

ER modeling is a tool for the analysis and design of database systems. For analysis, it is a tool to understand and express how business works with data. For design, it becomes a tool to work out the appropriate schema, that is, the way in which a database system can facilitate for the use of data. The introduction uses a simple case study to highlight how the ER diagram captures the schema design of relational tables for a database system.

Module 0: Introduction to ER Modeling	
1	Tool for database system analysis and design
2	Simple case study as an example
3	ER model captures the way business works with data
4	ER diagram expresses the model, ie, the schema.

Module 1: Entities and Attributes

We introduce the basics of ER diagram to present the concept of modeling things as entities, represented by a set of attributes. Students should recognize attribute names and attribute values as data types and data values. Deeper thinking is in identifying an entity to be some specific *thing* while an attribute is a non-specific *value*.

Module 1: Entities and Attributes	
1	Modeling things as entities and their attributes
2	Attribute names and attribute values
3	Entity as something <i>specific</i> , physical or conceptual
4	Entities and entity set

Module 2: Types of Attributes

We present and discuss the different types of attributes, namely, simple and composite attributes, stored and derived attributes, single-valued and multi-valued attributes, and optional attributes, each illustrated with examples.

Module 2: Types of Attributes	
1	Simple and composite attributes
2	Stored and derived attributes
3	Single and multiple valued attributes
4	Optional attributes

Module 3: Key and Key Attributes

Since entities are specific, each entity is to be uniquely identified. The choice of attributes for modeling the entities is relevant. A key is defined as a collection of attributes such that the attribute values in the collection uniquely identify an entity. We then introduce the concepts of key, super key, and candidate key. Students should learn to count and determine the candidate keys.

Module 3: Key and Key Attributes	
1	Uniqueness constraint for each entity
2	Key as a collection of attributes
3	Concepts of key, super key, and candidate key
4	Different candidate keys in the same entity set

Module 4: Tables for Entity Sets

To illustrate that the ER diagram captures the schema, we show how to translate the ER model into tables in the relational data model: one table for each entity set. An attribute generally results in a column in the table, yet some types of attributes may require some special handling. An optional attribute results in a column that allows the null value, for instance. Multi-valued attributes are not desirable, but can be resolved by an additional table.

Module 4: Tables for Entity Sets	
1	Relational table for an entity set
2	Primary key and secondary keys
3	Optional attribute as a column allowing the null value
4	Table for multi-valued attribute

Module 5: The Index Card Analogy

To visualize ER modeling, we introduce the analogy of index cards: one entity per card. The deck of index cards is each printed with the entity set name and attribute names. Each entity, being specific and uniquely identifiable, is represented by an index card with the proper attribute values filled in. It is a model of the entity, physical as well as conceptual.

Module 5: The Index Card Analogy	
1	Entity set as a deck of index cards
2	Entity set name and attribute names are printed
3	A card with the values filled in represents the entity

Figure 1 (next page) illustrates the ER model and the deck of index cards modeling the entity set of courses offered by the IS department.

