Expressing and organising business rules

P. Kardasis\textsuperscript{a}, P. Loucopoulos\textsuperscript{b,*}

\textsuperscript{a}Deloitte and Touche Consulting, 250-254 Kifissias Ave., Halandri, Athens 152 31, Greece
\textsuperscript{b}Department of Computation, UMIST, P.O. Box 88, Manchester M60 1QD, UK

Received 16 October 2003; revised 10 December 2003; accepted 15 December 2003
Available online 2 March 2004

Abstract

Business rules represent projections of organisations’ constraints and ways of working on their supporting information systems. Therefore, their collection, expression, structuring and organisation should be central activities within information systems analysis. The work presented in this paper concerns the definition of a repository schema for managing business rules, taking into account the objectives (a) to facilitate the rule collection process, (b) to assist the transition from analysis to design and implementation of the information system, and (c) to support business change once the new system has been delivered. These objectives are achieved through the enhancement of the rule repository schema with information on the logistics of the collection process and references to underlying enterprise informational and behavioural knowledge models. The proposed schema and way of working are demonstrated through a number of examples, which are derived from an industrial project concerning electronic procurement in the construction sector.

Keywords: Business rules; Event–condition–action paradigm; Repository; Rule management

1. Introduction

Business rules have been defined as ‘declarations of policy or conditions that must be satisfied’ [1], and their role is to determine how operational decisions within an organisation must be made. In other words, business rules specify action on the occurrence of particular business events, including ‘state of being’ changes concerning individuals and groups of individuals, infrastructure, consumables, informational resources, and even business activities.

The aim of this paper is to discuss an approach to a business rules centric development paradigm, known as Manchester Business Rules Management (MBRM), in the context of a project about e-procurement. MBRM comprises of a number of activities and techniques. The focus of this paper is in the elicitation, organisation and management of business rules. To this end, the paper proposes a repository schema that enables the adoption of a goal-based approach to rule collection. It also associates rule statements with information entities of their universe of discourse, in order to facilitate their management by allowing rules grouping and retrieval according to different criteria. MBRM has its origins in the Enterprise Knowledge Development (EKD) framework [2,3], a development framework that has been used in many large scale industrial applications in banking, electricity deregulation, and e-business [4–8].

The structure of this paper is as follows: Section 2 introduces a rule-intensive IS project, dealing with the development of an electronic procurement system. Examples from this application are used throughout the paper in order to demonstrate the underlying approach. Section 3 discusses methodological issues relating to the collection of business rules and convenient rule classifications that facilitate this process. Section 4 presents the repository schema for structuring and organising business rules in a way that efficient rule management may be facilitated. Section 5 gives details of descriptive and prescriptive business rules for the application introduced in Section 2. Section 6 compares and contrasts other related work. Finally, Section 7 details a number of observations regarding the business rules approach in the context of MBRM and its application to a large commercial application.
2. The example application: designing an electronic procurement system for the construction sector

The case study presented in this section illustrates the application of the proposed business rules approach in a rule-intensive project within the construction sector. The objective of this project was to design an electronic procurement system, which would assist a medium-sized construction company in purchasing raw materials from their suppliers, and in sub-contracting services to other companies. More specifically, the main aims of the project were to enable ‘electronic management of RFPs’ and ‘electronic catalogue procurement’. RFPs are the ‘requests for proposals’ on behalf of sub-contractors and involve the publishing of specifications for sub-contracted services, the evaluation of proposals and the management of contracts. Catalogue procurement concerns the purchasing of raw materials, and includes creation of purchase orders, invoicing, payments and monitoring of order execution.

The above goals are shown diagrammatically in Fig. 1, where goals are presented as rectangles, and goal relationships as triangles, connecting a high-level, abstract goal with lower-level, operational, leaf goals.

Fig. 2 contains a simplified information flow diagram that depicts behavioural elements of the application areas relating to RFP management.

The process of managing RFPs consists of a number of interrelated business activities. Initially, the RFP is prepared according to the needs and requirements of the organisation. The publishing of an RFP is followed by the collection of responses on behalf of service providers (e.g. construction companies). Offers/proposals are evaluated in various stages, taking into account the credibility of the proposer, the quality and suitability of the proposed solution, cost parameters and so on. Successful proposals lead to contracts. A contract contains all the necessary specifications for the work to be carried out: technical characteristics of the solution, methodological approach, structure of the project team, time plan, risk management procedures and so on. Finally, delivered work is evaluated against the aforementioned specifications and payments are executed accordingly.

The corresponding data flow diagram for catalogue procurement has been omitted as no examples from this application area have been given in the following sections of the paper.

3. Collecting and classifying business rules

3.1. Overview of the MBRM Approach

3.1.1. Background to business rules

The term ‘business rule’ has been used by different methodologists in different ways. In Ref. [9], business rules are ‘statements of goals, policies, or constraints on an enterprise’s way of doing business’. In Ref. [10], they are defined as ‘statements about how the business is done, i.e. about guidelines and restrictions with respect to states and processes in an organisation’. Mitchell Krammer [12] considers them as programmatic implementations of the policies and practices of a business organisation. ‘Depending on whom you ask, business rules may encompass some or all relationship verbs, mathematical calculations,'
inference rules, step-by-step instructions, database constraints, business goals and policies, and business definitions’ [13]. In this work, business rules are approached from an information systems analysis perspective, and can be defined as:

Projections of external constraints on an organisation’s way of working, and on its supporting information systems ‘functionality’

3.1.2. The development framework

The MBRM approach covers a number of key information system development stages, all of which are centred on a business rules paradigm. These activities are shown schematically in Fig. 3. Elicitation is concerned with the identification of stakeholders, the domain ontology and the rules that govern the behaviour of the business application [14]. Representation deals with the way that business rules are specified according to their typology. Mapping is concerned with the way that business rules specification is linked to equivalent software design structures. Implementation deals with the way that software designs are realised in software code and database structures.

In a user-driven system evolution environment, it is necessary to allow authorized application users or system administrators to manage business rule changes. The changes include the deletion and modification of existing rules, or addition of new rules according to the changes in business environments. In the context of a software maintainer, the system should provide business rule traceability to system components. Therefore, the MBRM approach aims to allow business rule management and traceability through a repository architecture.

