
Analysis and design of business rules

Brian Byrne
Greg Imirzian

18 September 2008

This is the first of two articles in a series that identifies the nature of business rules and the relationship of these rules to IBM® Industry Models. In particular, this article references the deployment stages of the IBM Industry Models during which rules analysis and design should be considered, and how rules can be identified and managed. The second article in this series discusses taking these analysis and design structures into the ILOG rule management environment to develop completed rule sets.

View more content in this series

What are IBM Industry Models?

The IBM Industry Models are a collection of interrelated models addressing different aspects of the analysis and design of applications, data storage, or integration solutions for a particular business domain. The models consist of a set of foundation models, which in turn support a set of detailed models focused on a specific problem or modeling domain.

The foundation models assist in the definition of project scope and in the identification of the dependencies between individual projects that may otherwise have been overlooked. In addition, the foundation models provide the basis on which all other Industry Model offerings are built, providing a lexicon of standard business terms that drive consistency across the model set. The derived models target the needs of the process, service, and data domains with process, UML service, and ER models respectively.

The focus of the Industry Models is on the provision of detailed business content. Each model is the result of many years of development with organizations worldwide to construct large enterprise models that may then be used to drive project efforts along a common path. The models are intended to be customized and extended during these project initiatives, which requires model customization and governance methodology.
The foundation models within each industry offering exist to drive consistency across the model suite and to enable a business entry point to the models. These models consist of a set of business definitions, primarily detailing the business concepts functions and actions that are considered across the model set.

Derived from these foundation models are a set of issue-specific models that extend these business definitions into the process, service, and warehouse domains. Figure 1, below, highlights the specific models that are discussed in this article:

**Figure 1. IBM Industry Models**

![Diagram of IBM Industry Models](image)

**Why are business rules important?**

Business rules manifest in a variety of locations within this model set, particularly within process analysis. Certain forms of rules are implicit in the definition of the business types in the model, and manifest in UML and entity relationship (ER) models of the business types of the modeled domain. Other forms of rules detail the allowed changes in these business types, manifesting as constraints within service definitions, and state transition models. We will also encounter rules that affect the flow control of business logic. These rules are usually encountered within process or service flow control, within the process and service analysis models.

**Figure 2. Sources of rules in Industry Models**

![Diagram showing sources of rules](image)
This gives rise to a defined set of rules origination patterns; parts of the Industry Models where rules tend to arise and require specification:

- **Rules relating to business processes or activities**: These rules relate to flow control logic within a business process model, affecting the branching, and flow control of a sequence of business tasks. Rules of this nature tend to arise at analysis time within process models and require formalization to drive soft coded process flow (for example, BPEL) or application logic.
- **Rules relating to services**: These rules relate to the execution of service logic, and relate closely to rules relating to business processes and activities. As a consequence, these rules often also arise during process analysis; however, they require further analysis as part of the business object model before being formalized as part of service specifications within IDM.
- **Rules relation to information**: These rules relate to information, relationships between information, and the permitted states of information. Rules of this nature arise during informational analysis, either as part of the business object model or logical data modeling. These rules may be formalized within service specifications to assert control over information structures or within the information platform itself.

All of these rule types must be identified, analyzed, and designed in a structured way, maintaining their relationship to the model artifacts in which they originated and the model artifacts that they reference or operate on. This requires a defined method to support the analysis and design of business rules, a defined set of formats in which these rules can be captured, and formalized relationships between these rules and other model artifacts. It's important to capture and manage these rules because they are an integral part of the requirements definition, analysis, and design of software systems, irrespective of whether these rules are ultimately externalized to a rules engine, hard coded in software, or implemented through human procedures. These rules compliment the data structures, processes, and services of the Industry Models, allowing a complete picture of business requirements to be expressed.

**What is a business rule?**

A business rule is a statement that defines or constrains some aspect of a business. The intent of a business rule is to control or influence some aspect of a business through the imposition of structure.

**Forms of a business rule**

The term *business rule* is extremely broad and requires further classification. For the purpose of this article, we'll deal with four distinct forms of business rule:

- **Business terms**: A *business term* is the most fundamental form of a business rule. Through the definition of a term, you are formalizing how stake holders communicate about a concept. Terms are traditionally captured with a business glossary or as concepts within a model (such as an ER model). An example of a business term is *customer*, defined for our purposes as "A role played by a party that is considered to be receiving services or products from the modelled organization or one of its organization units, or who is a potential recipient of such services or products".
• **Facts:** The structure of a business can be described in terms of facts, which relate business terms to each other. Facts may take the form of natural language statements or as attributions, relationships, or generalizations within a formal model. An example of a fact is "a customer has an address".

• **Derivations:** Derivations describe how information in one form may be transformed into information in another form through an inference or mathematical calculation. An inference produces a derived fact using logical induction or deduction. A mathematical calculation produces a derived fact according to a specified mathematical algorithm. Examples of derivations are "two customers with the same address, first name, last name, and date of birth are the same customer" (inference) or "average customer profitability is a summation of all customer profitabilities divided by the number of customer profitabilities" (mathematical calculation).

