White Paper

Modeling Business Rules

WP0193 | June 2015

Roger Evernden

Roger Evernden has been an Enterprise Architect since 1984, specializing in the highly practical use of EA to manage enterprise transformation.

He acts as advisor, mentor and coach on enterprise architecture initiatives, leads training workshops, and writes regularly about strategy and architecture. His work has been the basis for more than 400 business and IT architecture initiatives worldwide. As author of the Information FrameWork (IFW) – an architecture framework originally developed for the financial services sector – Roger pioneered many contemporary techniques, including the use of industry reference models, business capability analysis, and component-based architecture building blocks.

Modeling Business Rules

Business rules are one of the key components in an enterprise architecture. And yet for many EA teams, business rules hardly get a mention!

When you look at TOGAF 9.1, business rules are mentioned in passing during a brief discussion of activity models in a section on business modeling [Section 8.2.3]. TOGAF simply says that: "Activity models can be annotated with explicit statements of business rules, which represent relationships among the ICOMs. For example, a business rule can specify who can do what under specified conditions, the combination of inputs and controls needed, and the resulting outputs."

ArchiMate doesn't fare any better: business rules are not part of the current version of ArchiMate. And rules cannot be found in any of the cells of the Zachman framework.

So maybe we need to start this white paper by asking: why are business rules important for Enterprise Architecture?

The Role of Business Rules in EA

Enterprise Architects frequently need to explain the structure of the enterprise architecture. It might be to explain why the current architecture can't support important business needs. Or it might be to demonstrate how a different configuration would enable a critical new strategy. There are clearly many ways to explain the structure and behaviour of a particular architectural state. Some of these explanations are

based on EA theory. For example, we might use a layered architecture separating services from a process layer so that we can reuse services in many different process contexts. But if we keep asking why we need this separation, eventually we are likely to find that there is a business principle, policy or rule that determines this need. So one of the roles for business rules in EA is to help understand why we need to structure an enterprise architecture in a particular way.

If we approach this from the other direction, if we are given a business rule then it may well determine how we need to structure the enterprise architecture. As an aside, it is worth emphasizing these two related analysis techniques: repeatedly asking “why” will help discover the business directives that drive EA towards a particular state; repeatedly asking “how” will help develop an EA that supports the required business rules.

To put it simply: business rules lie at the center of much EA analysis and design! Which explains another important point about business rules: too often business rules are hidden or embedded within an architectural component, instead of being an external component in their own right. For example, an interest rate calculation might be hardwired into application software, or the terms and conditions for a product might be enforced through a related process. There are still organizations that have to get a programmer to update the interest rate in an application every time it changes. And there are even more organizations that have separate processes for each product type, instead of a parameterized template process that can be used for multiple products.

Business rules therefore have two key roles in EA:

1. Business rules determine some of the EA components, their structure, relationships and behaviours.
2. And when business rules are treated as a key component in their own right, they allow greater flexibility and agility in related parts of the enterprise architecture.

How to Model Business Rules

So how do we model business rules in EA? (And just to be clear, this White Paper is looking at business rules from the EA perspective, rather than modeling business rules in general.)

First of all we need to include business rules in the EA metamodel as a distinct construct in their own right. The Business Motivation Model (BMM, originally developed by the Business Rules Group (BRG), provides a scheme for managing business plans in a structured manner, and this is a good reference point for developing your own metamodel.