Business rules are captured and stored in structured forms or templates suitable for linking to software design components. In addition, each rule is associated with rule management information such as business process and process owner. Section 4 gives details of the way that business rules are organised and managed in a repository environment.

3.2. Elicitation and analysis of business rules

The MBRM framework, which constitutes the methodological basis of this work, identifies three views for approaching information systems analysis: the intentional

Fig. 2. Simplified information flow diagram for the RFP management process.

Fig. 3. The MBRM framework.
view, the operational view and the information systems view. Along these lines:

‘Intentional analysis’ contains preparatory activities aiming at an initial understanding of the organisation and crystallisation of the project’s scope and objectives.

‘Operational analysis’ involves the development of the necessary enterprise knowledge context, making references to central business process concepts, such as actors, activities, activity enablers and information objects that are used or produced by enterprise activities.

‘IS architecture analysis’ bridges the gap between analysis and design, through the transformation of implementation-free requirements to implementation-specific requirements and specifications.

During each one of the above analysis stages, business rules are treated in different ways. Therefore we distinguish between ‘intentional rules’, ‘operational rules’ and ‘IS architecture rules’.

‘Intentional rules’ are expressions of business rules seen from a business context perspective. They express laws, external regulations, or principles and good practices specifying the way an organisation conducts business and are usually expressed in the form of natural language statements. Their organisation in a repository environment focuses on the representation of their causal relationships (i.e. relationships with existing or future enterprise goals), information about the rules collection process itself (stakeholder that revealed them, version, etc.), and details about their enforcement (date of enforcement, expiry date, status, etc.).

‘Operational rules’ are expressions of business rules approached from a business process perspective. They prescribe action on the occurrence of some business event, or describe valid states of an organisation’s information entities. ‘Operational rules’ derive from the translation of informal ‘intentional rules’ to formal rule statements developed in accordance with a convenient rule language and repository schema, also making reference to other enterprise knowledge concepts (actors, activities, activity enablers, information objects and their attributes). The structure of ‘operational rules’ and their linking with the aforementioned knowledge entities is central to this approach, as they dictate the structure and contents of the rule repository which will support the approach’s adoption in a real business setting.

‘IS architecture rules’ are business rule statements examined from a IS implementation perspective. They are expressed in an implementation-specific manner, i.e. in accordance with the implementation paradigm that has been selected for the information system under development (e.g. through the choice of a specific customisable rule engine, an active database, and so on). Normally, the identification, formation and organisation of ‘IS architecture rules’ is covered by IS development approaches, and therefore, they are beyond the scope of this work.

3.2.1. Classifying intentional rules for supporting intentional rule analysis

Intentional rule analysis involves preparatory activities in order to:

- Identify the actors that play an important role within the organisation under investigation.
- Specify general project goals and boundaries of the application area.
- Determine different stakeholder viewpoints on the mission of the project application areas and the project objectives themselves (namely enterprise and change goals, respectively).
- Discuss aforementioned enterprise and change goals and collect: (a) rules that are enforced by them, (b) rules that hinder their realisation, and (c) rules that create opportunities for their realisation.

In Ref. [15], ‘viewpoints’ are characterised as vehicles for separation of concerns, allowing stakeholders to address only those concerns or criteria that are of interest, ignoring others that are unrelated. Stakeholders with different views are usually aware of different business constraints (in other words, different business rules). These stakeholders need to participate in the requirements analysis phase of the project, by stating their own goals and by discussing how these goals are translated to business rules.

Therefore, the classification of business rules at the stage of ‘intentional analysis’ must be in accordance with the identification of different viewpoints discussed above. The proposed classification is based on the distinction between ‘Domain-Dependent Rules’ and ‘Domain-Independent Rules’, and, in case of ‘Domain-Independent Rules’, on the identification of specific application areas (or business activity groups), which exist within most industry sectors. Extensive consulting experience has shown for example that management, business development, sales and marketing, procurement and administration activities can be found in a number of different markets: insurance, construction, manufacturing, banking, utilities, healthcare, real estate, retail, telecommunications, and so on. An empirical classification of rules along these lines is shown in Fig. 4, although it is certain that additional types of ‘Domain-Independent Rules’ could be identified.

‘Management rules’ are expressions of business policy or constraints that deal with managerial issues, such as strategy development, tactical planning, business performance management, financial management, investments management, project management, and risk management (quality assurance and issues management).

‘Business development rules’ deal with feasibility exploration concerning a new product or service idea, new product or service design, an existing product’s or service’s enhancement, trial/testing of a product or service, costing and pricing.
Sales and marketing rules relate with market analysis, trends forecasting, product or service marketing, product or service withdrawal, public relationships management, customer relationships management, tendering, personalisation of offerings, purchase/service order management, customer account management and customer training.

Core business rules constitute the wider rule category and depend on the particular characteristics of different industries. Some examples are: product manufacturing rules, distribution rules, returns management rules, store operation rules, operation scheduling rules, billing and revenues collection rules, benefits and claims management rules, service provision rules and service provision network management rules.

Procurement rules concern purchasing of materials or sub-contracting of business services. Relevant activities are supplier/service provider selection, supplier relationships management, material procurement, service procurement and payables management.

The term ‘administration rule’ refers to supporting business activity policies and constraints, and can represent asset management rules, facilities construction and maintenance rules, inventory control rules, IT management rules, legal issues management rules, document management rules, accounting rules and human resources management rules.

Finally, there are a wide variety of rules representing the obligations of an organisation towards the State and the society. These rules relate to: economy, taxation and market, culture and education, employment and equality, environment and energy, health and safety, security and justice, and privacy, data and intellectual property protection.

3.2.2. Classifying operational rules for supporting operational rule analysis

The knowledge modelling ‘exercise’ described in this work mainly aims at the design of business supporting information systems, which will either be built from scratch, or may be based on the configuration of a packaged IT solution. An initial step in information systems design deals with ‘architecturing’, which concerns the identification of its distinct (although usually integrated) operational parts. Each one of these parts has a specific objective (‘goal’). This objective is fulfilled through the facilitation of a specific workflow (‘activities’). A systemic workflow involves one or more types of users (‘actors’), uses or produces information (‘information objects’), and also engages other enterprise resources (‘activity enablers’). Operational rule analysis is based on this rationale, as shown in Fig. 5.

