

Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done

Keith D. Swenson (Editor)

Meghan-Kiffer Press

\$35.95 333 pages (\$26.95 for Kendel)

Reviewed by Paul Harmon

For several years now, business process theorists have been concerned with describing the difference between processes that are more-or-less procedural in nature and slow to change and those processes that either change frequently, are very complex and difficult to describe, or both.

A good example of a procedural process that is slow to change is a production line situation where each assembly worker's job is precisely defined.

A good example of a complex, dynamic process is a process that generates proposals for major engineering undertakings. The process begins when the engineering firm receives a request for a bid. Someone at the firm analyzes the request to determine if the firm is even interested in bidding. Assuming the request is something the firm is interested in, a team is assembled to analyze the problem, design a solution and generate a bid. In the course of the project, members of the team may send emails to colleagues around the world to find out about problems with similar jobs, to learn more about the needs of the company making the request, and to gather information about technologies that might be used. Similarly, there may be many meetings in which issues are argued out, solutions are discussed or the language of the final proposal is discussed. The proposal, when it is finally prepared and submitted may have characteristics in common with other proposals the engineering firm has submitted in the past, but it is also a unique response to a unique proposal. In other words, the response to the request was treated as a unique case. The approach and activities undertaken were adapted to the unique needs of the client and the skills of the team assembled to generate the proposal. And the entire effort was managed, at least in part, according to unique criteria associated with the specific request.

Historically, consulting firms have always used an approach more-or-less like the one just described. As other organizations offer more options and tailor their options to customer demands, they have also begun to introduce more flexibility into their processes. Similarly, as organizations rely more on knowledge workers who add value to services by refining them as they interact with customers, process analysts have been challenged to figure out how best to describe and specify improvements for complex, dynamic, knowledge-based processes. A variety of names have been proposed to describe these processes. Case Management has its roots in hospitals and insurance companies where the term reflects the idea that each patient's case needs to be considered as a unique case and that each insurance claim, is, to some extent, a unique case that needs to be carefully examined to determine which rules apply.

Theorists like Keith Harrison-Broninski have been discussing complex, dynamic processes for several years. The Object Management Group (OMG) began to discuss this type of process modeling problem in 2009 and has been using the term Case Management to describe the approach they are trying to define and standardize. I would prefer that they use something a little more descriptive, like "dynamic, complex processes," but can certainly use Case Management if

that's what the community settles on.

My personal perspective on complex, dynamic processes is very much influenced by my decade of work in AI and Expert Systems, where we focused on modeling human expert behavior with rules. My background leads me to use the idea of rules as a way to define complex behaviors. Before I discuss that, however, let's step back and consider Figure 1.