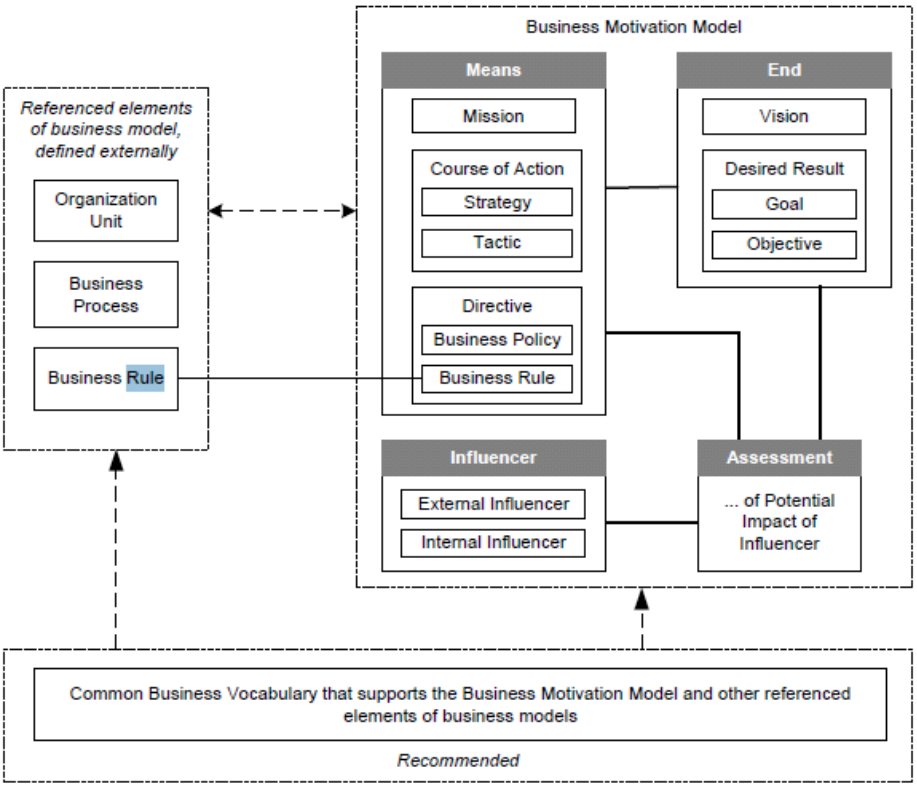

Figure 1: Overview of the Business Motivation Model

Many components in EA come in a variety of flavours – and this is true for business rules. For example, we might think of an application as commercial, developed in-house, or shareware; we might also distinguish between applications depending on their functionality, such as word processing, enterprise resource management or database. Exactly how you choose to classify the different types of business rule will depend on your circumstances and needs. We have included one suggestion in this white paper:

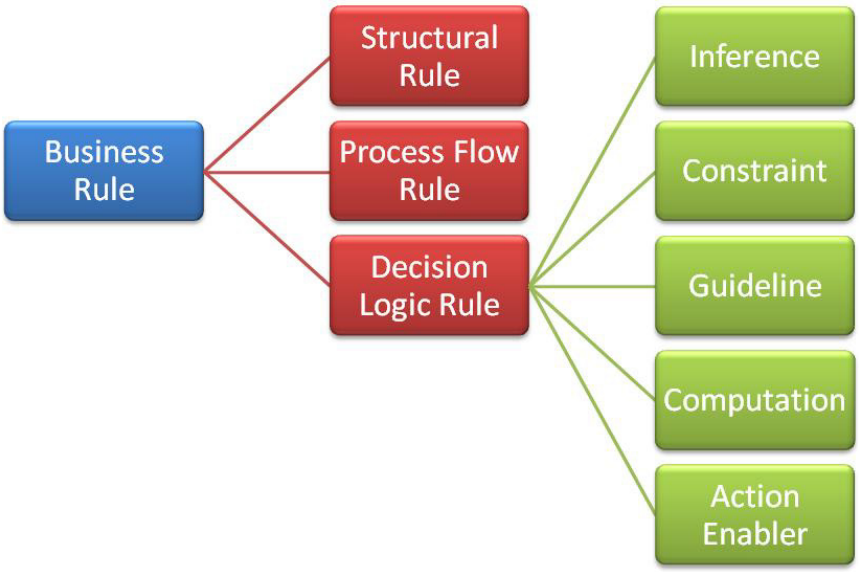

Figure 2: A Business Rule Classification for EA

The reason for making these distinctions is that the different types of business rule need to be modeled to reflect their role. For example, a structural rule is a rule that determines how other architectural components are organized or grouped together; a structural rule might determine how product elements are bundled together to create a product package that can be sold to a customer. This might be used by a telecom company to allow sales staff to tailor phone packages to suit particular customer needs. A process flow rule might describe the allowable sequences in a set of tasks or activities. This might be useful in adaptive insurance case management, where someone investigating a claim has to carry out certain procedures where the precise order of tasks is not predetermined.

Decision logic rules are the ones that most people think of when they talk about business rules. Our example classification breaks this type of rule down further, into a number of subtypes. For example, a guideline is a rule that indicates best practice or a recommendation, that isn't necessarily enforced. This needs to fit within the overall enterprise architecture so that the necessary guidance is provided when appropriate, without imposing unnecessary constraints on the flow of the decision process.

Another key point is to consider where the various types of rule components fit, relative to other components. In the past, business rules were often hardwired or embedded in other components.