The term ‘context’ in the above figure represents the boundaries of the (business or system) application under investigation, and is characterised by the aforementioned enterprise knowledge entities: ‘actors’, ‘activities’, ‘information objects’ and ‘activity enablers’. Operational rules describe structural and behavioural characteristics of the above enterprise knowledge entities. More specifically, ‘operational rules’ belong in two major groups: ‘descriptive rules’ and ‘prescriptive rules’ (Fig. 6). ‘Descriptive rules’ describe valid states of an organisation’s informational

---

Fig. 5. Operational rule analysis steps.
entities, and ‘prescriptive rules’ prescribe action on the occurrence of some business event. This approach is also followed by several existing methodologies, which distinguish between structural and action assertions (elsewhere referred to as integrity constraints and automation rules, or even data and process rules) [9,10,16].

‘Descriptive rules’ can be further specialised in ‘information object rules’, ‘actor rules’, ‘activity rules’ and ‘activity enabler rules’. ‘Information object rules’ examine the attributes of ‘information objects’. ‘Information objects’ represent entities containing information that is useful for an organisation. A service proposal, a product purchase order, or even a product are all represented by the corresponding ‘information objects’. ‘Information object rules’ specify how an ‘information object’ attribute derives from other ‘information object’ attributes (‘inf. object attribute derivation rules’), or define ‘information object’ sets based on their attribute values’ similarity (‘inf. object set rules’), or examine integrity (‘inf. object integrity rules’). ‘Actor rules’ are very similar, dealing with ‘actor’ attributes (i.e. attributes of employees, partners, customers and so on). ‘Activity rules’ describe status of activities. And ‘activity enabler rules’ are concerned with the status of ‘activity enablers’, i.e. the infrastructure that allows the execution of activities.

‘Prescriptive rules’ are distinguished into ‘workflow rules’ and ‘information assertion rules’. ‘Workflow rules’ examine a set of conditions on the occurrence of some event, and determine what workflow actions need to be taken. ‘Information assertion rules’ examine a set of conditions on the occurrence of some event, and perform information updates accordingly: creation or deletion of information entities (‘information operation rules’), assertion of information entity attribute values (‘attribute assertion rules’), or assertion of information entities association (‘association assertion rules’).

Based on the above, ‘descriptive’ and ‘prescriptive’ rules can have one of the lowest level types shown on Table 1.

### Table 1

<table>
<thead>
<tr>
<th>最低水平类型的业务规则</th>
<th>描述性规则</th>
<th>作用规则</th>
<th>信息对象规则</th>
<th>活动使能规则</th>
<th>轮廓规则</th>
</tr>
</thead>
<tbody>
<tr>
<td>主要</td>
<td>角色规则</td>
<td>行动规则</td>
<td>信息对象规则</td>
<td>活动使能规则</td>
<td>工作流规则</td>
</tr>
<tr>
<td>角色规则</td>
<td>角色属性</td>
<td>行动属性</td>
<td>信息对象属性</td>
<td>活动使能属性</td>
<td>工作流调用规则</td>
</tr>
<tr>
<td>角色规则</td>
<td>角色集规则</td>
<td>行动集规则</td>
<td>信息对象集规则</td>
<td>活动使能集规则</td>
<td>工作流规则</td>
</tr>
<tr>
<td>角色规则</td>
<td>角色完整性规则</td>
<td>行动完整性规则</td>
<td>信息对象完整性规则</td>
<td>活动使能完整性规则</td>
<td></td>
</tr>
</tbody>
</table>

4. Organising business rules in a repository environment

4.1. Overview

The schema of Fig. 7 presents the relations between various enterprise knowledge entities, which are central to the ontological analysis of business rules. Such entities are ‘goal’, ‘role’, ‘activity’, ‘event’, ‘activity enabler’ and ‘information object’ [2,6,17] (Kardasis and Loucopoulos, 1998).

Within a particular enterprise context, ‘actors’ are involved in the realisation of different business ‘goals’, through the execution of enterprise ‘activities’. ‘Activities’, which are grouped in enterprise ‘roles’, are triggered by business ‘events’, and make use of ‘activity enablers’, i.e. supporting material infrastructure. They also consume or produce information, which is ‘packaged’ in ‘information objects’. All the information entities mentioned above have

---

3 ‘Ontology’ is the branch of philosophy concerned with the study of what exists [35]. Or, according to Ref. [34], ‘ontology’ is the study of assumptions related to a certain phenomenon under investigation.
attributes. Therefore there are ‘actor attributes’, ‘activity attributes’, ‘activity enabler attributes’ and ‘information object attributes’.

Relationships that are worth mentioning within the business rules ontology context concern the facts that:

- ‘Goals’ are satisfied by ‘Actors’
- ‘Activities’ are part of ‘Roles’
- ‘Activities’ are triggered by ‘Events’
- ‘Activities’ are played by ‘Actors’ (‘Actor/Activity Relationship’)
- ‘Activities’ use ‘Activity Enablers’ (‘Activity/Activity Enabler Relationship’)
- ‘Activities’ produce/consume ‘Information Objects’ (‘Activity/Information Object relationship’)

The schema of Fig. 8 reflects the proposed rules classification, according to which rules can either be ‘intentional’, ‘operational’ or ‘IS architecture rules’. Relationships ‘Intentional Rule leads to Operational Rule’ and ‘Operational Rule leads to IS Architecture Rule’ indicate the methodological transition from one type of rule to another, which has already been discussed. From a repository schema point of view, this relationship facilitates tracking of rules rationale.

The purpose of all the relationships connecting ‘business rules’ to ‘goals’ is similar, in the sense that they facilitate tracking of the rules history and rationale. More specifically, a ‘goal’ may enforce a ‘business rule’, or a ‘business rule’ may hinder the realisation of a ‘goal’, or a ‘business rule’ may create an opportunity for the realisation of a ‘business rule’.

Fig. 7. Actors, activities, activity enablers and information objects and their interrelations.

Fig. 8. Relationships between rules and goals.
A ‘business rule’ is accompanied by a number of information attributes mainly dealing with its enforcement within a particular business context:

- ‘Rule type’: rule types have already been discussed in Section 3.2.2 (on the classification of operational rules)
- ‘Rule body’: the rule expression verbalism.
- ‘Current version’: version of the collected rule statement.
- ‘Enforcement authority’: the authority that enforced the rule.
- ‘Enforcement status’: e.g. Active, abandoned, etc.
- ‘Enforcement date’: first date of enforcement of the rule.
- ‘Expiry date’: last date of enforcement of the rule.
- ‘Enforcement level’: how strict is the rule? Can it be bypassed?

