2006-1628: REPRESENTING AND ENFORCING BUSINESS RULES IN RELATIONAL DATA MODEL

Reza Sanati, Utah Valley State College

REZA SANATI MEHRIZY is an associate professor of the Computing and Networking Sciences Dept. at Utah Valley State College, Orem, Utah. He received his MS and PhD in Computer Science from University of Oklahoma, Norman, Oklahoma. His research focuses on diverse areas such as: Database Design, Data Structures, Artificial Intelligence, Robotics, and Computer Integrated Manufacturing.

Curtis Welborn, Utah Valley State College

Afsaneh Minaie, Utah Valley State College

AFSANEH MINAIE is an associate professor in the Engineering Department at Utah Valley State College. She received a B.S., M.S. and Ph.D. all in Electrical Engineering from University of Oklahoma in 1981, 1984 and 1989 respectively. Her current interests are in computer architecture, embedded systems, digital design, and computer interfacing.

© American Society for Engineering Education, 2006

Representing and Enforcing Business Rules in Relational Data Model

Abstract

Organizations have many business rules to implement in their daily operations. This is done mainly by action assertions[1] traditionally implemented in procedural logic buried deeply within user's application program in a form that is virtually unrecognizable, unmanageable, and inconsistent. This approach places a heavy burden on the programmer, who must know all the constraints that an action may violate and must include checks for each of these constraints. An omission, misunderstanding, or error by the programmer will likely leave the database in an inconsistent state.

The more modern approach is to define assertions at a conceptual level without specifying how the rule will be implemented. Thus, there needs to be a specification language for business rules. We have seen that the Enhanced Entity Relationship (EER) notation works well for specifying many business rules. In fact, EER notation was invented to allow more business rules to be shown in graphical form than was allowed with the simpler ER notation.

In this paper, we use the ER/EER notation to represent business rules graphically. These rules will be used to enforce database consistency. Using the ER/EER notation, we represented the rules at conceptual level in relational data model without specifying how the rule will be implemented.

Introduction

By applying a business rule, it is intended to assert business structure, or to control or influence the behavior and daily operation of the business.[2] Organizations have many business rules to implement in their daily operations. Traditionally, this is done mainly by action assertions implemented in user's application programs in a form that is not clearly recognizable, manageable, and consistent. This approach places a heavy burden on the programmer to know all the constraints that an action may violate, to implement them carefully, and to include a check for each of these constraints. This is not a reliable approach because an omission, misunderstanding, or error by the programmer will likely leave the database in an inconsistent state.

The more modern and more reliable approach is to define assertions at a conceptual level without specifying how the rules will be implemented. The aim of this approach is to build the constraints into the system to reduce the chance of programming errors. Thus, there needs to be a specification language for business rules. We have seen the Enhanced Entity Relationship (EER) notation works well for specifying many business rules. In fact, EER notation was invented to allow more business rules to be shown in graphical form than was allowed with the simpler ER notation.[3] Associating business rules with the data to which they apply has a natural appeal because the rules are all about the data.[4,5]

In this paper, we use the ER/EER notation to represent business rules graphically. These rules will be used to enforce database consistency for the type of services a mechanic can provide for an airplane. Using the ER/EER notation places the rule at a conceptual level in a relational data model without specifying how the rule will be implemented. This process of capturing business rules starts with a simple example that will be improved gradually through the paper.

Capturing Business Rules

Figure 1 is an Entity Relationship Diagram that depicts the following information about airplane mechanics. A mechanic is an individual with skills that allows him to maintain airplanes. A mechanic must receive specific types of training related to maintaining airplanes. There are many different types of training that a mechanic can receive for maintaining airplanes, such as training on landing gear, training on engines, training on electronics, and so on. In turn, the types of training that a mechanic receives are used to determine the types of maintenance services that the mechanic can perform on an airplane. A specific maintenance service may require that a mechanic receive more than one type of training. Yet, a specific type of training may be useful in providing more than a single maintenance service.