Making them a distinct component means that there are likely to be strong relationships with the components where they were previously embedded. For example, there are likely to be strong relationships from rules to applications, processes, products, conditions, events, or triggers. There may also be strong relationships from several of these components to one rule. For example, in banks there are often separate account opening processes for each type of product, and there are often separate applications for each type of product. By extracting the business rules related to product conditions it is possible to have template products and processes that apply to families of related products; because the business rules are no longer rigidly fixed in the process or product components, they can be easily tailored to cover each product or process variation. In this example, there is a strong relationship between business rule, product, process and condition.

Another consideration is to think about where business rules fit within a layered enterprise architecture. For example, if there are a very large number of business rules, covering a complex range of business situations, then it might be useful to separate all business rules into a distinct layer. When it comes to decisions about solutions to realize the architecture requirements, this separate rule layer may be provided by a rules engine, or by a complex event processing system. It may also be implemented using more traditional approaches. The important point is that architecturally it helps to understand the overall enterprise architecture if rules are considered separately. As with all EA decisions, it is useful to record the rationale behind such choices; for example, it might be useful to place all business rules together because it is easier to maintain integrity across all rules. When EA teams consider business rules holistically in this way, they frequently come across inconsistencies that weren't apparent to stakeholders or business managers!

The next stage in modeling business rules is to catalogue the rules that are relevant to a particular EA initiative. Ronald Ross, who many regard as the "father" of business rules, and the Business Rules Group provide many guidelines and recommendations on the best ways to document rules . For example, there are a number of principles that can be applied to business rules:

- Each business rule should have an explicit expression.
- The formal presentation of business rules should have a coherent representation.
- The presentation should allow for evolutionary extension. For example, a business rule notation might be a natural extension of entity/relationship diagram notation; business rule notation might need to evolve in the future to handle emerging EA needs.
- A business rule should have a declarative nature, rather than a procedural one.

As with all of the suggestions in this White Paper, you will need to consider the exact requirements of your organization with regard to business rule modeling, and then modify these principles as necessary.

As the number of business rules modeled increases, so it might be necessary to consider classifying them using hierarchical indexing. It is good practice to group rules according to the subject matter that they relate to. For example, there may be a large number of rules that relate to customers, and it makes sense to group these together as a set. This makes it easier to see the big picture – the complete set of rules that determine how the enterprise engages with customers. It also makes it easier to spot any inconsistencies. Finally, by grouping rules hierarchically it makes it easier to identify when a more general rule could apply in many different situations. For example, a business rule might state: “a customer that has lodged an insurance claim must be assigned a dedicated case agent”. Another rule might be that “a customer who requests a review of their insurance products must be assigned a dedicated case agent.” Clearly there is some overlap between these two rules, and it might be possible to create a simpler, more general rule that covers both situations.

As the EA team work with a hierarchical index of business rules, they may also want to adopt a more formal and structured business language. The more that business rules are modeled, the easier it will be to standardize the language that is being used. I frequently get asked whether using a structured language is a problem: do business people resist using a standard words or a standard format for expressing rules? I have never found this to be a problem, although there may be some resistance at first. Business people can see the benefits as quickly as the EA team, and it isn’t usually long before business people are suggesting ways to improve and standardize the language. Business rules should be restated as the standard language emerges to steadily improve consistency.

Whenever the modeling process uncovers issues, gaps or conflicts – either in individual rules or in the set of business rules as a whole – they should be flagged for resolution. Again, this simple step helps to develop a consistent enterprise business rule model. Many of the terms used in rules will be found in data models or the data reference architecture. This, and other reference models or artefacts available to the EA team, should be reused whenever possible.

Conclusions

Business Rules are a neglected, but vital, component in an enterprise architecture. Once this has been recognized, business rules can be treated as a discrete component. This may require some adjustments to the EA metamodel, and in relationships between business rules and other components in the architecture. Many of the techniques for business rule modeling are the same as ones used for other EA components. In addition, there are many guidelines published by the innovators and experts who have pioneered business rule modeling, and these can easily be found on the Internet.

Modeling business rules is a particularly rewarding addition to EA practice, as it can help establish a more agile, flexible and adaptive enterprise architecture by creating dynamic business rule components, instead of perpetuating the hard-wired, rigid business rule approaches of the past.

www.businessrulesgroup.org/first_paper/BRG-whatisBR_3ed.pdf

Orbus Software UK
London

Orbus Software US
New York

Orbus Software AUS
Sydney

Orbus Software RSA
Johannesburg

enquiries@orbussoftware.com | **www.orbussoftware.com**

Seattle Software Ltd. Victoria House, 50-58 Victoria Road, Farnborough, Hampshire, GU14 7PG. T/A Orbus Software. Registered in England and Wales 5196435