• **Constraints:** While facts describe relationships between types within the business domain, constraints describe limitations in that domain. Constraints act upon a business term (a subject) through a correspondent rule, usually a fact. A correspondent rule is the rule that enables a constraint. An example of a constraint is "a customer must have exactly one legal address". The subject of this constraint is customer. The correspondent rule (fact) is that customers have addresses.

**Degrees of formalization**

After defining these four forms of a business rule, let's discuss the meaning of "rules" and how rules relate to each other. However, different participants in the requirements-analysis-design cycle express rules with varying degrees of formalization and structure.

Business users tend to express rules extremely informally, usually though the use of natural language, and usually grouping multiple rules into a single statement (for example "...potential customers must be credit checked and identified..."). Here, we call these informal definitions Business Rule Statements. Business users also tend to express themselves through Business Policy, such as "we only deal with credit checked and verified customers". Policies and rules, although closely related, are distinct constructs. Business policies tend to originate and group distinct business rule statements, acting as a general guideline or context to the rules.

Analysts tend to break rules into atomic rule elements - rule definitions that have been broken down as far as possible without the loss of business information. A given business rule statement often results in multiple interrelated atomic business rules. While still expressed in the form of natural language, atomic business rules tend to be more formalized and precise, and will normalize a rule definition, eliminating overlap between related rule statements.

Rule designers can then take these atomic rules and express them in a format specific to the target environment. Where a rules engine is to be used, this format is usually specific to this rules engine. Note that a single atomic business rule may manifest as a number of formal rule statements, each with a different formal expression type. It is not necessary that this expression be related to an externalized rule execution environment or rules engine, rules may often be expressed "hard-coded" into application logic, or indeed executed by staff within an organization.
Figure 3. Degrees of formalizing business rules

In using the Industry Models, a given organization will pass through the phases of planning, requirements gathering, analysis, and design in constructing solutions to meet business objectives. In order to formalize these phases, a roadmap containing a set of methodological principles has been defined to accompany the models. For example, in the SOA space, this roadmap guides the organization through identifying the boundaries of projects, analyzing business processes, identifying services, and on through the development lifecycle. (The roadmap is included with the Industry Models.)
While this roadmap provides general guidance for project teams using the models, it is focused on the particular challenge of identification, analysis, and design of reusable services. Clearly this is not the only thread of analysis and design to be considered. You must also consider the analysis and design of elements, such as user interface, human behavior, and of course business rules.

Figure 4. Customizing Industry Models

Figure 5. Multiple threads of analysis and design

The intent of this article is to document the related thread of analysis and design of business rules, in the context of IBM Industry Models.

In practice, we find that unless a project is entirely green field, in addition to a top-down analysis, there is a simultaneous bottom-up analysis of pre-existing code and business logic. Code, spreadsheets, database tables, and so on are examined and give rise to atomic business rules, and potentially entirely new data constructs. This analysis is then reconciled with the results of top-down analysis to give more complete picture.
Rule services

Exposing business rules as services eliminates much of the complexity associated with the variations in rule expression syntax and rules engine-specific support. Identifying a reusable service definition to encapsulate atomic rules allows those rules to be exposed as a Web service, expressed in WSDL, isolating the client of the rules from the complexities of how those rules are implemented. For this reason, the approach described in this article retains a strong connection between atomic rules and the parts of the business model in which they originated. Many business rules may be mapped to business activities, which in turn drive the definition of business services.

Figure 6. Defining rule services

Retaining this link provides a means to analyze and define the services that encapsulate rules in a way that is independent of the definitions themselves. Additionally, rules then operate on reusable business objects, derived from IDM, rather than allowing each rule to operate on its own specific data sets and platform specific representations. Having established a consistent set of WSDL interfaces through which rules may be invoked, those rules may be implemented using a wide range of technologies without forcing the clients of those rules to accommodate a range of different techniques.

Business policies

Business policies are informal expressions of business direction, usually formulated by business users to guide the business as a whole. These policy definitions tend to appear at an early stage in the analysis design lifecycle and are often expressed additionally as detail during business process analysis. Policies themselves rarely manifest as business rules; they tend to act as the guiding principles within which rules are expressed.
**Figure 7. Example business policy**

```
BUSINESS POLICY

."..we only deal with fully credit checked and varified customers.."
```

**Business rule statements**

Business rule statements provide the additional detail necessary to enforce a business policy. These are also informal and unstructured statements, expressed directly by business stakeholders. Business rule statements often contain multiple discrete instructions that are closely related, and it is often non-trivial to break business rule statements into atomic business rules.