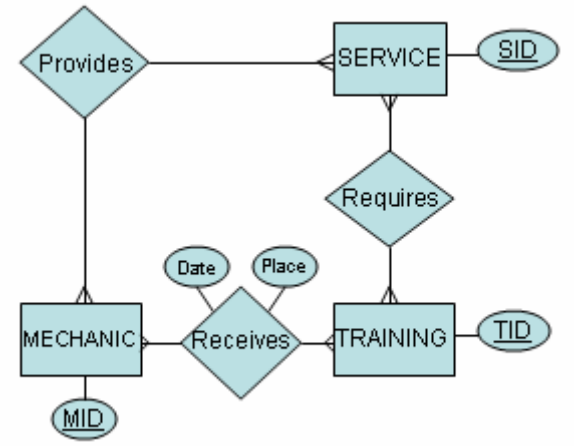

Figure 1: Mechanics

While the entities (Mechanic, Training, Service) and relationships (Received, Requires, Service_Provided) of Figure 1 depict the information discussed about mechanics, training and services, there is an implied constraint upon this information that is not depicted in Figure 1. The constraint could be stated as "a mechanic can only provide a maintenance service if he has received all the training required for that service." The ER model presented in Figure 1 indicates that a mechanic may receive many different types of training and provide many different services. It does not guarantee that the mechanic will receive the required type of training for the type of service that he provides.

To make sure that the mechanic has received the required type of training for the type of service he provides, we must make sure that for each row that appears in the Provides table with the attributes MIDi and SIDj, there is a row in the Receives table with the same attributes MIDi for all TIDs required by the service identified by SIDj. The modified ER diagram represented in

Figure 2 shows the same Entity Relationship Diagram as Figure 1 with the addition of this constraint. It enforces this constraint (business rule) explicitly. The arrows mean the relationship "Provides" enforces the relationships "Receives" and "Requires."[3]

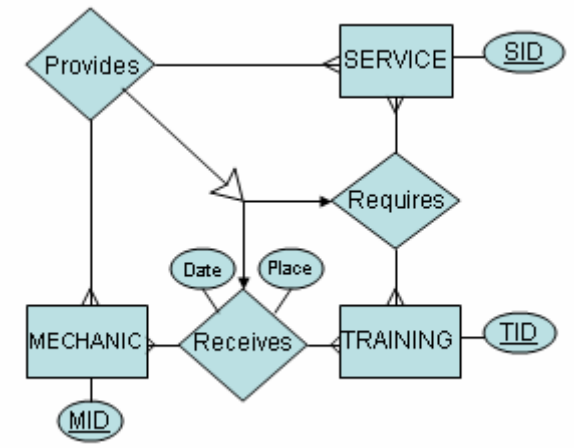

Figure 2: Mechanics + Training Constraint

Figure 3 is the schema for the ER diagram represented in Figure 2.

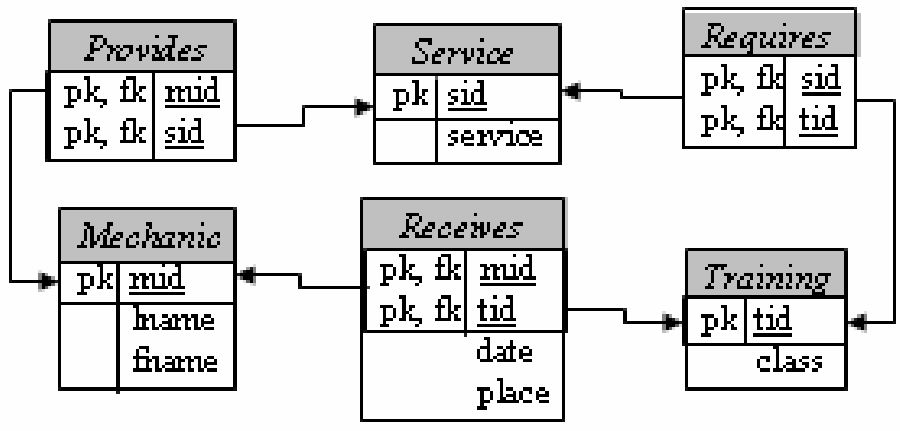

Figure 3: Schema

The following tables show some entities for the tables created from Figure 2 with the addition of some new attributes for Service, Mechanic and Training.

