Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Electronic Commerce Research and Applications 3 (2004) 54–73

Electronic
Commerce Research
and Applications

www.elsevier.com/locate/ecra

# Structuring business rules interactions

Veerendra Kumar Rai *, C. Anantaram

*Tata Consultancy Services, 54B, Hadapsar Industrial Estate, Pune 411 013, India*

Received 24 January 2003; received in revised form 15 September 2003; accepted 19 September 2003

**Abstract**

Motivation for arranging business rules is not confined to efficient processing of rules, as in case of traditional rule bases, but to aid in decision making as well. Efficient processing and ease of decision making, therefore, are the twin objectives behind arranging and grouping of business rules as envisaged by business rules approach to system development. This paper strives to make a beginning to realize these objectives. Six types of rule groupings have been proposed in this paper. Examples from industrial domain have been taken to elucidate the rule groupings.


*Keywords:* Business rules; Business policies; Grouping of rules; Rules interaction; Rule sets

## 1. Introduction

Business rules approach to system development may not be a paradigm shift but it definitely undertakes a noticeable departure from the way information systems were developed traditionally. In traditional information system development writing software meant writing programs and generating code. Everything was put into code and a software system meant a body of code. In such a scenario a system was accessible and visible thorough its code only and in no other way. This approach overlooked the fact that a business decision-maker may not and need not be a software professional. Business rules could pop up almost at every business decision making process. Which customer should be treated as privileged customer, on what basis an insurance policy should be renewed, how to reimburse an insurance claim are some of the examples where business rules are needed to take decision. In the absence of any overt articulation of business rules, a decision-maker has to look into the code to know them. You know there is a problem if you have to 'look at the code', for there is no explicit statement about what business rules are [18]. If business rules are imbedded into the code, then the only way to get them out is by mining business rules through a kind of reverse engineering activity on software [16]. Mining business rules through reverse engineering has two problems. One, business rules might be scattered throughout the entire code and it may not be effective, in terms of cost and effort, to undertake this exercise. Two, reverse engineer-

* Corresponding author. Tel.: +91-20-687-1058; fax: +91-20-681-0921.
*E-mail addresses:* vkrai@pune.tcs.co.in (V.K. Rai), anant@pune.tcs.co.in (C. Anantaram).

1567-4223/$ - see front matter 
doi:10.1016/j.elerap.2003.09.007

ing for business rules may be based on the assumption that code was developed with business rules in mind. This assumption may not be tenable. Whether or not business rules may be viewed as expressing functional as well non-functional requirements could be debatable and much depends upon how one defines business rules. Requirements captured by business rules are, in principle, no different from other kind of requirements. However, business rules are characterized by their strategic importance to the business and deserve special attention [18]. If business rules are intended to facilitate business decision making, then there must be mechanisms to make rules accessible to business decision-makers. This further requires that business rules are expressed in terms of an enterprise level model and not in terms of programs and databases [5,13].

Business rules have received a lot of attention in the trade press and other literature as holding answers to many information technology problems. Sandifer and Halle [19] have even suggested that one can gather a list of English language statements describing the business rules informally as a first step towards taking a business rules approach. Other approaches are more structured, relying on the syntax of E–R diagram. Roland [17] has proposed a much more elaborate diagrammatic syntax. Still others propose a conceptual model approach for describing an enterprise with a notation for specifying rules that further specify the requirement on the enterprise [5,13]. Martin and Odell [14] limit the enterprise model to data and object rules. Some researchers treat non-functional requirements as goals that have to be met through a decision making process in which change is expected. Lamsveerde et al. [10] propose a goal directed requirements elaboration methodology which attempts to cope with the ''deidelaization of unachievable goals'' and also models assumptions attached to goals. Fickas and Feather [9] discuss requirement monitoring to instrument the running system to determine whether and to what extent requirements are being met by the system. Eventual aim of taking business rules approach to information system development is to automate the business processing through business rules. It is, in other words, aiming at compilable and executable specifications [7].

Not much literature is available with respect to arranging business rules with the purpose of ease of decision making and efficiency of rules processing. Rosca et al. [18] discuss three levels of business rules. Criteria level, argument level and assumption level and describe them as follows:

The rules at criteria level are the highest level rules and they are the most general types of rules. These rules relate to high level decision-making and code criteria for decisions in IF and THEN form. Rules at argument level express the heuristics used in deciding how well an alternative satisfies the criterion when several arguments are presented for or against a solution. Rules at assumption level are the most detailed of rules where business objectives find their implementation level meaning. This classification of rules by Rosca et al. [18] is very helpful in business rules implementation, however, it does not tell how to model these rules and how to establish a relationship between the rules of the same class or type for the purpose of ease of decision making and for the efficient processing of rules. This paper will take up these issues in some detail.

Anantaram [2] has studied formation of sets of rules from the possible interactions that can occur between rules such that these interactions are meaningful in the specified domain. In this study the following types of interactions between rules have been discussed: Rule chains, rules assertions and rule groups. In rule chain, this study considers intersection of ''rule-attributes'' in the antecedent of one rule with ''rule-attributes'' in the consequent of another rule. A rule chain is said to be created when the antecedent of a rule (say, Rule 1) refers to the consequent of another rule (say Rule 2). In rule assertions, the study examines intersection of ''rule-attributes'' in the consequents (or the then-part) of the rules. Rule groups in this study are defined based on some commonality in the condition (or antecedent).

A similar approach for partitioning rulebase was taken by Brown and Pomykalski [4], where a rule base is partitioned into a set of chains (the horizontal partition) and a set of groups (vertical partitions).