



## Knowledge Automation: How to Implement Decision Management in Business Processes

**Alan N. Fish**

Wiley, 2012

\$57.22, 180 pages

### Reviewed by Paul Harmon

Computing languages have always depended on rules to specify how data is to be interpreted and manipulated. COBOL, for example, includes rule statements within its code.

In the Eighties there was a major movement, termed either Artificial Intelligence (AI) or Expert Systems, that sought to commercialize a new approach to software development that relied on stand-alone rules that could be examined by a software algorithm termed an inference engine. In essence the rules were logical statements:

If A is true and B is false, then C is true.

With the use of an inference engine, the software developer was freed from developing long chains of logic, and only had to define specific statements that an inference engine could examine and link at runtime. Obviously, if this approach was to work, developers had to be careful how they used their terms. Thus, if a backward-chaining inference engine was asked if C was true, it would search its rule base to see if it had any rules that concluded that C was true. If it found the rule given above, it would then go looking for rules that concluded values for A and B. This approach only worked if A, B and C were used consistently. Thus, if A represented Employee Payroll Contribution, that concept needed to be used in exactly the same way in all rules in which A was used.

In the Eighties many organizations set out to build expert systems, software applications that had hundreds or thousands of rules and that could solve problems that human experts routinely solved. To develop such systems, expert system developers would sit with human experts for many hours, asking them to talk through cases, explaining what they looked for, and what conclusions they drew from what they observed. The human expert might not think in terms of rules, but the developers would capture the knowledge and render it as rules. [1]

In the late Eighties the work of the expert system developers influenced database practitioners and they began to think about modifying relational databases or creating object-oriented databases that could store both conventional data and rules. [2] In essence these practitioners wanted to take COBOL applications apart and separate the data from the rules that manipulated the data. They wanted to do this because it simplified the maintenance of the applications. When data and procedures were mixed

in one block of code, it was usually hard to modify. It was much easier, on the other hand, to be able to look at the rules used in an application and simply modify one of the rules.

The interest in expert systems began to collapse in the early Nineties. Some very good expert systems had been built, but they were very hard and expensive to maintain. Real human experts read books and attend conferences and continually revise their understanding of the world. If expert system applications are to remain current, they need constant revision. Scientists returned to the AI labs at university and resolved to learn more about learning in order to create systems that could learn and improve themselves.

For a short period, it seemed as if the software companies that had created expert system building software tools would go bankrupt. Many did. Most, however, mutated, and stopped focusing on very large rule based systems. Instead, they stressed that the tools could be used to build more mundane rule-based systems that could be used to solve everyday problems. A good example is ILOG, a company founded to create expert systems building software applications. When the expert systems market collapsed, ILOG rebranded itself as a company that offered specialized rule-based software, and was then acquired by IBM and incorporated into IBM's BPM package as its decision management component.

The demand for more mundane rule-based systems was driven by those from the database world who had pioneered what they began to term Business Rule systems. Business rule systems were initially designed to enforce the policies in an organization. Thus if the organization had a policy that it would only extend credit to individuals who owned their own home, one could write a rule:

If X does not own home, then X is ineligible for credit.

It turned out to be much simpler to build systems that began with clear cut policies. The knowledge inside the heads of human experts was unorganized, constantly evolving, and difficult to extract. The knowledge implicit in policy statements was easy to define and document. Compared to the problems they faced in trying to capture and maintain the knowledge of human experts, business rule software vendors could modify their tools and easily handle the business rule systems that the new generation of business rule people wanted to develop.

The organizations that were very interested in business rule systems were highly regulated organizations like financial institutions of all kinds and government agencies. Much of the work done that this type of organization involves applying policies and rules to determine eligibility, appropriate requirements, and so on.

This is largely where things stood in the early years of this millennium. Since then, however, business process management has grown in importance, and the rule field is in the process of mutating a third time. Everyone has always understood that lots of business processes involve decisions, and, increasingly, rather than trying to decompose a business process to describe decision steps, business analysts are

content to define the decision in terms of business rules. Inversely, as companies have developed rule sets, they have looked for a way of defining where specific rules are used, and have increasingly decided to define rule use in terms of business processes or activities that use rules. This effort is facilitated by software tools that can interpret a set of rules and arrive at a decision. It is made even easier if the BPMS vendors incorporate inferencing techniques right in their BPM software.

The initial business process software vendors were largely modelling vendors that evolved out of the software (and especially the CASE) tradition and used flow models to describe processes. In recent years the modelling vendors have been joined by business rule vendors, who have added process flow elements to their products but are primarily focused on offering process modelling tools that are especially adept at capturing processes that involve decisions. This tendency has been accelerated because process theorists are increasingly interested in modelling complex, dynamic processes – increasingly termed “case management” – that rely on rules as a way of describing how the process people approach dealing with complexity.