A rule is also associated with a ‘rule collection session’ label, which contains information about the stakeholder that revealed the rule, the project for which this exercise takes place, the date that this rule was collected, and so on.

The organisation schema of Fig. 8 is sufficient only for supporting the ‘intentional’ phase of rule analysis, as it covers logistics’ issues and goal–rule relationships, but it does not show the links between rules and their universe of discourse. Therefore, we use the schema above during intentional rule analysis where we collect unstructured rule expression in natural language, and we adopt a more complicated schema for ‘operational rules’, in order to show internal rule structure with references to the information entities of Fig. 7. The following sub-section deals with this topic. As it has already been mentioned, ‘IS architecture rules’ are beyond the analysis activities described in this work, and therefore, there is not a dedicated repository schema for them.

4.2. The repository schema

Assche et al. [18] propose the structuring of rules according to the ECA (Event–condition–action) paradigm (Fig. 9), which was also widely used in the area of active databases [19]. According to the ECA paradigm, ‘as soon as an event occurs, and as long as all the conditions hold, the specified action must be performed’. A certain ‘operational rule’ may contain zero, one or more than one events (‘WHEN part’), zero, one or more than one conditions (‘IF part’) and one and only one action part (‘THEN part’).

An ‘operational rule’ may also reference one or more relationships of the following types: ‘activity/activity enabler relationships’, ‘activity/information object relationships’ or ‘actor/activity relationships’. This type of reference determines the context of the rule, especially in cases of complex condition parts, where different informational entities are examined, and it is not clear how these entities relate to each other. For example, a certain rule in the context of e-procurement may examine the status of a supplier and the price of a product in order to accept or reject this product offer. This rule references two relationships, one indicating that the supplier makes an offer, and one indicating that the offer concerns a product. Obviously, this information already exists in the information model developed during the initial stages of rules analysis, however, including in the rule statement itself makes things clearer during the whole analysis process.

As far as rules’ ‘WHEN part’ is concerned, it is used for storing the business ‘event’ that triggers their execution. Events can be defined as ‘indivisible atomic units of action, with no duration, that occur in the real world, and affect at least one business entity’ [20]. Although, event modelling is a quite complex area, and one can identify a wide variety of event types, we consider sufficient to refer to distinct business ‘events’ that trigger the execution of ‘activities’, without analysing them any further. On contrary, the analysis of ‘operational rule’ conditions and actions is significant in the context of rule management, and therefore, is discussed in detail in Sections 4.2.1 and 4.2.2.

4.2.1. A schema for describing the IF part of rules

Conditions contained in the ‘IF part’ of an ‘operational rule’ can be of the following types:

![Fig. 9. Structuring operational rules in the rule repository.](image-url)
Static attribute value conditions examine the value of information entity attributes (i.e., 'actor attributes', 'activity attributes', 'activity enabler attributes' and 'information object attributes'). Some patterns of static attribute value conditions are given below:

- \( (\text{Information Object}).(\text{Information Object Attribute}) \geq (\text{Value}) \)
- \( (\text{Information Object}).(\text{Information Object Attribute}) \in (\text{Value} 1), (\text{Value} 2) \)
- \( (\text{Information Object}).(\text{Information Object Attribute} 1) = (\text{Information Object}).(\text{Information Object Attribute} 2) + (\text{Information Object}).(\text{Information Object Attribute} 3) \)
- \( (\text{Information Object} 1).(\text{Information Object Attribute} 1) \geq (\text{Information Object} 2).(\text{Information Object Attribute} 2) \)

Dynamic attribute value conditions examine the transition of attribute values for a certain information entity:

- \( (\text{Information Object}).(\text{Information Object Attribute}) \text{ CHANGES FROM } (\text{Value} 1) \text{ TO } (\text{Value} 2) \)
- \( (\text{Information Object}).(\text{Information Object Attribute}) \text{ CHANGES FROM } (\text{Value} 1) \text{ TO } (\text{Value} 2) \text{ WITH } (\text{Value} 1) \geq (\text{Value} 2) \)

Static association cardinality conditions examine the number of information entity instances that one information entity instance is related to through relationships of a certain type:

- \( \text{CARDINALITY OF } (\text{Activity}).\text{uses.}(\text{Activity Enabler}) \geq (\text{Value}) \)
- \( \text{CARDINALITY OF } (\text{Information Object} 1).\text{relates_to.}(\text{Information Object} 2) \geq (\text{Value}) \)

Dynamic association cardinality conditions examine the changes of information entity instances that a certain information entity instance is related with through a certain relationship type:

- \( \text{CARDINALITY OF } (\text{Activity}).\text{uses.}(\text{Activity Enabler}) \text{ CHANGES FROM } (\text{Value} 1) \text{ TO } (\text{Value} 2) \)

Set conditions examine the whole information base, instead of specific information entity instances. Most of the times they contain functions like \( \text{SUM}, \text{MIN}, \text{MAX}, \text{AVG} \), and so on, supported by commonly used database query and programming languages. Examples of set conditions are:

- \( \text{SUM OF } ((\text{Information Object}) \geq (\text{Value}) \)
- \( \text{SUM OF } ((\text{Information Object} \geq (\text{Information Object Attribute}) \geq (\text{Value}) \)

The 'IF part' schema presented in Fig. 10 shows how rule conditions can be associated with information entity attributes and/or information entities, so that appropriate grouping of rules is facilitated. It is worth mentioning that one condition may be associated with more than one information entities or information entity attributes. For instance:

- Condition \( '(\text{Information Object}).(\text{Information Object Attribute}) \geq (\text{Value})' \) is defined on 'Information Object Attribute'.
- Condition \( '(\text{Information Object} 1).(\text{Information Object Attribute} 1) \geq (\text{Information Object} 2).(\text{Information Object Attribute} 2)' \) is defined on both 'Information Object Attribute 1' and 'Information Object Attribute 2'.
- Condition \( '\text{SUM OF } ((\text{Information Object}) \geq (\text{Value})' \) is defined on 'Information Object'.
- Condition \( '\text{CARDINALITY OF } (\text{Activity}).\text{uses.}(\text{Activity Enabler}) \geq (\text{Value})' \) is defined on 'Activity' and 'Activity Enabler'.

