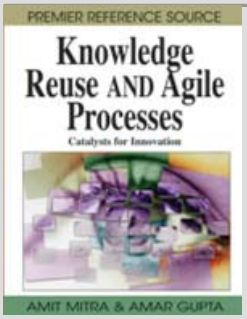

Knowledge Reuse and Agile Processes: Catalysts for Innovation

Amit Mitra and Amar Gupta
Information Science Reference
2008
$180.00 410 pages

Reviewed by Paul Harmon

Amit Mitra, a consultant and former Chief Methodologist at AIG, and Amar Gupta, a professor at the University of Arizona and MIT, have coauthored three books:

- *Agile Systems with Reusable Patterns of Business Knowledge* (Artech House Press, 2005)
- *Creating Agile Business Systems with Reusable Knowledge*. (Cambridge Press, 2006)
- *Knowledge Reuse and Agile Processes*. (Information Science Reference, 2008)

Together these books describe a knowledge-based approach to business process management. In essence, they conceptualize the company as a huge expert system and describe how the concepts and business rules used by the company can be captured and automated.

New technologies, like new ideas, take time to become established. When they are first presented they are met with a mix of enthusiasm and skepticism. Once tired, if success is not immediate – and it hardly ever is – those who opposed the innovation are quick to point out that they said the innovation would never work. Later, after the idea and the culture have had time to get to know one another and the new idea or technology is understood better, it often begins to flourish.

The idea of describing business processes as knowledge networks and sets of rules began in the Eighties with what were then called expert or knowledge systems. The first expert systems used rules to capture the knowledge of business experts and then made that knowledge available to other experts by putting the rules into a software system that, given information about a specific problem, could make an expert-level recommendation. As the early expert systems got larger, it was determined that rules alone were too clumsy. Hence, by the mid-Eighties, most of the more sophisticated expert system-building tools incorporated objects (they were called Frames in those days).

In essence, the objects in a sophisticated expert system-building tool formed a network that described the vocabulary of the problem, and rules were added to reason about the facts as they were accumulated by the system. When one used these more sophisticated expert system-building tools, one began by accumulating knowledge from experts. Thus, if one wanted to build an expert system to assist with home loans one would begin by working out the vocabulary of loans. One would probably identify vocabulary objects like: Home, Payment, Credit, Interest, Calendar, etc. Payments would probably have attributes like down payment and monthly payment, while Credit might have attributes like income, credit history, etc. In other words, one would construct a cognitive model of all of the concepts or words that a load officer typically used. Questions, in effect, that the loan officer would ask. Then one would begin to add rules that could reason with the information one had about a specific case. For example:

If the individual's credit history was superior, and her salary was $130,000 a year, and she could make a down payment of $50,000, what type of loan would she qualify for?

The objects and rules formed an abstract model of the concepts and rules that a human expert could use to organize knowledge about a particular subject and to reason about it to reach conclusions.

By the end of the Eighties, most companies had given up on expert systems. They concluded that expert systems could be built, but that the knowledge in the systems degraded too rapidly. One could capture human expertise in an expert system, but the system quickly became obsolete. Real human experts are constantly learning, reading journals, talking with colleagues about their latest experience, and attending conferences. They are constantly updating the knowledge structures and rules they use to analyze and solve problems. Thus, the problem with expert systems wasn't in the construction but in the maintenance. It was easier to keep the expert, because the system that would replace him required that you keep him anyway, to maintain the expert system.

This might have been the end of the idea that rules could be useful, but, in fact, it was only the beginning. Individuals who had learned about rules while building expert systems quickly realized that they could build systems to capture and automate more mundane human decisions – those based on well-defined corporate policies. Policies and associated business rules were easier to capture and changed less frequently. Thus, the interest in expert systems, in the Eighties, mutated into an interest in Business Rules in the Nineties and that application of rule technology is now flourishing. Many financial companies have large business rule groups that are responsible for defining and managing the business rules used throughout their organizations.