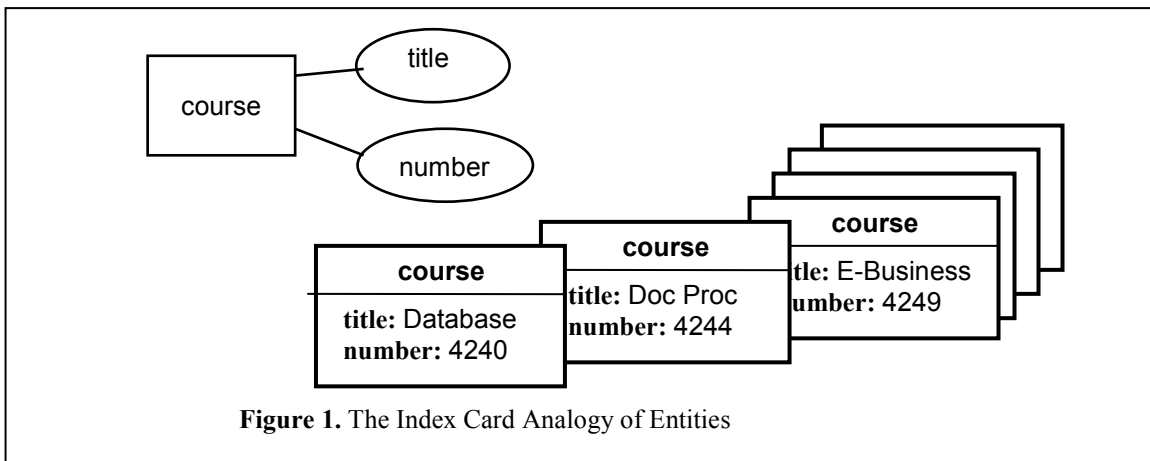


Figure 1. The Index Card Analogy of Entities

Module 6: Relationship and Relationship Instances

Specific entities may relate to one another, similar to relationships in real life. We introduce the diamond to model relationships, each associating two entity sets. We need

to justify the use of only binary relationships in our ER modeling. It will be sufficient for most business use.

We then introduce the notion of relationship instance to identify each case of associating two specific entities in a relationship. Each diamond therefore represents a relationship set, for the entire collection of relationship instances. Relationship thus may have attributes as entity may have attributes, but there will be no key attribute since the relationship instance is uniquely identified when the two associated entities are uniquely identified. We extend the index card analogy to visualize a relationship instance. Figure 2 (next page) illustrates our extended index card analogy to visualize the relationship instances in the relationship set, and how each is linked to a pair of specific entities.

Module 6: Relationship and Relationship Instances	
1	Binary relationship in ER modeling
2	Relationship set, relationship instances
3	Relationship attributes
4	Relationship has no key attributes

Module 7: Participation Constraint

An entity set involved in a relationship may participate partially or totally in the relationship. We introduce the symbolic adornment

for the two respective cases of participation in the ER diagram.

Module 7: Participation Constraint	
1	Total participation
2	Partial participation