Business rule statements are usually closely related to activities within a business process, and it should be possible to identify the activity or activities within a process where a set of business rule statements are relevant. Doing so helps to ensure a clean linkage between this activity in a business process and the rules that are ultimately required to support it. Despite this relationship between business rules and the business activities that invoke them, the actual invocation of these rules is through business services. The Industry Models Customization Roadmap tells us that most business activities result in service candidates. This is no different in the case of business rules -- a business activity that implies rule statements will resolve to a business service. That service is in fact the implementation of those rules, often within a rules engine. This decoupling of processes and rules, through the use of services, means that services that encapsulate rules do not operate on the data structures of the process. These services have input and output data structures just like any other service. Communication between the process and the rule itself is through the interface of the service.
Atomic business rules are formalized rule statements, expressed in natural language, independently of any specific rules syntax. Transforming these atomic rules into a given rule syntax yields rule expressions. Note that each set of rule expressions is just one way of expressing a given set of business rules, and is dependent on the syntax used to express them. It is reasonable to assume that a given rule may be expressed in multiple formats at different times, and that multiple rule expression types may be in use in parallel. The syntax used in rule expressions statements is highly dependent on the target execution environments. Here, the choices are wide ranging -- from compiled languages to dynamic rules engines. This article explores just a few of these possibilities in the following chapters.
Capturing policies and rule statements

Existing industry model customers are familiar with the recommendations of the Industry Models Customization Roadmap, which highlights the benefits of capturing textual requirements definitions during process analysis. An example template for these requirements is provided, a subset of which is shown in Table 1, below. (The template is included with the Industry Models.)

Table 1. Activity description document

<table>
<thead>
<tr>
<th>Activity</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business rules</td>
<td>A definition of any specific rules pertaining to this activity. These are often provided as a bulleted list or “if... then” conditional statements that describe how the activity is to be executed.</td>
</tr>
<tr>
<td>Critical success factors</td>
<td>A definition of any preconditions or success factors that must exist for this activity to execute successfully. These are often provided as a bulleted list. This may include a description of innovated areas that will be done differently in the new process.</td>
</tr>
<tr>
<td>Constrains</td>
<td>A definition of any known limitations or restrictions on the process. This may include legal or regulatory requirements as they pertain to this activity.</td>
</tr>
<tr>
<td>Dependencies</td>
<td>A definition of any logical units or other processes upon which this activity will depend for its operations. This includes policy changes and organizational decisions that must be made in order to implement this activity.</td>
</tr>
<tr>
<td>Non-functional requirements</td>
<td>Definitions of any supporting requirements that are not directly involved in the business definition of this activity, but still affect the successful outcome of this activity, such as key performance indicators and systems support requirements for the business.</td>
</tr>
</tbody>
</table>

The intent of this template is to encourage the capturing of textual requirements in a semi-structured way, such that analysts may later work with them. Documents based on these templates may be created and managed alongside the formal models, providing the requirements input needed to drive analysis and design threads.

Rule-based requirements in RequisitePro

Management of these semi-structured requirements may be accomplished in a number of environments, however, IBM Rational® RequisitePro provides many useful capabilities. RequisitePro can exist as an eclipse-based plug-in into either the IBM WebSphere® Business Modeler (WBM) or Rational Software Architect (RSA) toolsets, enabling structured requirements documents to be mapped to model elements. This allows business policy and rule statement information to be captured in textual form, and mapped directly to the business activities and UML structures to which they relate.
Relationships may also be established between these requirements types. For example, Business policies may be linked to the business rule statements that support those policies, as illustrated in Figure 12 below:

Figure 10. Policy and rule statements in RequisitePro

As well as relating these requirement types to each other, they can be related to the business model elements where they are relevant. For example, in a banking context, the business policy "...we only deal with fully credit checked and verified customers..." could be expressed as a requirements document, mapped to the business process "Provide Banking Product Arrangement Offer", which deals with taking a documented set of customer needs and producing an offer to meet those needs. Similarly, the business rule statement "...potential customers must be credit checked and identified..." could be mapped to the activity Analyze Customer Relationship within this same business process. This provides a means to capture not just the simplified examples presented here, but complex policy and rule statements, linked to the business contexts in which they apply. It is a simple matter to define custom templates in RequisitePro, adding properties as required by the organization to best capture their rules requirements and management of those rules. However, starting with the Industry Models Activity Description Documents provides useful guidance.

Note that this use of RequisitePro to manage business policy and business rule statements fits well with current tooling recommendations around the Industry Models. For example, there are documented procedures for managing business scope, functional and non-functional requirements related to the models using Requisite Pro. These recommendations, combined with the above
management of policies and rule statements allows these elements to be related to specific project initiatives and business goals, both within Rational Software Architect (RSA) and WebSphere Business Modeler (WBM).

Conclusion

This article discussed the nature of business rules and their relationship to IBM Industry Models. The article then discussed techniques for the capture of rule-based requirements and degrees of formalization of these requirements. The next article in this series takes these results and discusses the use of these rule structures within the ILOG BRMS to fully specify rulesets for invocation during business process or application execution.

© Copyright IBM Corporation 2008
Trademark
(www.ibm.com/developerworks/ibm/trademarks/)