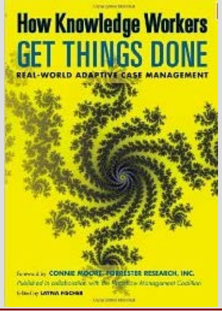

How Knowledge Workers Get Things Done: Read-World Adaptive Case Management

Layna Fischer (Ed.)

Future Strategies, Inc.
$44.96 233 pages

Reviewed by Paul Harmon

For many years now, Layna Fischer has been editing an annual review in conjunction with the Workflow Management Coalition. Each focuses on a specific area of concern within the overall business process/workflow field. In 2012 she edited a review entitled *Taming the Unpredictable: Real-World Adaptive Case Management: Case Studies and Practical Guidance*. This year she returned to the same topic, and has put together a book on Adaptive Case Management in the context of Knowledge Workers.

Adaptive Case Management is a growing process niche that focuses on how software vendors deal with dynamic, complex processes. Imagine a process that involves lots of employees interacting with each other around the world via email and smart phone messages to arrive at a proposal to submit to a client. Or imagine the process that a team of physicians goes through to determine what problem a new patient has and what treatment will be most appropriate for that individual. In essence, each case varies.

I reviewed the earlier book, *Taming the Unpredictable*, in 2012, and found it excellent. The field is new and rather small, however, and I was a bit surprised to see that Fischer and the WfMC decided to return to the same topic this year. Having read the book, I think the decision was justified. Much has changed, and those working in the Case Management and ACM area are learning and refining their views rapidly. This year, for example, there are several articles that focus on subtle differences between different types of Case management. At the same time there are some good case studies that illustrate what all this means in actual practice.

Let me simply look on one area of concern. When most readers think of an ACM tool, they probably think of a BPMS suite, with a facility for dealing with dynamic situations or complex communications between employees. IBM’s latest BPMS offering, for example, includes 5 facilities, one of which is a Case Management facility. In *How Knowledge Workers Get Things Done,* there are several examples that claim to be ACM, because they allow either the employee or the customer to respond to the situation in a flexible manner. Consider an extreme case: Is Excel an ACM product? Say I have a process that requires a manager to prepare an annual budget. I could offer a process model that described what the manager was to do, step-by-step, or I could offer the manager an Excel package, and let the manager work up his or her own budget in a spreadsheet. The latter is very flexible and can accommodate almost any kind of budget the manager wants to create. On the other hand, it will probably drive someone who is responsible for pulling all the specific budgets together into a common budget a little crazy, since each manager will do their budget in a slightly different way.

In the book, no one cites Excel as an ACM application, but several talk about making a software application that one might use, for example, to calculate a loan repayment plan, available to customers so they can calculate their own repayment plans. Do we really want to say that making an application available to customers is a Case Management process? It is more flexible, and it's harder to diagram as a process, since we don't know exactly how the customer will use it, but is it what we mean by Case Management or ACM.

When I first began to examine Case Management, I proposed the following figure as a way of understanding the field.

Figure 1 reflects my personal perspective which is derived from two decades of work with Artificial Intelligence and Expert Systems, where we primarily focused on modeling human expert behavior with rules. My background leads me to use the idea of rules as a way to define complex knowledge, and to put that on one axis. On the other axis, I consider the nature of the tasks (the level of abstraction) you are trying to model.