Fig. 10. A schema for describing the IF part of operational rules.
4.2.2. A schema for describing the THEN part of rules

Declarations or assertions contained in the ‘THEN part’ of a business rule can be:

- **Information operations**, dealing with the creation and deletion of information entities, e.g.:
  - CREATE (Actor)
  - DELETE (Information Object)

- **Attribute assertions**, asserting values to certain information entity attributes or deriving information entity attributes from other attributes of the same or different information entities, e.g.:
  - (Information Object). (Information Object Attribute) = (Value)
  - (Information Object 1). (Information Object Attribute 1) = (Information Object 2). (Information Object Attribute 2) + (Information Object 3). (Information Object Attribute 3)

- **Association assertions**, establishing association relationships between two instances of two different information entities (which already exist), e.g.:
  - (Actor). (Relationship). (Information Object)

- **Activity invocations**, initiating business or system activities:
  - (Activity)

- **Integrity violation reports**, indicating that the violation of the condition part leads to an erroneous state of the information base.
  - (Integrity Violation Report)

Fig. 10 presents a repository schema for structuring ‘THEN parts’. According to this schema different types of assertions, or declarations, or actions are defined on different enterprise information entities (‘actors’, ‘activities’, ‘activity enablers’, ‘information objects’ and their attributes).

It is worth mentioning that the type of an ‘operational rule’s’ ‘THEN part’ characterises the rule as a whole to some extent, although this cannot be shown on the above figure. Not all combinations of rule types (discussed in Section 3.2.2) and ‘THEN parts’ are acceptable.

According to the above table, an ‘operational rule’ with an ‘activity invocation’ in its ‘THEN part’ is an ‘activity invocation rule’. A rule with an ‘information operation’ in its ‘THEN part’ is an ‘information operation rule’. ‘Attribute assertions’ are contained in different types of rules: ‘prescriptive rules’ that assign values to information entity attributes on the occurrence of some event (we call these ones ‘attribute assertion rules’), ‘descriptive rules’ that specify sets of information entities (e.g. ‘inf. object set rules’, ‘act. enabler set rules’, etc.), and ‘descriptive rules’ that derive information entity attributes from other information entity attributes. The aforementioned derivation rules are linked with both attributes that are found on the left and on the right side of a derivation, as shown indirectly on Fig. 11, through the two types of relationships ‘is defined on’ and ‘is used by’ Table 2.

5. Business rules for the example application

The rule examples given in this paper relate to the evaluation of offer proposals on behalf of different suppliers, which concern sub-contracting of services in the context of electronic procurement for the construction sector.
The relevant enterprise activities are: 'Proposal Submission', 'Proposal Submission Termination', 'Proposal Evaluation', 'Proposal Evaluation Termination', 'Negotiation' and 'Negotiation Termination'. There are also various events that are of interest: 'Proposal Submitted', 'Proposal Evaluation Initiated' and 'Negotiation Offer Submitted'. A fraction of the information model describing the application under investigation is given below (Fig. 12):

- 'Proposal Evaluation' is an 'activity' and 'Supplier' is an 'actor'. 'Proposal', 'Negotiation Invitation', 'Negotiation Offer' and 'Contract' are 'information objects'.

The 'operational rule' examples of the following two sections (one for 'descriptive rules' and one for 'prescriptive rules') are presented according to a tabular template, where:

- the first row shows the context of the rule
- the second row presents the event that triggers the rule’s execution (if there is one)
- the third row contains conditions that need to be examined together with their types and the information entities that these conditions are defined on
- the last line provides the action part, again with its type and associated information entity

5.1.1. Descriptive rule examples

The first ‘operational rule’ example derives from a 'procurement' intentional rule which deals with ‘supplier/service provider selection’. Its natural language expression is as follows: 'A supplier with market presence rank > 15, experience rank > 10 and capacity rank > 50 is considered to be a preferred supplier'. The ‘THEN Part’ of the rule is an ‘attribute value assertion’ for ‘Supplier’, who is an ‘actor’, therefore, we are talking about an ‘actor attribute value assertion rule’ (a specialisation of ‘actor rule’).

Table 2

<table>
<thead>
<tr>
<th>Rule Type</th>
<th>Acceptable “THEN Part” Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor Attribute Derivation Rules</td>
<td>Attribute Assertion</td>
</tr>
<tr>
<td>Actor Set Rules</td>
<td>Attribute Assertion</td>
</tr>
<tr>
<td>Actor Entity Integrity Rules</td>
<td>Integrity Violation Report</td>
</tr>
<tr>
<td>Activity Attribute Derivation Rules</td>
<td>Attribute Assertion</td>
</tr>
<tr>
<td>Activity Set Rules</td>
<td>Attribute Assertion</td>
</tr>
<tr>
<td>Activity Entity Integrity Rules</td>
<td>Integrity Violation Report</td>
</tr>
<tr>
<td>Inf. Object Attribute Derivation Rules</td>
<td>Attribute Assertion</td>
</tr>
<tr>
<td>Inf. Object Set Rules</td>
<td>Attribute Assertion</td>
</tr>
<tr>
<td>Inf. Object Integrity Rules</td>
<td>Integrity Violation Report</td>
</tr>
<tr>
<td>Act. Enabler Attribute Derivation Rules</td>
<td>Attribute Assertion</td>
</tr>
<tr>
<td>Act. Enabler Set Rules</td>
<td>Attribute Assertion</td>
</tr>
<tr>
<td>Act. Enabler Entity Integrity Rules</td>
<td>Integrity Violation Report</td>
</tr>
<tr>
<td>Activity Invocation Rules</td>
<td>Activity Invocation</td>
</tr>
<tr>
<td>Information Operation Rules</td>
<td>Information Operation</td>
</tr>
<tr>
<td>Attribute Assertion Rules</td>
<td>Attribute Assertion</td>
</tr>
<tr>
<td>Association Assertion Rules</td>
<td>Association Assertion</td>
</tr>
</tbody>
</table>

The first row shows the context of the rule
The second row presents the event that triggers the rule’s execution (if there is one)
The third row contains conditions that need to be examined together with their types and the information entities that these conditions are defined on
The last line provides the action part, again with its type and associated information entity

5.1.1. Descriptive rule examples

The first ‘operational rule’ example derives from a 'procurement' intentional rule which deals with ‘supplier/service provider selection’. Its natural language expression is as follows: 'A supplier with market presence rank > 15, experience rank > 10 and capacity rank > 50 is considered to be a preferred supplier’. The ‘THEN Part’ of the rule is an ‘attribute value assertion’ for ‘Supplier’, who is an ‘actor’, therefore, we are talking about an ‘actor attribute value assertion rule’ (a specialisation of ‘actor rule’).