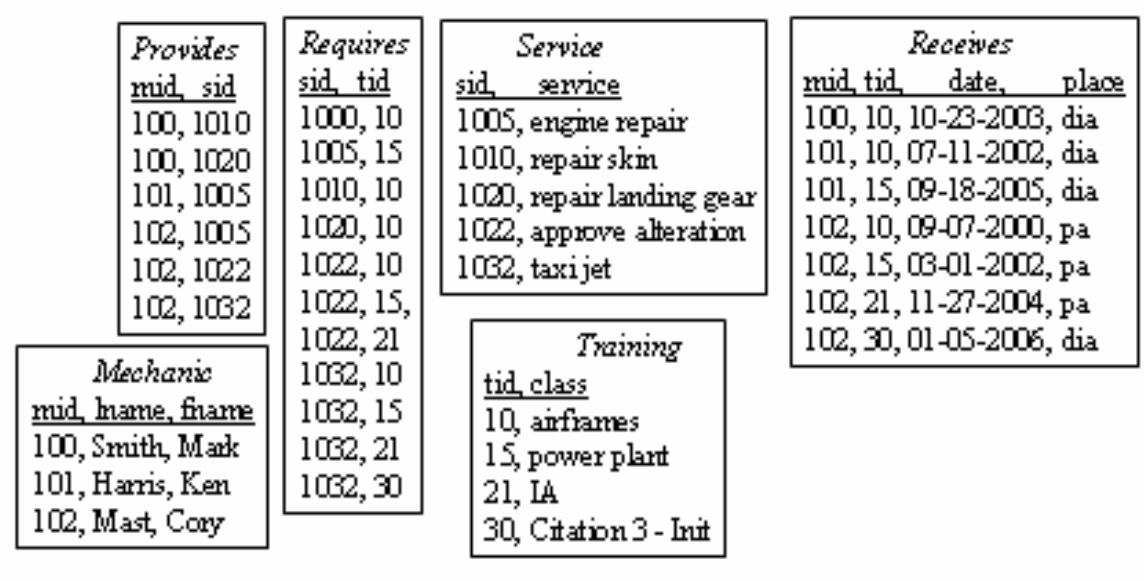

Figure 4: Sample Data

Continuing on with our mechanic example, we learn that one or more specific tools are often required to perform a maintenance service. Figure 5 depicts the mechanic example with the addition of TOOL entity type.

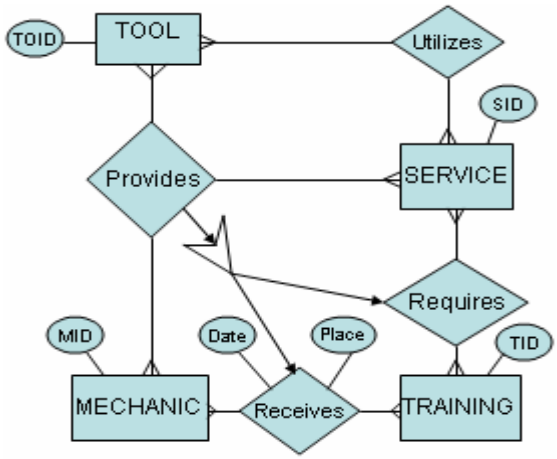

Figure 5: Mechanic + Tools

With the addition of TOOL entity type, our original constraint can now be expanded. While the example still has the constraint that "a mechanic can only provide a maintenance service if he has received all the training required for that service," in addition the constraint should include that "a maintenance service must be done using required tools." This requirement (constraint) can be enforced as in Figure 6.

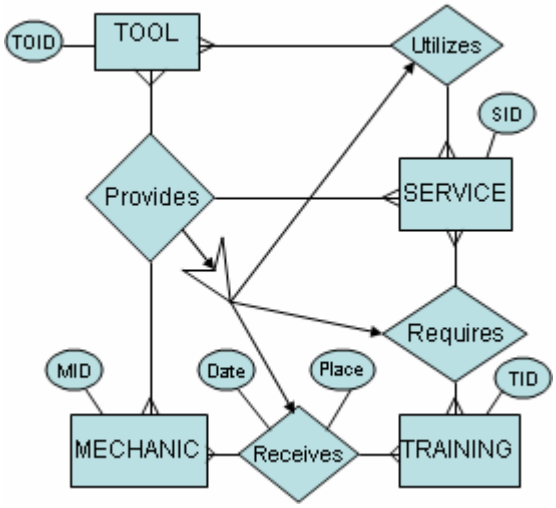

Figure 6: Mechanic + Tool Constraint

By looking at our example a little more, we realize we do not want mechanics and tools just to exist. They should be associated with airplane hangers where the maintenance work is provided. An individual hanger can have more than one mechanic working in it, and a hanger would have many tools. Within this example we will limit a tool to being in only one hanger, while a mechanic can work in more than a single hanger. Adding these entities and the relationships to our example results in Figure 7.

