Harmon on BPM | Paul Harmon

Modeling Dynamic Processes

Case management, dynamic processes, iBPM. There are lots of names for the fact that many companies are trying to model processes that, by their very nature, keep changing. It's not as if these concerns were the dominant concerns at most companies. Indeed, in our recent survey, we asked how many companies were currently engaged in analyzing and developing this type of process, and only 2 % said they were. Still, more will be doing so in the future, and vendors are already working on software features that will make the analysis, modeling and development of dynamic processes quite a bit easier. So now is the time to begin to think about the nature of these processes and whether or not your organization ought to consider investing in case or dynamic technology when it becomes available.

The term *case management* which is probably the most popular term for dynamic processes, comes from medical practice, so let's use a medical example. A patient calls at an emergency reception area, or drops in at his doctor's office, with a problem. He becomes a *case*. If you imagine that there was an established process – Diagnose and Treat Patients – then, in essence, the hospital creates an instance (or case) of that process for the individual patient.

It's easy to imagine the high level process, which we've pictured in figure 1, using BPMN.

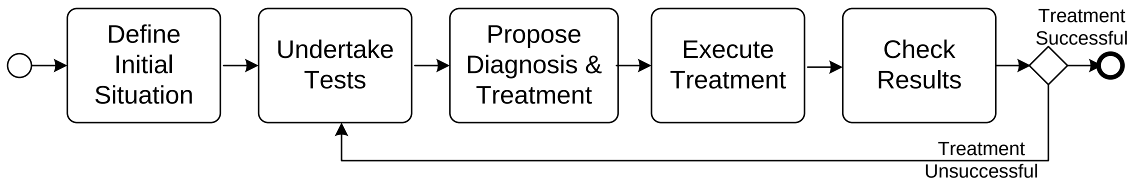


Figure 1. Diagnose and Treat Patients

Obviously we could refine the model shown in Figure 1. We could show a swimlane for the patient, and perhaps another for the lab when tests required specialists, or we could add adornments to indicate that each of the subprocesses shown in Figure 1 was undertaken by a person (a manual process), rather than being automated. Overall, however, for a variety of purposes, the figure shown in Figure 1 would give us a good overview of the process.

It's hard to imagine that anyone would think the process shown in Figure 1 is rigid or lockstep, in spite of its being rendered in BPMN notation. It's at such a high level of abstraction, and each subprocess could cover such a wide range of activities – from those appropriate for treating a heart attack to those used to deal with a broken arm to still others for treating the flu. If anything, surely the major complaint would simply be that it is vague. The process describes a generic approach to treating all medical problems.

So let's think about how we might refine the process in Figure 1. One way might be to introduce a branching point between subprocess 1 and 2. Something like what we show in Figure 2.

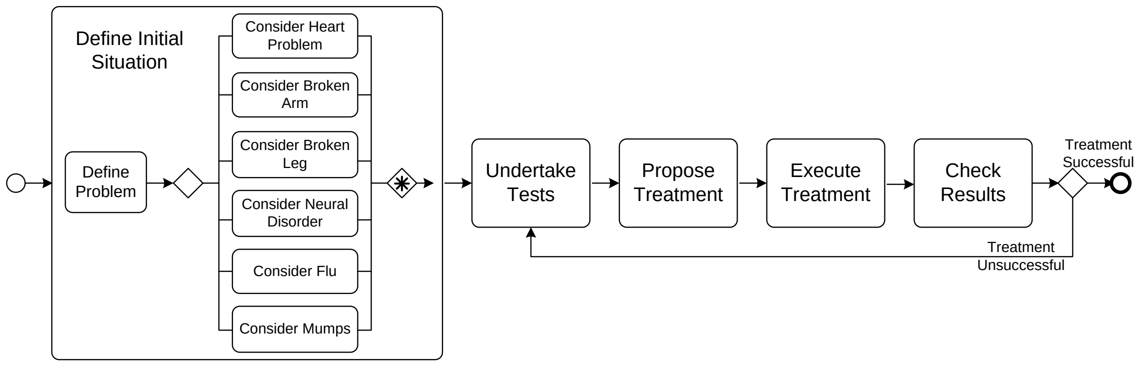


Figure 2. Diagnose and Treat Patients with some options shown.

Everyone can see what's wrong with the solution in Figure 2. We don't begin to identify the thousands of problems that an emergency care facility or a physician might confront when a patient comes in for help. We could obviously create a hierarchy of problems, and do the diagnosis in a series of decisions, as a botanist does, using a key when he or she tries to identify a plant. Still it would be impossibly complex.

Figure 3 represents a more elegant solution, but hardly improves on Figure 1. In essence, in Figure 3 we indicate that we will use business rules to make the decision during the Define Initial Situation subprocess. As shown, however, this is almost as vague as Figure 1. It would only become more concrete if we showed you the thousands of knowledge rules that we would need to actually make the diagnosis. Still, it could be done and it does represent a kind of solution.

