Business Process Management: The Next Wave: Harnessing Complexity with Intelligent Agents

By Jim Sinur, James Odell, Peter Fingar

Meghan-Kiffer Press, 2013 $50.67 ($48.14 Kindle) 276 pages

Reviewed by Paul Harmon

Readers who have followed what I have written for the last 2-3 months know that I have spent a lot of time thinking about case management, dynamic business processes, and the need to capture and integrate events and data via business process models. Nor am I alone in this. In one case I cited a series of papers by IBM researchers on data and artifact-centered modelling. In another case I cited the OMG's recent work on their CMMN standard for case management notation. All of this, broadly speaking, relates to the fact that organizations are trying to describe more complex, dynamic business processes.

In the case of CMMN, for example, the OMG case management task force suggests that we give up on conventional process flow modelling and begin to think of processes as collections of "tasks" that can be assembled, dynamically, as the process is executed. This is hard to think about if you are the analyst trying to understand the process to be modelled. (I showed one of the early diagrams suggested by the CMMN team in my November Column, but developing such a diagram raises more problems than it solves.) Dealing with rapidly evolving or networked process is also hard to think about if you are the IT developer trying to think about how to build a software tool to support this kind of modelling, let alone the subsequent dynamic execution of such a process.

A new book, *Business Process Management: The Next Wave: Harnessing Complexity with Intelligent Agents* has recently been published to address some of these issues. The book is written by a three authors, Jim Sinur, who has recently retired from Gartner, where he was their senior BPM analyst, James Odell, a long time Agent guru, and Peter Fingar, a management writer who co-authored, with Howard Smith, the book *Business Process Management: The Third Wave*, the book that launched the current interest in BPMS applications.

The book is about *Intelligent Agents*, an approach to software development that encapsulates a set of software functions in an object. The term *Agent* refers to the fact that the encapsulated software functions act as if they had goals of their own. Intelligent Software Agents, as a concept, was developed by AI researchers in the Eighties. One of the nicest examples of the use of agents was one I witnessed, in the mid-Eighties that used fighter plane agents. In effect, there was a software agent that contained all the information about what a fighter plane did – it flew a mission and when it encountered enemy planes, it attempted, in coordination with any of its fellow fighter planes that were in the area, to shoot the enemy planes down. Most of the functionality of the fighter-plane-agents was provided by rules embedded in the agent. There was also an inference engine to manage the execution of the rules.

In the demonstration, the programmer placed a number of “instances” of the fighter plane agent on the screen, gave each its allegiance and its own position in space, and then started them up. The fighter plane agents each sent out messages to determine who else existed in their space, and then proceeded to work with their own planes to try to kill the enemy planes. In effect, they undertook a dogfight. There were two versions of this, one running on a single machine, where agents were threads, and one running on a multi-processer computer where each agent had its own CPU. In the latter case,. the fight was very fast and very realistic. (AI and agents are currently very popular in software game development.)

Increasingly, we are designing very dynamic processes that rely on lots of individuals working together to solve problems. In many cases, each different individual is located on a separate computer, on an iPad, or on an iPhone, and only occasionally actually linked to the computer that is monitoring the overall execution of the process. Imagine that we created an agent for each participant. For example - when a physician was asked to consult on an instance of a heart surgery diagnostic process, he or she would launch an agent, with specific goals and instructions. The agent would watch the ongoing process, and alert its owner, when he or she might be interested in a new development. In some cases the agent would know to request information, or to inform others that its owners would like time to make an input.

Imagine another situation - you are a customer and want to look for a buying opportunity. You launch several agents on several sites to look for the product or service you are interested in buying, and the agents either act for you and buy the product when the opportunity arises, or they notify you, as they were instructed.

Or, imagine a process like the ones that are being conceptualized by the case management (CMMN) folks, where initiation of the first subprocess starts by defining the problem and then assembling a number of available tasks into a unique process to deal with the specific case. Assume I decided that I would assemble 10 tasks into a new process. What order would they take, and what would each do? It would be nice, from a software developer’s perspective, if each of the tasks knew what it was to do and proceeded to undertake work more or less independently of other tasks, especially in cases were individuals were involved who would need to be contacted via their own computers or smart devices.