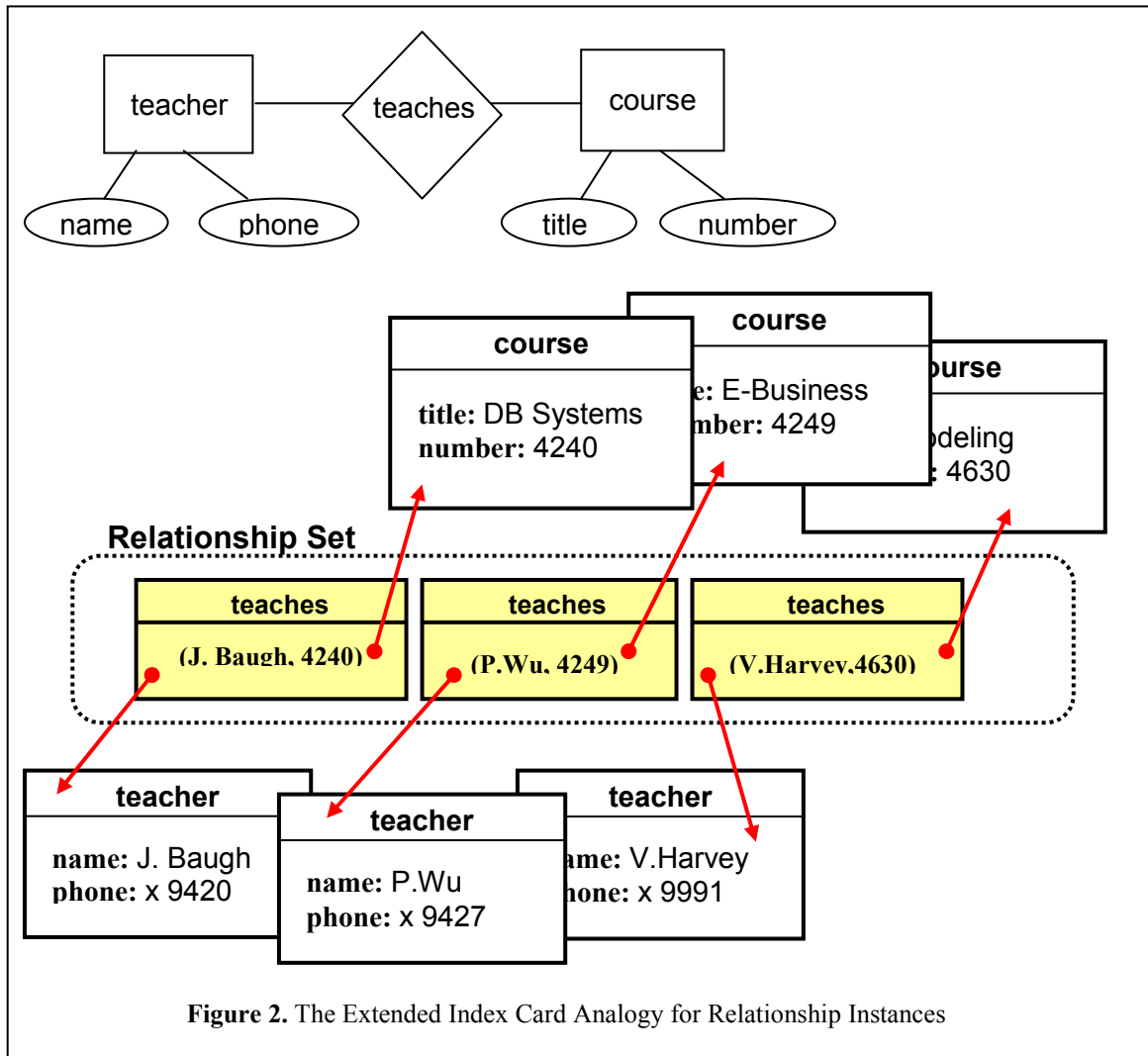
Module 8: Cardinality Constraint

The other constraint adorning the relationship is the cardinality constraint, expressed as the cardinality ratio to indicate how one entity may be associated with one or many entities. The cardinality ratio must be one of these: one-to-one, one-to-many, many-to-one, or many-to-many. The cardinality constraint is independent of, but also works in concert with the participation constraint to model the relationship. Students must learn to properly interpret the constraints.

Module 8: Cardinality Constraint	
1	1-to-1 cardinality ratio
2	1-to-many and many-to-1 cardinality ratios
3	Many-to-many cardinality ratio
4	Cardinality ratio working with participation constraints
5	Properly interpreting cardinality and participation

Module 9: Design Rules and Tips

Students completed with Module 8 should be able to properly interpret ER diagrams in general. ER diagram interpretation is easier than design. Module 9 is intended to help students to construct the ER model in design. The design rules are simple statements pointing to common mistakes students may make. The intricate touch is in sorting out the intention of the designer when there are



mistakes, and to provide appropriate guide to synthesize the proper ER model solution.

Module 9: Design Rules and Tips	
1	Two entity sets can connect only thru a relationship
2	Attribute values should never imply a relationship
3	Relationship cannot connect with another relationship

Module 10: Tables for Relationship Sets

Students now learn to translate the ER model into tables for a relational database, when the ER model involves relationships. Each entity set is translated into a relational table, but a relationship may or may not need to be translated into a table. The cardinality ratio becomes critical. We consider

each case accordingly. We note how the tables, or table extensions, make use of foreign keys, and then because of partial participation, columns in the table extensions may need to allow the null value.

Module 10: Tables for Relationship Sets	
1	Additional table is needed for M-to-M relationship
2	M-to-1 relationship: extend table on the "one" side
3	1-to-1 relationship: can extend either or both tables
4	Use of foreign Keys in these tables
5	Allowing null values for partial participation

Module 11: Weak Entities and Dependence Relationship

A weak entity is also specific, and should still be uniquely identified. Its attributes are not sufficient to identify each entity in the set, and it must be related to another entity for it to be properly identified. The weak entity set must participate totally in a dependence relationship. We introduce the symbols for the weak entity set, and for the dependence relationship. When multiple weak entities are related to the same entity in the dependence relationship for identity, a partial is therefore necessary. The values for the partial key attributes must be sufficient to identify each weak entity among those associated with the same *strong* entity.

Module 11: Weak Entities and Dependence Relationship	
1	Entities dependent on it relationships for identity
2	Symbol for weak entity set / dependence relationship
3	Total participation in the dependence relationship
4	Partial key needed for the M-to-1 relationship

Module 12: Tables for Weak Entity Sets

For a weak entity set in the ER model, the table for a relational database should incorporate the key attributes from the strong entity set it depends on, along with its partial key attributes, to form the primary key. The dependence relationship can be translated in the same way as a relationship.