Context:-
WHEN Part
– Supplier.
IF Part
Market_Presence_Rank > = 15
Supplier.
Experience_Rank > = 10
Supplier.Capacity_Rank > = 50
Defined on
Supplier.
Market_Presence_Rank,
Supplier.Experience_Rank,
Supplier.Capacity_Rank

(continued on next page)
THEN Part Supplier.Overall_Rank = 'Preferred_Supplier'

Type: Attribute Value Assertion
Defined on Supplier.Overall_Rank

The second 'operational rule' also derives from a 'procurement' intentional rule that deals with 'supplier/service provider selection': 'A banned supplier cannot become preferred supplier'. This is an 'actor attribute value transition rule' (a specialisation of an 'actor integrity rule', which is a specialisation of 'actor rule').

Context:-
WHEN Part
IF Part Supplier.Overall_Rank CHANGES FROM 'Banned_Supplier' TO 'Preferred_Supplier'

Type: Dynamic Attribute Value Condition
Defined on Supplier.Overall_Rank
THEN Part Integrity_Violation_Report
Type: Integrity_Violation_Report

The third rule is concerned with 'service procurement': 'The status of the proposal evaluation process can only be scheduled, active, postponed, cancelled or completed successfully'. The corresponding 'operational rule' is an 'activity attribute value integrity rule' defining the allowed values of activity attribute 'Proposal.Evaluation.Status'. 'Activity attribute value integrity rule' is a special type of 'activity integrity rule', which is a specialisation of 'activity rule'.

Context:-
WHEN Part

Type: Static Attribute Value Condition
Defined on Proposal.Evaluation.Status
THEN Part Integrity_Violation_Report
Type: Integrity_Violation_Report

The last three 'descriptive rules' are all 'information object rules', also deriving from 'service procurement' intentional rules. 'The overall rank of a proposal is the sum of its financial rank and its technical rank': This one is an 'information object attribute derivation rule' (specialisation of 'information object rule').

Context:-
WHEN Part
IF Part Proposal.Overall_Rank

Type: Attribute Value Assertion
Defined on Proposal.Overall_Rank,
Proposal.Financial_Rank,
Proposal.Technical_Rank

And another 'information object association cardinality rule' is the expression 'The number of proposals submitted by a supplier can only increase'.

Context:
Supplier.submits.Proposal
WHEN Part
IF Part CARDINALITY OF Supplier.submits.Proposal CHANGES FROM n to m

Type: Dynamic Association Cardinality Condition
Defined on Supplier, Proposal
THEN Part Integrity_Violation_Report
Type: Integrity_Violation_Report

5.1.2. Prescriptive rule examples

All the 'prescriptive' rule examples given here derive from 'service procurement rules' at the intentional level. The first one is an 'information assertion rule': 'As soon as

Context:-
WHEN Part -
IF Part -

Type: Attribute Value Assertion
Defined on Proposal.Overall_Rank,
Proposal.Financial_Rank,
Proposal.Technical_Rank

An 'information object association cardinality rule' is the following: 'One proposal can only lead to one contract'. 'Information object association cardinality rule' is a specialisation of 'information object integrity rule', which is also a special type of 'information object rule'.

Context:
Proposal.leads_to.Contract
WHEN Part
IF Part CARDINALITY OF Proposal.leads_to.Contract.

Type: Static Association Cardinality Condition
Defined on Proposal, Contract
THEN Part Integrity_Violation_Report
Type: Integrity_Violation_Report

And another 'information object association cardinality rule' is the expression 'The number of proposals submitted by a supplier can only increase'.

Context:
Supplier.submits.Proposal
WHEN Part
IF Part CARDINALITY OF Supplier.submits.Proposal

Type: Dynamic Association Cardinality Condition
Defined on Supplier, Proposal
THEN Part Integrity_Violation_Report
Type: Integrity_Violation_Report
The proposal evaluation process terminates successfully, a new negotiation phase must be initiated. The result of this rule’s enforcement is the creation of a new information entity, namely, a new ‘Negotiation_Invitation’.

Context:
Proposal_Evaluation.uses/produces.Proposal
Proposal.leads_to.Negotiation_Invitation

WHEN Part Proposals_Evaluation_Initiated
IF Part Proposal_Evaluation.Status = ‘Completed Successfully’

Type: Static Attribute
Value Condition Defined on Proposal_Evaluation.Status
THEN Part CREATE Negotiation_Invitation

Type: Information Operation Defined on Negotiation_Invitation

The second ‘information assertion rule’ results in the establishment of a relationship between two information entities: ‘A proposal record can be associated with a supplier record if and only if the secret identifier of the proposal is identical with the secret identifier of the supplier’.

Context:
Supplier.submits.Proposal

WHEN Part Proposals_Evaluation_Initiated
IF Part Proposal.Evaluation.Status = ‘Completed Successfully’

Type: Static Attribute
Value Condition Defined on Proposal_Evaluation.Status
THEN Part CREATE Negotiation_Invitation

Type: Information Operation Defined on Negotiation_Invitation

The last example is a ‘workflow rule’: ‘If at some point during negotiation, the lower negotiation offer price is found to be lower than the 80% of the lower initial offer price, the negotiation process can be terminated’.