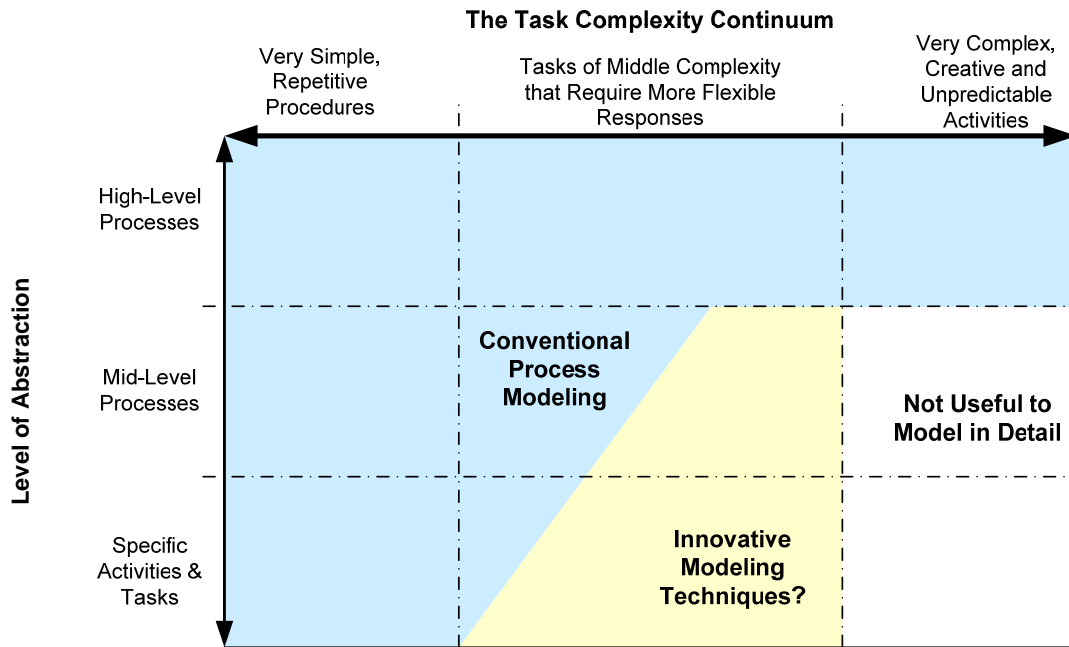


Figure 1. Levels of Abstraction and Task Complexity

I consider both abstraction and complexity to make a point about modeling. At a high-level of abstraction we can model any process. Consider the development of a new product (an auto design, a new drug, a patient diagnosis). At a high level we can show the steps that the developer goes through. What we do not want to attempt is to drill down into the details. Figure 2, for example, provides one possible description of a high-level medical problem diagnosis process. At this level of abstraction we can define the steps the physician should go through.

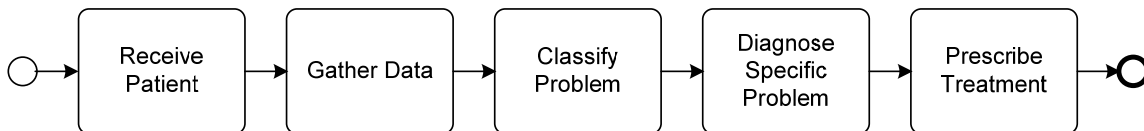


Figure 2. Medical Problem Diagnosis Process

We could easily decompose some of the processes described in Figure 2. We could, for example, spell out some of the steps a physician might go through to gather data or to classify a problem. But we will probably not want to drill down into the details of diagnosis. It's simply too complex and patient specific.

The problem in drilling down into a diagnosis is not that there are alternative possible paths. If it were simply a matter of choosing between alternative paths, it would, in my opinion, be a problem

of middle complexity. This is the kind of thing we ask medical technicians to do, and which we can, when required, define in a step-by-step manner.

Research into cognitive psychology and expert systems suggests that human experts take years to learn how to solve certain kinds of problems, and, in the course of those years, they learn tens of thousands of rules. The Stanford team that built Mycin – an expert system that could diagnose meningitis problems – ended up incorporating over 12,000 rules. These rules did not form a neat decision tree. Their use depended on an inference-based rule engine that searched the rules dynamically. (Put in computer terms, the problem space was too large to search with algorithmic techniques.) Given one fact, rules would be called that asked other questions and gained additional information. The development of an expert system required working with physicians and going through hundreds of cases to identify all the rules required. Even then, Mycin would not usually provide a single answer. Instead it would generate a list of the possible causes and rank them according to consequences.

Based on this experience, my definition of very complex processes are processes that it would take tens of thousands of rules to define and that would result in multiple possible solutions among which humans would ultimately need to choose.

To make matters worse, we don't use Mycin today. The domain of meningitis diagnosis is too dynamic. New diagnostic heuristics and new medicines are constantly being introduced. A meningitis expert routinely reads medical journals and attends conferences. In the process the expert is constantly changing some rules and adding others to his or her knowledge base. It turns out that Mycin is too expensive to maintain. Even if we can build such an expert system – which the Stanford team did and Mycin briefly performed as well as meningitis specialists – we can't afford to maintain it. Until we can create software systems that can learn in ways we can't quite specify at the moment, it's easier to rely on human physicians. And the same conclusion applies to business executives, new product designers, the people who create ads or devise marketing campaigns, and the architects who design new software systems.

Adaptive Case Management

Now let's shift and consider Adaptive Case Management (ACM), which is the topic of the articles collected in *Mastering the Unpredictable*. (The acronym is awful, of course, as it will continually be confused with that of a major computer society.) As the authors define "Adaptive Case Management," it is an approach for capturing and automating the work that knowledge workers do. In the terms described in Figure 1, it is the light yellow area shown on the map. In essence, the authors propose a systematic approach for the description and capture of processes undertaken by knowledge workers – processes I would suggest range from a few hundred to a thousand rules.