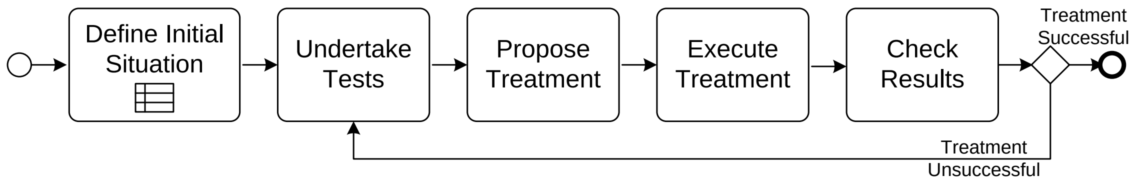


Figure 3. Diagnose and Treat Patients with an indication that a decision (e.g. business rules) will be made in the Undertake Tests subprocess.

Unfortunately, even if we could handle the decision in the Define Initial Situation subprocess, we would face another task, even more daunting when we tried to describe all the tests we would undertake, depending on the possible problems we identified in Define Initial Situation.

This is similar to the situation faced by analysts in the Eighties when they began to try to develop systems that could handle problems that human experts handled. They found that branching models with activities and flow arrows were inadequate.

The number required were simply overwhelming. Instead, expert system developers shifted to business rules, and then later to a combination of business rules and semantic objects. The objects described the knowledge entities that existed in the problem domain, and the rules applied logical reasoning to determine which objects were required for specific types of problems, and then used them further, to reason about the exact nature of the specific problem.

Expert system developers, being focused on where there were essentially declarative problems, never bothered to try to develop flow diagrams to describe the kinds of problems they were dealing with. They relied, instead, on diagrams of networks of objects and lists of rules that could be used to assign values to the attributes of the objects pictured in the networks.

A few years ago the Objet Management Group (OMG) created a task force to see what could be done to establish some standards in the case management area. The companies represented on the task force include Agile Enterprise Design, BizAgi, Cordys, IBM, Oracle, SAP, Stiftelsen SINTEF, Kofax, TIBCO, and Trisotech. In January of 2013, the task force released a Beta draft for the **Case Management Model and Notation (CMMN)** and have been working throughout this year to refine it. Undoubtedly the Beta draft will be changed before it is finally released for public comment. In the meantime, however, I thought readers might find it interesting to see how the OMG group is approaching the problem.

At this point the task force has suggested that the existing BPMN (2.0) is appropriate for defining lockstep processes and contrasted it to their CMMN approach that is appropriate for dynamic, complex processes, which they prefer to term *cases*. A case is represented by a file folder, with the name of a type of case on it. This makes a case diagnosis much more specific than the example we looked at in figures 1-3. The OMG team assumes someone walks into a doctor's office and announces that she has a broken arm, and the doctor needs to analyze that problem.

Next the OMG team assumes that a case involves a number of tasks, which are represented by rectangles with rounded corners--the same graphic that BPMN uses to represent a process or activity. And, although we won't go into so much detail in this article, the team assumes that one type of task could be a process.

Tasks are not connected by flow arrows. It is assumed that a given case includes many tasks, only a small subset of which might be used to deal with a specific instance of a case. (Imagine that the first subprocess in Figure 2 was a *case*, and each of the alternative possible problems was a *task*.) Some tasks do depend on others, and a light dotted line is used to link tasks. When you see the notation, you are to assume that the left or upper of the two boxes must be done before the right or lower box. (No arrow heads are used to show which is prior or subsequent.)

Some rectangles are bordered with a solid line and some with a dotted line. Those with a dotted lined are discretionary, and can be invoked at any time.

In addition, a diamond placed on the boarder of a box indicates that the task can be "triggered" by some set of circumstances. If you imagine this as being done by rules, then the diamond, which is termed *Entry Criteria*, describes the situation that would trigger the task.

Figure 4 pictures what is currently termed a *Case Plan Model.* Specifically, it's a Case Plan Model for Treat Fracture. We assume someone has arrived at a hospital with a fracture and the diagram below describes what the hospital might do. The small adornment on the top of the folder line with a grid and a minus sign is termed a Planning Table. The negative indicates that it is optional, but assuming it is used, it defines possible relationships among the tasks. In this case, a patient could either begin at the Examine Patient task, or at the Prescribe Medication task. Assume she began at the Examine Patient task. Depending on the diagnosis (resulting decision), the patient could either be given a sling, asked to get an XRay, or discharged.

There are two symbols for manual in CMMN. The hand is referred to as "non blocking" and means that another task could take place simultaneously – the doctor could examine and pause to administer a pain killing drug. The little person's head and shoulders is a blocking manual task. When that task is underway, no other tasks can be applied to that patient. There are lots of other adornments and I only mention a few. Obviously readers interested in the detailed notation will have to be members of the OMG to get the complete Beta at this time, but I am only interested at giving a flavor at this point.