Context: Proposal.leads_to.
Negotiation_Invitation
Negotiation_Offer.responds_to.
Negotiation_Invitation

WHEN Part Negotiation_Offers_Submitted
IF Part MIN OF (Negotiation_Offer.Price) < 80% *MIN OF (Proposal.Price)

Type: Set Condition
Defined on Negotiation_Offer.Price, Proposal.Price
THEN Part Negotiation_Termination

Type: Activity Invocation Defined on Negotiation_Termination

As a conclusion, all the examples in Sections 5.1.1 and 5.1.2 demonstrate the strengths of the proposed rule approach, which allows expressing, structuring and organising business rule statements of many completely different types (e.g. integrity constraints, derivations, workflow rules) in a homogeneous way. Moreover, such expressions are easy to read and understand, as they are close to natural language, while containing widely used operators for expressing relations between entities/sets of entities. Finally, the MBRM approach assumes the existence of underlying information and workflow models, which ensures structural and notational consistency of rule expressions, and allows efficient rule grouping and retrieval. This last issue is also discussed in detail in Section 7 of this work.

6. Related work

Business rules, as part of requirements gathering and systems analysis, have not been ignored by structured analysis, information engineering or object-oriented analysis approaches [21], which, to varying degrees, subsume or represent them as part of notation schemes used to specify application requirements [22]. Ross [23] comments that traditional IS methodologies have addressed rules poorly, and only relatively late in the system development lifecycle. Rules dealing with information structure may be represented by any of several flavours of Entity—relationship or Object Class Diagrams, and responses to events may be shown via Essential Data Flow Diagrams [24] or as Entity Life History Diagrams [25]. There are fewer notations available, however, to describe action assertions. Most notable of these few is probably Object Role Modelling [26], derived from the Nijssen Information Analysis Method [27,28] investigate exhaustively the issue, and agree (more or less) on the same conclusions.

In recent years there has been an increasing interest of the IS community in business rules, which has resulted in dedicated rule-centric modelling frameworks and methodologies [23,29] (Gottetsdiener, 1999) international initiatives for the investigation of business rules’ role in the context of knowledge management [16], conferences, workshops and tutorials [30,31], and rule-centric rule management tools.
According to this criterion, a natural language rule (medium expressiveness) and 'LE' (low in expressiveness). Three possible values: 'HE' (highly expressive), 'ME' (medium expressiveness) and 'LE' (low in expressiveness). According to this criterion, a natural language rule statement is highly expressive, while a rule written in C++ or Prolog is not. The idea is that a good formalism should allow business people to understand, modify, or even write from scratch rule expressions with little support from IT experts and knowledge engineers.

‘Guidance’: the term ‘guidance’ refers to the methodological support that the approach provides, and again, this can be sufficient, or poor, or it may not exist at all. A methodology is by definition supposed to offer guidance about how to use it. A good rule methodology should provide precise information about how to collect and organise rules at the analysis level, and how to translate them to formal rule expressions at the design level.

‘Organisation’: ‘organisation’ has to do with how manageable rule models are. Rule models can be highly manageable, or they can be medium or low in manageability. A modelling framework with high manageability should allow the retrieval of rules according to different criteria, imposed by the needs of the various proposed analysis and design activities.

‘Rationale’: ‘rationale’ is concerned with the traceability of decisions that enforce or change rules, and its values are ‘HT’, ‘MT’ and ‘LT’ (high, medium and low traceability, respectively). It has been selected as an additional criterion of rule models manageability, as it is useful for changing and refining rules.

‘Formality’: ‘formality’ examines whether rule statements can be expressed based on a formal language or not (‘F’ is formal, and ‘inF’ is informal), and whether they are in accordance with other models that are necessary for the development of an information system (e.g. a data model).

‘Openness’: ‘openness’ deals with whether the examined approach assumes that the IS will be developed according to a specific implementation paradigm or even platform (e.g. using a certain active database). An ‘open’ approach allows the adoption of different architectures and platforms, in order to be able to choose the one that satisfies the needs of the particular application area better.

In BROCOM [10,11], the rule language is a type of structured English, and therefore it is highly expressive. Moreover, rules are organised according to a rich metamodel, and can be retrieved based on a number of different criteria. On the other hand, the rationale behind business rules is not addressed at all. As far as methodological guidance is concerned, Herbst proposes the development of various models, which are of great help during the analysis phase, but the process of creating and using them is not clearly defined and described. As an overall criticism, one could say that the BROCOM approach offers a quite satisfactory rule-centric solution for the phase of IS analysis. However, the transition from analysis to design and implementation has not been addressed, which is not necessarily a bad thing, as a specific implementation paradigm has not been taken for granted, and therefore, the adoption of any convenient architecture and platform is allowed.

DSS [9,32] is another approach that can assist in the analysis phase of IS development. Its focal point is the support of the rationale behind the establishment of rules. Moreover, DSS adopts the ECA (Event–Condition–Action) paradigm for structuring rule expressions and also links these expressions to the entities of an underlying enterprise model. Therefore, DSS rule models are generally manageable. On the other hand, the existence of a formal rule language could extend the applicability of DSS to the rule specifications generation and rule implementation phases.

The GUIDE project [16] identifies terms and facts in natural language rule statements, and consequently, the expressiveness it allows is very high. The meta-model it provides for describing the relations between these terms and facts is very detailed. This is good for two reasons: (a) GUIDE rule models are highly manageable; and (b) they are very formal and fully consistent with the information models of a specific organisation. A disadvantage of GUIDE is that it does not distinguish between informational and workflow concepts.

IDEA [29] is the only methodology that fully covers all the stages of the IS development lifecycle. The maintenance of formality and consistency with underlying business models are two of its focal characteristics. Another advantage is that it offers guidance for every activity being involved in the development of a rule-centric information system. A disadvantage is that the IDEA methodology is directed towards the use of specific active and deductive databases, and of the corresponding rule languages. As a result of this, (a) IDEA rules are rather difficult to be expressed or even understood by business people; and (b) the choice of technologies to be employed for the development of an information system is rather limited.

Martin and Odell [1] focus on defining business rules and on proposing a rule typology that has been adopted by many methodologists. They also indicate ways that rules can be integrated with various other (OO) IS analysis techniques.

The Ross Method [23] is one of the most complete methodologies presented here. It is formal, in accordance with the underlying data models of an organisation, offers sufficient methodological guidance, and allows management of rule expressions based on a very detailed metamodel. It is also the only methodology that adopts a graphical notation for expressing rules. However, this does
not seem to be an advantage, as the complexity of the resulting diagrams and the vast amount of graphical symbols they contain makes them impossible for to be understood by business people.