Module 12: Tables for Weak Entity Sets	
1	Weak entity set translates into table in the same way
2	Combine key from strong entity with partial key, if any
3	Extend the table for dependence relationship

Module 13: Creating Entity Sets in Design

We address a few design issues in creating ER diagram. When it is necessary to see an attribute as an entity, it becomes an attributive entity. When there seems to be a need to relate one relationship to another, we should view the relationship instances as entities, creating an associate entity, which

is usually a weak entity. Quite often, an associate entity is a weak entity with multiple dependence relationships.

Module 13: Creating Entity Sets in Design	
1	Entity versus attribute: an attributive entity
2	Relationship versus entity: an associative weak entity
3	Weak entity with multiple dependence relationships

After completing this module, we expect the students to be capable of properly interpreting ER diagrams.

Module 14: Specialization and Generalization

This module introduces extended ER modeling, covering superclass and subclasses for entity sets. Specialization of an entity set forms a subclass – a subset of the entity set. The subclass inherits the attributes and relationships from its superclass. Specialization of the superclass entity set may be total or partial; when there are multiple subclasses, they may be overlap or disjoint. The ER diagram is extended with new symbols. Generalization is understood to be simply specialization in reverse.

Module 14: Specialization and Generalization	
1	Specialization and subset of an entity set
2	Superclass and subclasses with inheritance
3	Total and partial specialization
4	Disjoint and overlap specialization
5	Generalization as the reverse of specialization

Module 15: Tables for Extended ER Model

For the extended ER model, the superclass is translated to a table in the relational database just like any entity set. For a subclass, the table should have the primary key from the key attributes of the superclass, and the primary key also serves as a foreign key to the superclass table. For total and disjoint specialization, we may not need the superclass table if have the attributes as columns in all its subclass tables, but the relational database system may have to support foreign key referencing multiple tables.

Module 15: Tables for Extended ER Model	
1	Table for superclass: same for an entity set
2	Table for subclass: primary key from superclass table
3	Note foreign key in subclass tables
4	Special case in total and disjoint specialization

Module 16: (min,max) notation for Structural Constraints

We introduce the use of the (min,max) notation to adorn the ER diagram with structural constraints. Students should learn how to interpret the (min,max) notation to understand the ER model in terms of its structural constraints – that is, the participation constraint and the cardinality ratio.

Module 16: (min,max) Structural Constraints	
1	Definition for (min,max) notation
2	Interpreting (min,max) into structural constraints

Module 17: Bachman Notation

This is an optional module to cover use of the Bachman notation for ER diagrams.

Module 17: Bachman Notation	
1	Entity sets – attributes listed inside (or pop-up)
2	A line connecting entity sets indicates relationship
3	Adornment for the structural constraints
4	Notes on associative and attributive entities

Module 18: ER Diagram using UML

This is an optional module to cover a subset of UML for ER model.

Module 18: ER Diagram using UML	
1	Basic UML: classes and objects
2	Public, private, and protected attributes
3	Connection and multiplicities
3	ER model in UML

4. MODULE DEPENDENCY GRAPH

The teaching modules have prerequisite relationships between them. While they generally progress in sequential order, there are