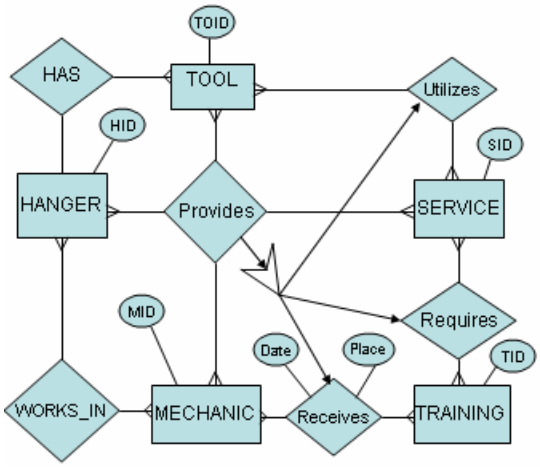

Figure 7: Mechanic + Hanger

Our constraint can now be expanded to include the fact that "a maintenance service is only provided in a hanger." The complete constraint would be that "a hanger can only provide a maintenance service if there is a mechanic that works in the hanger that has received all the training required for that service and the hanger has all the tools needed and the mechanic is using the required tool for that service." The new constraint is show in Figure 8.

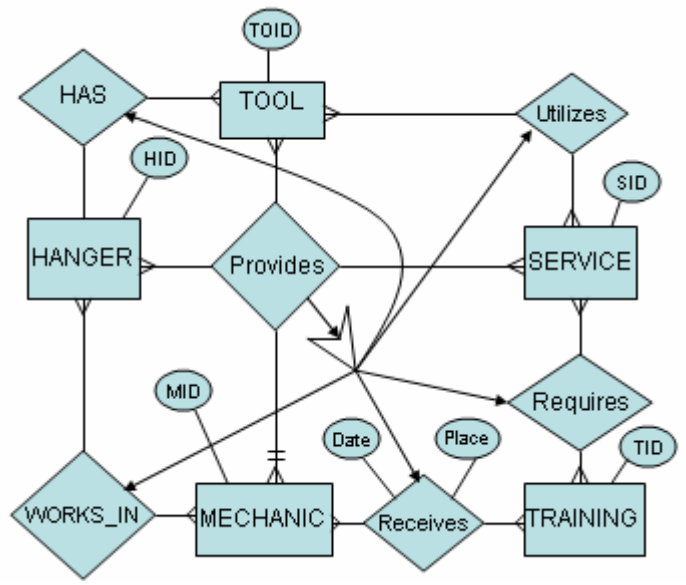

Figure 8: Mechanic + Hanger Constraint

A constraint now can be expressed as: "a service can only be provided by a hanger if there are two mechanics that can provide the same service." The justification for the additional constraint is that one mechanic performs the maintenance service while the other mechanic reviews the work that is performed. This additional condition is represented in Figure 8 by the cardinality symbol ( ) next to the MECHANIC entity type.

Conclusion

In this paper, we use the ER/EER notation to conceptually represent business rules in a relational data model without specifying how the rule will be implemented. The constraints that have been represented in this paper have so far focused on the relationships between entities and could be characterized as existence constraints. Yet, business rules are not limited to only these types of constraints. The future of this work will be to examine other types of constraints to ensure that there is a method of representing and enforcing the other type of constraints. Constraints we are currently researching involve one or more of the following constraint characteristics: cardinality constraints, attribute value constraints, polymorphic constraints and temporal constraints.

References

1. The Business Rules Group, "Defining Business Rules – What Are They Really?", February, 2006, http:www.BusinessRulesGroup.org
2. Perkins, "Business Rules = Meta Data", The proceedings of the: Technology of Object-Oriented Languages and Systems, IEEE, 2000.
3. J. A. Hoffer, M. B. Prescott and F. R. McFadden, "Modern Database Management", Seventh Edition, Prentice Hall, 2005.
4. G. Ronald Ross, "Business Rule Concepts", Business Rule Solutions Inc., 1998.

5. B. von Halle, “Building a Business Rule System, Part 1”, Data Management Review, Faulkner & Gray, January 2001.