Mastering the Unpredictable is a book of readings and the contributors include many people who have been involved in the OMG effort. The contents of the book is as follows:

Introduction – Nathaniel Palmer

1. The Nature of Knowledge Work – Keith D. Swenson
2. What to Do When Modeling Doesn't Work – Jacob P. Ukelson
3. Moving from Anticipation to Adaptation – Tome Shepherd
4. Technology for Case Management – John T. Matthias
5. The Elements of Adaptive Case Management – Max J. Pucher
6. Data Orientation – Dana Khoyi
7. Templates, Not Programs – Dana Khoyi and Keith D. Swenson
8. Healthcare – David Hollingsworth

9. Improving Knowledge Work – Frank Michael Kraft
10. Innovation Management – Henk de Man, Shiva Prasad and Theodoor van Donge
11. Achieving Agility – Dermot McCauley
12. The Next Evolution of Continuous Improvement – Caffrey Lee and Julie Miller
13. Historical Perspective – Keith Swenson

As with any book of this type, some chapters are better than others. Chapters 5, 6 and 7 form the heart of the book and should be read by anyone interested in the future of BPMS and process automation.

The authors often contrast what they are recommending with BPM, which they say begins with and focuses on processes (procedural sequences). In essence, they are contrasting BPMS tools based on Enterprise Application Integration and older workflow approaches and their approach which depends on dynamic planning, “Tasks” (“templates”) and rules.

Let’s begin with the idea of tasks or templates, as these terms are used by the authors. A “case” is something that you want to accomplish. As the authors are using these terms, a process is a sequence – as I pictured in Figure 2 above. Each subprocess gets done in a specific order. A “task” is one or more activities that need to be accomplished to complete a case, but its use or its order can’t be determined until we know the specific case. Thus, instead of a flow plan, the knowledge worker about to undertake the specific case considers a list of tasks and decides which he or she will use for this specific case, and in what order the tasks will be attempted. In other words, one of the first tasks in the case involved planning the tasks and tentative sequence for the specific case. Note that this places a limitation on automation – ACM applications are designed to support knowledge workers, not replace them. For any given case, only a subset of the tasks may be employed. The structures of the tasks themselves are largely based on the use of rules. The rules used, however, are mostly derived from knowledge workers, however, and not from organization policies.

Stepping back a bit, we are seeing an effort to reestablish some of the concepts used with expert systems development in the 80s. Instead of structuring the approach around a flow (procedural) we are going to structure the approach around knowledge concepts (data structures) and rules – a declarative approach. Our concern, in almost all the examples provided in the book, is not with accomplishing a task, but in reaching a decision or defining a solution. One might suggest that BPMS vendors began with procedural techniques, then began to add rule-based techniques. Adaptive Case Management suggests how the rule-based techniques might take over and provide developers with tools that make it easier to model and automate knowledge structures and knowledge-based tasks.

This is an important book. It does not provide the kinds of concrete examples one might like, with detailed discussions of how a specific set of cases might be processed, but hopefully that will come in a subsequent volume. What this book does is define the fundamentals that might be used to develop what I think of as rule-based or agile workflow systems.

This is a significant step beyond the more or less independent business rule approaches that have been popular in the last decade and represents a return to knowledge-based techniques that predominated in the Eighties. It is a direct result of some very serious thought about how rule or knowledge-based techniques can be used to help model and automate complex, dynamic processes.

The contrast several authors set up between BPMS and their ACM approach is dramatic, but probably not as significant as they think. Consider Figure 1 again. Lots of good enterprise work can be done with high-level processes that will be completely compatible with the use of ACM

techniques at more detailed levels. ACM is not an alternative, but a set of tools that can be used on one set of problems that process analysts face.

I do not believe the techniques described in this book can be scaled to deal with really complex problems – for the same reason that expert systems failed – because the rule maintenance problems would be too expensive. I do believe, however, that the time is right to apply these techniques to extending BPMS tools for use with processes that include tasks that depend on knowledge. Moreover, as these applications illustrate, the tools only work if there are knowledge workers to plan each case and adjust the tasks and choose among the options offered by the ACM tools. In other words, we are always talking about a Decision Support tool here rather than a fully automated solution. Being stimulated to think about how this integrates with today's popular BPMS applications is worth the price of this book.

I recommend caution, however. This book will help you with process problems that lie in the middle of Figure 1, and as such, it's a major advance. It will not, however, prepare you to tackle problems that lie on the right side of Figure 1. Use these new techniques to explore the middle of the figure and avoid projects that lie on the right and would be very likely to fail, either initially, or when their maintenance proved too costly.

I strongly recommend this book. Others will come out with different ways of dealing with knowledge-based processes, but this is a very good start and suggests an approach that will certainly be rapidly developed in the year ahead. This book will not prepare you to build an Adaptive Case Management system, but it will certainly give you lots to think about.

Paul Harmon is the executive editor of www.bptrends.com

BPTrends LinkedIn Discussion Group

We recently 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**.