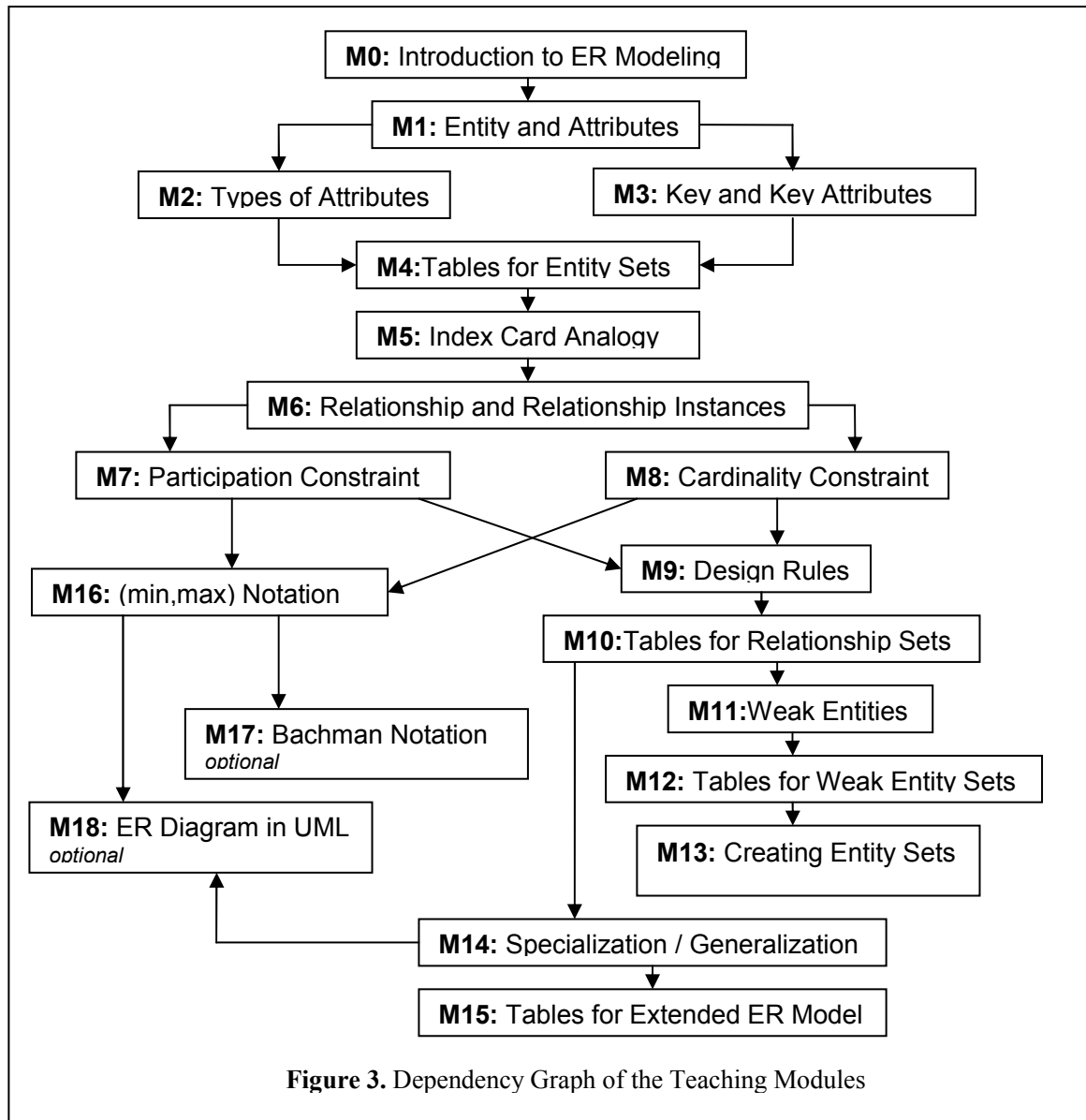
two optional modules as well as other choices in the required sequence to cover them. Figure 3 in the following page depicts the dependency graph to illustrate the prerequisite relationships between the teaching modules.

5. FURTHER DISCUSSION

To provide the flexibility for self-paced learning when teaching a course on-line, we believe the teaching plan should consist of small teaching modules of highly focused objectives. In our previous paper (Wu, Baugh and Harvey 2005), we focused on teaching SQL in the context of an undergraduate Database Systems course. Here we have chosen to work on teaching Entity-Relationship modeling. It is an important topic in the Database Systems course, but it is also an essential tool for information system analysis and design, a skill highly sought after among IS professionals, and thus also a highly desirable topic in IS education (Caputo et al, 2004).

Students engaged in self-paced learning need discipline to make progress in their learning. A good teaching plan must help to entice the students to the next step following each in the learning process. Our small and focused teaching modules are designed with that in mind. Each module presents the student with only a few clear objectives. Therefore, the examples used must also be tactfully designed to avoid unnecessarily complicating the issues involved. Each module consists of several illustrative examples focused on the learning objectives, followed by many practice problems. An on-line discussion forum for each module allows students to raise questions, seek help, or make comments.