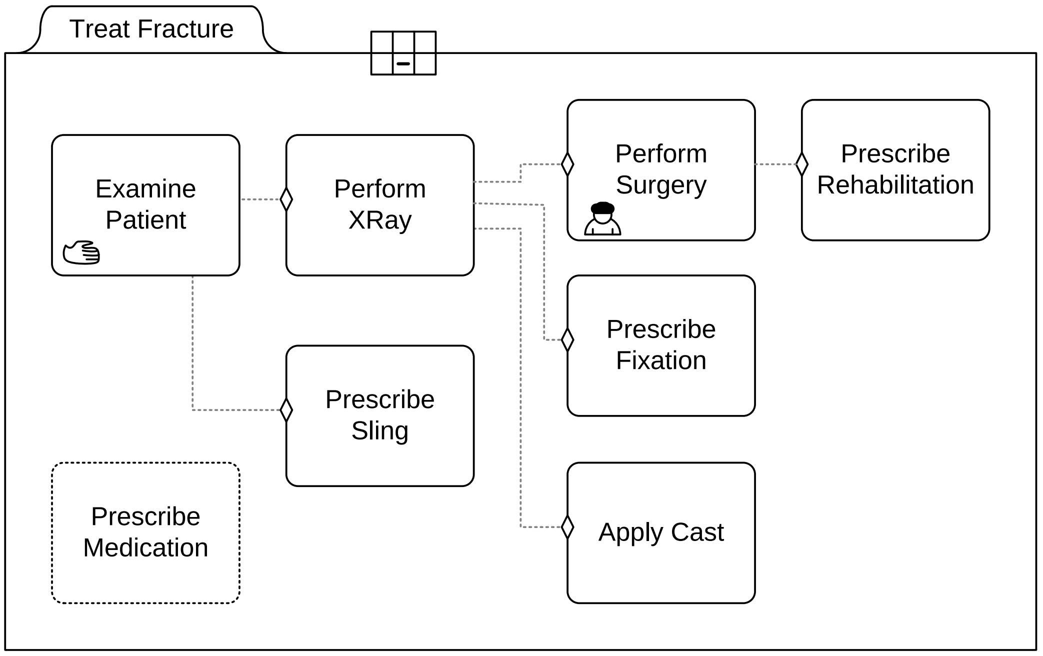


Figure 4. A Case Plan Model for Treat Fracture with several tasks and a option planning table.

Let's step back and see where the CMMN notation is at this point.

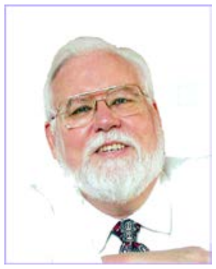
Clearly the CMMN team assumes that some tasks will be automated, but that many will be performed by human performers. (Expert systems were designed as software applications, but most assumed that a human worked with the application and made inputs and responded to questions to give the system the information it needed to work.) Something similar is envisioned in CMMN.

Rules (or Decision Management if you prefer) will be heavily relied upon to define moves among tasks – in most cases to document the logic, but probably not to automate the process. This leaves the information and, in most cases the semantic networks that underlie the use of the rules. So far the CMMN team seems to be trying to ignore this. I don't think that will prove successful. I suspect that, as they evolve this notation, they will find that they want to treat most tasks as a semantic net that capture knowledge about the task (or they may keep the tasks as a nod to the procedural flow and associate a semantic net to each task.) (There is already an icon for a *CaseFileItem* – a page with the top right corner turned down – which could serve this purpose if it was developed.) Developers are going to have to specify the semantic networks anyway to formally define all the objects and attributes to be used in the knowledge rules, and I suspect in the long run it will be worthwhile including it in the notation and storing it in whatever software product is developed to support CMMN.

I don't frankly like the idea of developing CMMN as a separate notation. I created Figures 1 and 3 to highlight that the overview of the process could be developed using BPMN notation. I would rather include tasks within BPMN processes because I'd like a way to go from high-level and very abstract processes to more concrete and dynamic subprocesses. A special type of process notation to indicate that the process or activity would include rules and tasks seems straight forward to me. But I'm not on the task force and haven't been involved in their discussion, so I may be overlooking lots of things.

All this returns me to a point I made in an Advisor in October. We are beginning to create process diagrams that include both procedural and declarative elements, and no matter how we do it, modeling is going to become more complex. We are going to need tools that let us keep track of high level flows, of rules that help us negotiate moves among hundreds or thousands of possible "tasks," of the semantic networks that describe the vocabulary used in the rules, and of the data that we use for specific case instances.

Most companies are not trying to define these very dynamic processes at the moment. Or they are doing so without a formal notation. As we try to formalize the knowledge employees apply to solve problems in an increasing complex business environment, however, we will need such a notation, and a methodology to go with it. So, I suggested earlier, it's at least time to start thinking about what it might look like.



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.

Notes

Case Management Model and Notation (CMMN). OMG Specification. FTF Beta 1. Document Number dtc/2013-01-01

To examine an expert system with thousands of rules that solved medical diagnosis problems, see: Buchanan, Bruce G. and Edward H. Shortliffe. *Rule-Based Expert Systems: The Mycin Experiments of the Stanford Heuristic Programming Project.* Addison-Wesley, 1984.