INTRODUCTION

Though informal, the concept of business rule is very important to the modeling and definition of information systems. Business rules are used to express many different aspects of the representation, manipulation and processing of data (Paton, 1999). However, perhaps due to its informal nature, business rules have been the subject of a limited body of research in academia. There is little agreement on the exact definition of business rule, on how to capture business rules in requirements specification (the most common conceptual models, entity-relationship and UML, have no proviso for capturing business rules), and, if captured at all, on how to express rules in database systems. Usually, business rules are implemented as triggers in relational databases. However, the concept of business rule is more versatile and may require the use of other tools.

In this article, I give an intuitive definition of business rule and discuss several classifications proposed in the literature. I then argue that there is no adequate framework to capture and implement business rules and show some of the issues involved, the options available, and propose a methodology to help clarify the process. This article assumes basic knowledge of the relational data model (Date, 2000).

BACKGROUND

The term business rule is an informal, intuitive one. It is used with different (but overlapping) meanings in different contexts. GUIDE (2000) defined business rule as follows: “A business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business.” Thus, the rule expresses a policy (explicitly stated or implicitly followed) about some aspect of how the organization carries out its tasks. This definition is very open and includes things outside the realm of information systems (e.g., a policy on smoking). I will not deal with this kind of rule here; instead, I will focus only on business rules that deal with information. Intuitively, a business rule in this sense usually specifies issues concerning some particular data item (e.g., what fact it is supposed to express, the way it is handled, its relationship to other data items). In any case, the business rule represents information about the real world, and that information makes sense from a business point of view. That is, people are interested in “the set of rules that determine how a business operates” (GUIDE, 2000). Note that while data models give the structure of data, business rules are sometimes used to tell how the data can and should be used. That is, the data model tends to be static, and rules tend to be dynamic.

Business rules are very versatile; they allow expression of many different types of actions to be taken. Because some types of actions are very frequent and make sense in many environments, they are routinely expressed in rules. Among them are:

- enforcement of policies,
- checks of constraints (on data values, on effect of actions), and
- data manipulation.

Note: When used to manipulate data, rules can be used for several purposes: (a) standardizing (transforming the implementation of a domain to a standard, predefined layout); (b) specifying how to identify or delete duplicates (especially for complex domains); (c) specifying how to make complex objects behave when parts are created–deleted–updated; (d) specifying the lifespan of data (rules can help discard old data automatically when certain conditions are met); and (e) consolidating data (putting together data from different sources to create a unified, global, and consistent view of the data).

- time-dependent rules (i.e., rules that capture the lifespan, rate of change, freshness requirements and expiration of data).

Clearly, business rules are a powerful concept that should be used when designing an information system.

Types of Business Rules

Von Halle and Fleming (1989) analyzed business rules in the framework of relational database design. They distinguished three types of rules:
1. **Domain rules:** Rules that constrain the values that an attribute may take. Note that domain rules, although seemingly trivial, are extremely important. In particular, domain rules allow

- verification that values for an attribute make “business sense”;
- decision about whether two occurrences of the same value in different attributes denote the same real-world value;
- decision about whether comparison of two values makes sense for joins and other operations.

For instance, EMPLOYEE.AGE and PAYROLL.CLERK-NUMBER may both be integer but they are unrelated. However, EMPLOYEE.HIRE-DATE and PAYROLL.DATE-PAID are both dates and can be compared to make sure employees do not receive checks prior to their hire date.

2. **Integrity rules:** Rules that bind together the existence or nonexistence of values in attributes. This comes down to the normal referential integrity of the relational model and specification of primary keys and foreign keys.

3. **Triggering operations:** Rules that govern the effects of insert, delete, update operations in different attributes or different entities.

This division is tightly tied to the relational data model, which limits its utility in a general setting. Another, more general way to classify business rules is the following, from GUIDE (2000):

1. **Structural assertions:** Something relevant to the business exists or is in relation to something else. Usually expressed by terms in a language, they refer to facts (things that are known, or how known things relate to each other). A term is a phrase that references a specific business concept. Business terms are independent of representation; however, the same term may have different meanings in different contexts (e.g., different industries, organizations, line of business). Facts, on the other hand, state relationships between concepts or attributes (properties) of a concept (attribute facts), specify that an object falls within the scope of another concept (generalization facts), or describe an interaction between concepts (participation facts). Facts should be built from the available terms. There are two main types of structural assertions:

   a. **Definition of business terms:** The terms appearing on a rule are common terms and business terms. The most important difference between the two is that a business term can be defined using facts and other terms, and the common term can be considered understood. Hence, a business rule captures the business term’s specific meaning of a context.

   b. **Facts relating terms to each other:** Facts are relationships among terms; some of the relationships among terms expressed are (a) being an attribute of, (b) being a subterm of, (c) being an aggregation of, (d) playing the role of, or (e) a having a semantic relationship. Terms may appear in more than one fact; they play a role in each fact.

2. **Constraints (also called action assertions):** Constraints are used to express dynamic aspects. Because they impose constraints, they usually involve directives such as “must (not)” or “should (not).” An action assertion has an anchor object (which may be any business rule), of which it is a property. They may express conditions, integrity constraints, or authorizations.

3. **Derivations:** Derivations express procedures to compute or infer new facts from known facts. They can express either a mathematical calculation or a logical derivation.

There are other classifications of business rules (Ross, 1994), but the ones shown give a good idea of the general flavor of such classifications.

**Business Rules and Data Models**

There are two main problems in capturing and expressing business rules in database systems. First, most conceptual models do not have means to capture many of them. Second, even if they are somehow captured and given to a database designer, most data models are simply not able to handle business rules. Both problems are discussed in the following sections. As a background, I discuss how data models could represent the information in business rules. Focusing on the relational model, it is clear that domain constraints and referential integrity constraints can be expressed in the model. However, other rules (expressing actions, etc.) are highly problematic. The tools that the relational model offers for the task are checks, assertions, and triggers.
Related Content

Set Comparison Queries in SQL
[www.igi-global.com/chapter/set-comparison-queries-sql/8281?camid=4v1a](www.igi-global.com/chapter/set-comparison-queries-sql/8281?camid=4v1a)

Process-Embedded Data Integrity
[www.igi-global.com/article/process-embedded-data-integrity/3307?camid=4v1a](www.igi-global.com/article/process-embedded-data-integrity/3307?camid=4v1a)

Object-Oriented Database Design
[www.igi-global.com/chapter/object-oriented-database-design/8277?camid=4v1a](www.igi-global.com/chapter/object-oriented-database-design/8277?camid=4v1a)

Hypermedia Document Management: A Metadata and Meta-Information System
[www.igi-global.com/article/hypermedia-document-management/3262?camid=4v1a](www.igi-global.com/article/hypermedia-document-management/3262?camid=4v1a)