Good examples and relevant practice problems are very difficult to come by. They have to make use of the features covered in each module but not require the knowledge of more advanced ones. We are in the process of gathering more and more of these examples and practice problems. An on-going project is to develop a web-based system for sharing and gathering. Our goal is to eventually offer many on-line testing modules for each teaching module to allow students to declare their progress in a self-paced manner, in a complete on-line system offering the course.



6. CONCLUSION

We presented a detailed teaching plan to cover Entity-Relationship modeling in the context of an undergraduate course in Database Systems, a core course in the IS model curriculum (IS2002 2002). The plan is designed to teach the course on-line for self-paced learning. We listed the topics under the objectives in the syllabus to outline the scope of course coverage. The teaching plan comprises 19 small learning modules organized in a dependency graph following the prerequisite requirements of each module.

We presented each teaching module in some details, and then the dependency graph. Each of the teaching modules has a set of very specific learning objectives. We have carefully chosen the learning objectives so that the teaching plan is complete in content coverage, but more importantly it facilitates for teaching on-line to provide great flexibility in self-paced learning.

In each module, the students may review illustrative examples and work on exercise problems. We also provide an on-line forum for discussion and questions. We are also

planning a web-based system to gather a sufficiently large pool of problem sets for on-line testing and examination.

7. REFERENCES

- Barker (1994). A Technological Revolution in Higher Education, *Journal of Educational Technology Systems*, Volume 23, Number 2.
- Baugh (2004). A First Course in Database Management. *Information Systems Education Journal*, 2 (31). <http://isedj.org/2/31/>. ISSN: 1545-679X. (Also appears in *The Proceedings of ISECON 2003*: §3423. ISSN: 1542-7382.)
- Bordoloi and Bock (2004). *Oracle SQL*, Prentice Hall, Upper Saddle River, NJ., 2004.
- Caputo, Kovacs, and Turchek (2004). Defining the Essential Skill and Functional Areas of Study in IT as Measured by a Survey of Field Professionals. *Proceedings of ISECON 2004*, Newport, RI., v.21, SS.2215, October 2004.
- IS2002 (2002). Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems, 2002, curriculum effort of Association for Computing Machinery (ACM), Association for Information Systems (AIS), Association of Information Technology Professionals (AITP).
- Learmonth, Rod, 2001, "Flexible Delivery of Information Systems as a Core MBA Subject" Annual Joint Conference Integrating Technology into Computer Science Education, *Proceedings of the 6th annual conference on Innovation and Technology in computer science education*, Canterbury, United Kingdom.
- Lenox and Woratschek (2005). The Pros and Cons of Using a Comprehensive Final Case Project in a Database Management Systems Course: Marvin's Magnificent Magazine Publishing House. *Information Systems Education Journal*, 3 (24). <http://isedj.org/3/24/>. ISSN: 1545-679X. (Also appears in *The Proceedings of ISECON 2004*: §2425. ISSN: 1542-7382.)
- McMaster, Anderson, and Bilyeu-Dittman (2005). "A Reverse Life-Cycle Database Course with Mini-Projects." *Proceedings of Information Systems Education Conference (ISECON 2005)* Columbus, Ohio, October 2005.
- Naugler and Surendran (2004). Simplicity First: Use of Tools in Undergraduate Computer Science and Information Systems Teaching. *Information Systems Education Journal*, 2 (5). <http://isedj.org/2/5/>. ISSN: 1545-679X. (Also appears in *The Proceedings of ISECON 2003*: §2232. ISSN: 1542-7382.)
- Riccardi, G (2003). *Database Management with Web Site Development Applications*, Addison-Wesley, 2003.
- Robbert, Wang, Guimaraes and Myers (2000). "The Database Course: What Must Be Taught," *SIGCSE Bulletin Proceedings of 31st SIGCSE Technical Symposium on Computer Science Education*, March, pp. 403-404.
- SIGCSE (2001). *CS Body of Knowledge - Information Management Computing Curricula 2001*, Association of Computing Machinery, 2001.
- Springsteel, Robbert, and Riccardo, (2000). "The Next Decade of The Database Course: Three Decades Speak to the Next" *SIGCSE Technical Symposium on Computer Science Education*, March, pp.41-45.
- Wu, Baugh, and Harvey (2005). "Teaching SQL in Database Management Systems for Adult Continuing Education." *Proceedings of Information Systems Education Conference (ISECON 2005)* Columbus, Ohio, October 2005.