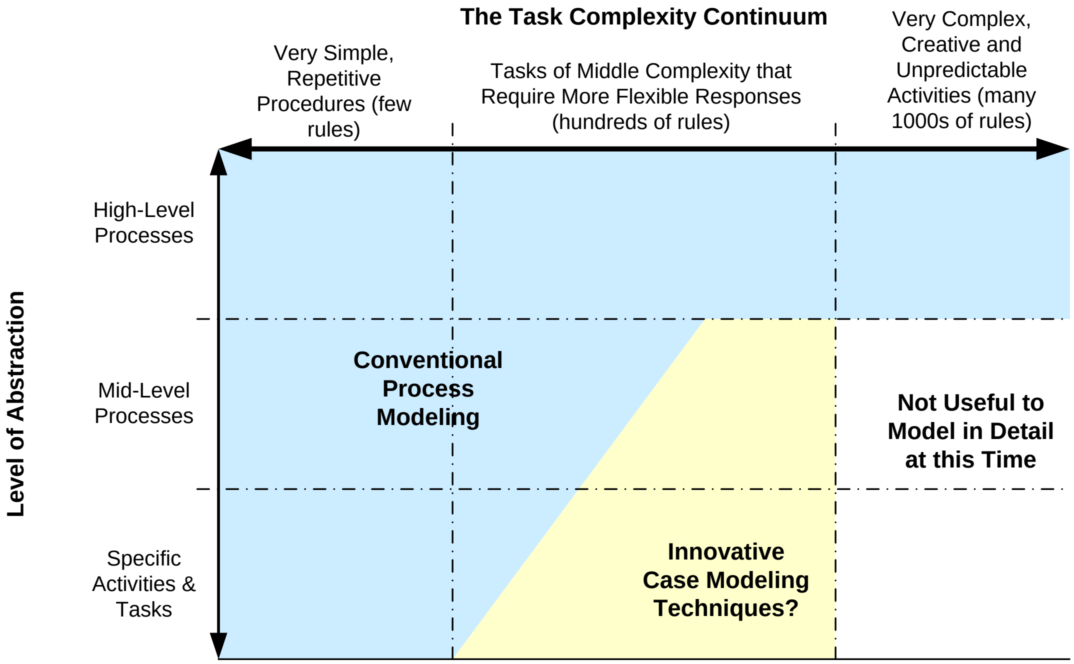
The Task Complexity Continuum
Very Simple, Repetitive Procedures (few rules)
Tasks of Middle Complexity that Require More Flexible Responses (hundreds of rules)
Very Complex, Creative and Unpredictable Activities (many 1000s of rules)
Level of Abstraction
High-Level Processes
Mid-Level Processes
Specific Activities & Tasks
Conventional Process Modeling
Not Useful to Model in Detail at this Time
Innovative Case Modeling Techniques?

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.

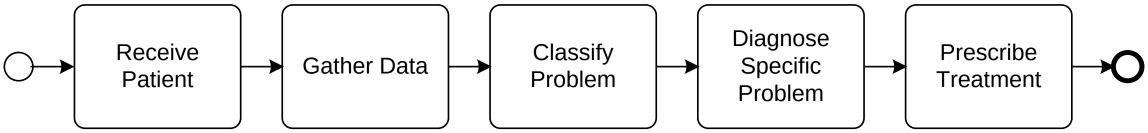

Figure 2. Medical Problem Diagnosis Process

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 and that humans would ultimately need to choose between.

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 it's too expensive to maintain Mycin. 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.

I review this discussion to point out that the authors in this year's book on *How Knowledge Workers Get Things Done* are starting to run into the problems I described. Several seem confused about whether they can or can not model a Case Management problem. The answer is that they can do it at a high level, but don't want to try to drive down into the details.

Keith Swenson, in his article, proposed a differentiation between Production Case Management and Adaptive Case Management – a distinction he claimed to observe as he read the various submittals to this year's book. In essence he is trying to sort out the difference between what I called tasks of mid-level complexity and very complex tasks. As I predicted, considerable progress is being made on dealing with Production Case Management and less is being made at the more complex end. To be fair, Keith is looking at a lot more variables than rules, reflecting the concerns of the emerging field. Thus he considers if social messaging is involved, if experts know a great deal, if customers are being asked to help solve problems, and so forth.

Another emerging issue – closely tied to the emphasis on knowledge workers – involves the degree to which feedback is used to change the process as it is performed. In essence any manager who looks at the quality of outputs and then provides employees with feedback on how successful they are is providing feedback. Building it into the process, as any professional does when they consciously observe their own performance, and adjust it, is clearly a key element in creating more agile processes that can respond to changing situations. This doesn't require a new approach to process modeling, although software tools can help facilitate getting feedback to the right performers, but it requires thinking about the human performance elements of the process. It's something that has been done for years by those working in the Rummler-International Society for Performance Improvement (ISPI) tradition of Human Performance Technology, and it is being brushed off and given new attention by those who are thinking about how to design Case Management applications.

As I suggested, the field has advanced and many new topics and approaches are being considered (or reconsidered in some cases). The case studies provide the best view of the actual state of practice, and predictably lag behind where the thought articles are. A couple of the cases studies reveal vendors too eager to re-label their software Adaptive Case Management software, but all the cases underline the fact that companies are trying to capture more complex and dynamic processes than they tried to model in the past, and, in many cases, producing advances of real value to their organizations. I especially liked the Cognocare healthcare example and the system developed by the Norwegian Food Safety Authority, but they were all worth reviewing.

I recommend that those who are interested in dynamic processes and want insight into the latest thinking and techniques available for modeling and improving these processes buy this book. Two or three of the articles, including those by Harrison-Broninski, Nathaniel Palmer and Keith Swenson, or the case studies I mentioned are well worth the price of this collection.

___

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

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.**