Business Process Management: The Next Wave, doesn’t seem to be a very well focused book. Peter Fingar spends his time providing an overview that is a little too vague. Jim Sinur focuses on an interesting, but tangential, model of how companies respond to change. The heart of the book, and a very nice set of appendices, were presumably written by Jim Odell, and describe agent technology.

Unfortunately, none of the authors really address what I consider to be the key issue for those of us in BPM: How do you identify and model dynamic processes. Within the agent discussion, BPMN diagrams are used. At no point does anyone discuss the OMG’s work in CMMN – which would actually interface with agents rather well. No one considers what would be involved in defining a process in which a physician might want to have an agent represent him in an ongoing heart diagnostic process. Clearly the physician wouldn't design a software agent. The business analyst probably wouldn't either, but he or she would at least need to know how to recognize the appropriate circumstances, and how to define appropriate software requirements.

The authors do not use terms with the precision I would like, so let me be clear - one concern is how business or process analysts document and analyze business processes (BPM). Another concern is creating and using software tools that might provide automation for processes, once we understand them (BPMS). If there is a "next wave" in BPM, it will involve a rigorous approach to describing and analyzing dynamic business processes. In fact, that is a work in progress, only being done by leading edge organizations, and there is, as yet, no consensus on how to do it.

A second concern is how to design and use software tools to support the analysis, design, and execution of dynamic processes. If agent technology has a role to play in the evolving world of dynamic BPMS tools, then it is as a software technology.

The book, *Business Process Management, The Next Wave*, is about "the next wave," only in the sense that we are entering a period when we are going to have to expand our knowledge of business problems to deal with more dynamic situations. Analysts are going to have to capture not only flows, but semantic networks and knowledge rules about the decisions that will need to be made during the execution of the business process. New BPMS products will eventually be offered that will support the documentation of these more complex business processes.

Business Process Management, The Next Wave, talks, breathlessly, about the huge change that is occurring. In fact, there is a change taking place, but it is very gradual, and only beginning to gather momentum. And, in most cases, it will not involve the business people involved in documenting business processes nearly so much as it will impact those trying to create software to support those processes. It is the software developers who will be faced with creating systems that can deal with frequent input from analytic systems or from customers or employees who are interacting via distributed computers or intelligent phones. It is the BPMS applications that will have the task of assembling the tasks that will make up a process, given a specific set of circumstances.

I think the authors targeted this book wrong. It's offered as a contribution of BPM and it's really a contribution of software design. This book doesn't tell us much about the nature of the new process problems, or about how to analyze them. It doesn't introduce a new notation, or talk about the problems of managing or measuring dynamic processes. Instead, it talks about a software technique that lets software developer's package rules and data in ways that makes it possible for them to serve as agents.

Are agents, in fact, the best way to package the functionality needed for dealing with dynamic business processes? I don't know the answer. Nor does this book provide one, since it doesn't treat the issue broadly, but focuses instead on describing only one specific software design approach – agents. (It would have been interesting to hear from someone actually at work on the design of one of the leading BPMS tools. Are any of the BPMS tools actually using agents?)

Information about agent technology has been around for quite a while. Jim Odell is one of the best authors on this subject, and this book represents his latest summary of the field. If you are a BPMS software tool designer, you should probably read this book to see what agents might be able to bring to new BPMS designs. If you are a business process analyst, you might want to recommend this book to someone in your IT department who is focused on new software design.

If you are a business process analyst, however, waiting for some help in conceptualizing how you might approach the analysis and redesign of case management or dynamic processes at your organization, this probably isn't the book you have been waiting for.

Paul Harmon is the executive editor of www.bptrends.com. He is the author of *Business Process Change*, and the chief methodologist of BPTrends Associates, a training and consulting company.