The Object Constraint Language of UML [33] has also got advantages and disadvantages. Although it is tightly bound with the widely accepted UML diagrams that assist in the design of OO information systems, the formalism it proposes is not very simple to understand and use. Moreover, it does not provide any methodological guidance for the collection of rules. Finally, it does not propose a meta-model for organising them. Instead, the structure it allows is implied by the allocation of rules to classes, attributes, associations and operations.

7. Summary and observations

This paper puts forward a repository structure for organising business rules for use within rule-intensive projects. The proposed repository organisation is inline with a goal-based rule-collection methodological approach, which identifies three distinct rule analysis stages: intentional rule analysis, operational rule analysis and IS architecture rule analysis.

During intentional rule analysis different stakeholders express their view on the objectives and constraints of the project. The various kinds of rule expressions collected during this stage are reflected by the intentional rules classification, where rules are examined with respect to their business context. The classification is grounded on pragmatic use of business rules from its use on extensive consulting experience from analysing different industrial domains, and from designing information systems either from scratch or based on the available functionality of customisable packaged solutions.

The second stage of rules analysis is concerned with the impact of business constraints on business processes. During this stage, ‘intentional rules’ are translated to ‘operational rules’ expressions, which describe structure of information entities or prescribe action on the occurrence of business events. ‘Operational rules’ are structured according to the ECA (Event–Condition–Action) paradigm, where the event part is present in ‘workflow’ or ‘information assertion rules’, while it can be omitted, as trivial, in case of ‘actor’, ‘activity’, ‘activity enabler’ and ‘information object rules’. Such rules deal with information derivation, information integrity and information entity sets definition.

The third stage of rule analysis is beyond the scope of this work, as it depends on the selected rule implementation paradigm (i.e. on whether rules will be implemented in an active database system, a dedicated rule engine and so on).

The main contribution of the work presented here relates to the repository schema for structuring and organising rules, which links rules with goals and other derived rules, with information about the rule collection sessions that revealed them, with business or information system events that trigger their execution (if there are any), and with the various information entities (i.e. ‘actors’, ‘activities’, ‘activity enablers’ and ‘information objects’) that constitute their context, and that characterise their ‘IF’ and ‘THEN parts’. More specifically, the adoption of the proposed rule management approach can be useful in the following contexts:

- Strategic planning
- Business process (re-)engineering
- Resource management and training
- Project management
- Data management

The following table explains how this is possible:

<table>
<thead>
<tr>
<th>To assist strategic planning…</th>
<th>For example, we can retrieve…</th>
<th>This is possible…</th>
</tr>
</thead>
<tbody>
<tr>
<td>… by allowing grouping of rules enforced by the same enforcement authority.</td>
<td>… all the rules enforced by the ‘European Commission’.</td>
<td>… by using the repository schema of Fig. 8.</td>
</tr>
<tr>
<td>… by allowing grouping of rules relating to a certain business goal (and in some cases indicating rule conflicts or overlaps).</td>
<td>… all the rules that realise goal ‘To Support Electronic Evaluation of RFPs’.</td>
<td>… by using the repository schema of Fig. 8.</td>
</tr>
<tr>
<td></td>
<td>… all the rules that hinder goal ‘To Support Electronic Evaluation of RFPs’</td>
<td></td>
</tr>
</tbody>
</table>

(continued on next page)
... by allowing grouping of rules pertaining within a particular application area (sales and marketing, procurement, etc.).

... all the ‘operational rules’ that derive from ‘suppliers relationship management intentional rules’.

... by using the repository schema of Fig. 8.

---

**To assist business process (re-)engineering...**

... by allowing grouping of rules which are triggered by a particular business event.

... by allowing grouping of rules which examine the status of a particular business activity.

... by allowing grouping of rules which affect the status of a particular business activity

**For example, we can retrieve...**

... all the rules that are triggered by event ‘Proposal Evaluation Initiated’.

... all the rules that are executed when the status of activity ‘Proposal Evaluation’ is ‘Cancelled’.

**This is possible...**

... by using the repository schema of Fig. 8.

... by using the repository schema of Fig. 9.

---

**To assist resource management and training...**

... by allowing grouping of rules that concern a particular type of actor (e.g. sub-contractor, project manager, etc.)

... by allowing grouping of rules that concern a particular type of activity enabler (e.g. equipment, warehouse, etc.).

... by allowing grouping of rules that concern a particular type of information entity (e.g. document, account, etc.).

**For example, we can retrieve...**

... all the rules that examine a supplier’s overall rank in their ‘IF part’.

... all the rules with ‘THEN part’ defined on actor ‘sub-contractor’.

**This is possible...**

... by using the repository schema of Fig. 11.

... by using the repository schemata of Fig. 10 and Fig. 11 respectively.

---
To assist project management…

… by allowing grouping of rules revealed by the same stakeholder.

… by allowing grouping of rules belonging in the same rule model version.

… by allowing grouping of rules associated with the same enforcement status (e.g. enforcement date).

… by allowing grouping of rules associated with the same enforcement level (i.e. rule importance).

To assist data management…

… by allowing grouping of rules that examine the integrity of information entities.

… by allowing grouping of rules that affect a given information entity.

For example, we can retrieve…

… all the rules reported by stakeholder ‘Sales Representative’.

The final version of all the rules of the rulebase.

… all the rules that are going to be enforced after the 1st January 2004.

… all the rules with enforcement level ‘Legal Requirement’.

For example, we can retrieve…

… all the rules with ‘integrity violation report’ in their ‘THEN part’.

… all the rules with attribute assertions in their ‘THEN part’, defined on proposal’s technical rank, financial rank and overall rank.

This is possible…

… by using the repository schema of Fig. 8.

… by using the repository schema of Fig. 8.

… by using the repository schema of Fig. 8.

… by using the repository schema of Fig. 8.

Obviously, the exploitation of the strengths of the rule management approach presented in this paper is tightly bound with the availability of suitable software tool support. The implementation of a professional rule management package based on the proposed rule repository organisation schema (beyond the boundaries of research activities reported here), constitutes a major aspiration for future work and development.

Acknowledgements

The authors wish to acknowledge the contribution of the industrial project partners for their assistance and cooperation in the application described in this paper. Also, the authors wish to express their gratitude to the anonymous reviewers for their useful comments on an earlier version of this paper.

References