At the same time that the business rules movement was showing how business rules could be used in practical situations, others were exploring patterns, business processes and automated software tools that support business process modeling. Today business rules and business processes are being integrated in new and creative ways.

Amit Mitra and Amar Gupta propose to apply what I think of as a mixture of the expert systems approach to business process modeling and to the now popular business rules approach. In essence, they would build object models that described the vocabulary and business rules of an area of business – say Financial Management. If one then sought to create a business process in the area of Financial Management, one would, in essence, create process objects that would inherit information from the more generic Financial Management model. Mitra and Gupta refer to their high-level constructs as reusable patterns of business knowledge. They have written three books to provide a comprehensive explanation of their approach. This is the third.

In the first, *Agile Systems* they proposed a Universal Pattern that includes objects like Event, Fund, Energy, Physical Object, Person or Organization, Place and Information. They work out the basic attributes of these objects and define some of the rules or constraints that apply to them. Then they start to create sub-models for more specialized business activities. They consider, for example, a shipment and transportation cluster, a document and information cluster, a task-resource cluster, a meeting and agreement cluster, and a buying and selling cluster.

Mitra and Gupta went on to propose that companies consider creating a knowledge machine. In essence, it would be a huge expert system that had all the knowledge of all the terms used by businesses and all the critical constraints or business rules. Anyone with a specific process

problem would define the process, determine what elements of the process inherited what vocabulary, and instantly get an analysis of all the considerations and rules that might apply.

To provide a foundation for the knowledge machine, Mitra and Gupta have explored all the technical problems one faces in creating this type of inheritance hierarchy. This kind of system cannot rely on the simple inheritance one finds in simpler object-oriented languages. It requires that one object can inherit from multiple parents, and that some objects can inherit some features but not others from a given parent. These are programming problems that bedeviled the expert systems designers in the mid-Eighties and they still create technical and conceptual problems today. I mention this only to suggest that this book is not light reading. It not only offers a survey of the high-level vocabulary and concepts of business, but a survey of some very complex programming concepts as well.

The second book, *Creating Agile Business Systems with Reusable Knowledge*, discusses the underlying ideas that form the foundation of the earlier book. This book probed the truly fundamental concepts involved, including the nature of reality and business, the nature of objects and attributes and the meaning of domains.

Knowledge Reuse and Agile Processes, the third in the series, describes how the underlying concepts described in the second book can be transformed into the business patterns described in the first book. Thus, the books weren't published in what would seem to be the logical sequence, but now that all three are available they can be read in whatever order the reader prefers. I found it easier to begin with the first book, which shows how everything fits together to create a business system, and then to work back into the underlying theory once I understood why I would need it. Thus, I'd recommend reading the 2005 book first, the latest book, second, and then the 2006 book, if you want to go into the fundamental, underlying logic of the system.

No matter were you begin, the journey will be hard. It will also be rewarding if you really want to understand the potential for systematic, rule-based business systems analysis.

No company that I am aware of today has the discipline to implement Mitra and Gupta's vision. Today's companies don't understand themselves well enough to know all of the concepts and rules they use in all their processes. Besides, the maintenance problems that bedeviled expert systems are still insurmountable. What Mitra and Gupta provide, however, is a description of a world that might exist in 20 years, after many technical problems have been solved. We all know that companies will continue to automate and that tomorrow's companies will rely on automation technologies that have yet to be invented. Some companies are already on this journey. Several financial institutions have teams of hundreds of people working to define, organize, and manage their organization's knowledge, policies, and business rules. For those interested in such a journey, this book, and the two others in the series, provide a magisterial vision of the goal they are working to achieve and a comprehensive description of the current state of the technology on which they must build.

Paul Harmon is the executive editor of BPTrends. He is the author of *Business Process Change* and has coauthored several books on expert systems and object technologies.