Note, however, that at this point we are moving away from the earlier simplicity that was introduced when database theorists sought to define business rules as rules defined by business policies. When you look at business processes, especially complex business processes, you certainly find some decisions that derived from policies, but you find other decisions that rely on the knowledge or expertise of specific employees. These employees may not be as knowledgeable as the human experts the AI vendors were focused on, but they certainly rely on knowledge they have in mind, and defining their approach to solving problems requires elicitation techniques that are closer to the techniques used by the knowledge engineers that developed expert systems than the database folks who derived business rules from specific policies. [1]

Imagine a process that involved granting auto loans. In this case one could probably rely on existing policies and well established guidelines and write a small rule base that could easily make the decision. (Most banks already have systems like this.)

Imagine, instead, a process that involves granting loans to small businesses. In this case one would probably want to work with some good branch managers who have routinely done this successfully. One would find, as one talked through examples, that things like management skills and insights, local competition, and the nature of the organization’s business model would get involved -- and they aren’t so much a matter of policy as judgment. Suddenly one is writing business rules that are going to depend on loan officers making observations and judgments.

Thus, today, we have two different ways of approaching business rule systems for use with processes – a top-down approach that logically derives decision rules from policies, and a decision-specific approach that derives rules from specific knowledge workers who have experience in solving the problems being considered.

As if all of this wasn’t complex enough, we also need to consider a third alternative. There is, today, a growing interest in Analytics -- in applying a pattern matching algorithm to a set of existing body of data to see if we can make useful predictions.

Analytics derives from what is called data mining, which in turn is derived from AI work in the 80s. This isn't a rule-based approach, as such, but it definitely generates decision support. In effect, it lets a decision maker identify correlations that he or she might not have noticed. Thus, for example, we may examine the data from a large number of loans to companies engaged in heavy equipment sales, and find that companies with a specific combination of characteristics nearly always got in trouble and failed to repay their loans. Thus, we may conclude that we should not make loans to companies with those specific characteristics. In effect, the Analytic system is doing what a human expert with years of experience does, relying on lots of experience to identify situations to avoid or embrace.

As we try to deal with all of this complexity, we begin to conceptualize the entire rules process as more complex than the approach that just derived rules from business policies. It feels more like the situation faced by expert system developers, when it was necessary to break a decision down into steps, define rules sets for each step, keep track of the knowledge we need to define for each rule set and specify who is responsible for maintaining each rule set. [3]

No matter the complexity of the decisions or the decision logic we are dealing with, there are other issues when it comes to automating rule systems. If the process was very simple and specific, one might imagine storing the decision rules with the process. In most cases, however, business rules that get used in one decision are also used in other decisions. It turns out that business rules, especially rules derived from policies, ought to be stored and managed independently of business process management software. This is particularly the case if we are using a rule-based or analytic software package that will analyze a rule or data set and provide the user with advice about the decision. In other words, the architecture of software systems designed to provide decision support for process-based decisions can be quite complex.

Alan Fish has an extensive background in IT, in Artificial Intelligence, and in rule-based systems development. He has written a book that addresses two broad issues: the use of rule-based systems to aid decision making and the problems in automating the use of rules. This is not a technical book: It does not provide detailed coding examples. Instead, it is a book for managers and business analysts that focuses on giving readers an overview of the topic. This book, and Fish, have been heavily involved in the OMG's effort to develop Decision Model Notation (DMN) and some of the ideas in DMN derive from Fish's simple DRAW methodology for defining decision making requirements implicit in a decision. [3]

I recommend this book for those who are new to rules and who are interested in how one uses rules in the context of business process work. This book is more current than earlier business rule books that only focused on deriving rules from policies, is well written and easy to understand.

---

Paul Harmon is the executive editor of [www.bptrends.com](http://www.bptrends.com)

### Notes

[1] If you want a good introduction to expert systems and the techniques used in their development, see *Creating Expert Systems* by Paul Harmon and Brian Sawyer. If you want a good book on how to acquire the knowledge needed to develop an expert system, see: *A Practical Guide to Knowledge Acquisition* by Carlisle Scott, Elizabeth Gibson and Jan Clayton. Both books are now out of print, but you can still get copies from Amazon.

[2] The work on database-oriented business rule systems, as it was promoted in the early years of the millennium, was initially derived from work undertaken by IBM's GUIDE users group in the 1980s and later captured in the Business Rules Manifesto issued by the Business Rules Group (BRG).

[3] The new uses of business rules and analytics has led the Object Management Group (OMG) to begin working on a new way of approaching decision points in processes, which is broadly referred to as Decision Management. Specifically, the OMG has a team working on developing a Decision Model Notation – which I described in a recent Advisor, Processes and a Decision Model Notation, on December 18, 2012.

#### **BPTrends LinkedIn Discussion Group**

We created a BPTrends Discussion Group on LinkedIn to allow our members, readers and friends to freely exchange ideas on a wide variety of BPM related topics. We encourage you to initiate a new discussion on this publication, or on other BPM related topics of interest to you, or to contribute to existing discussions. Go to LinkedIn and join the [BPTrends Discussion Group](#).