Business Process, BPMN and Business Rules

I follow lots of on-line discussion groups where people are exchanging predictable ideas about topics that I find boring and of marginal interest to someone interested in business process trends and methodologies. Recently, however, I was jolted awake by an exchange by one of the OMG standards groups.

It began when an individual wrote to ask how to use BPMN to model a problem he had encountered at a client's organization. I've modified the problem, a little, to avoid focusing on the individual or the client involved. But, in essence, the problem was:

At the tax collection agency a tax return will be held (not processed) if it claims payments that have not yet been received or claims credit elects from previous years that have not yet been posted. (There is a limit or time-out on this.)

How do we diagram the process when one instance of "tax return" is waiting for information from another instance of a "tax return"?

For any tax return process, after a credit elect is posted, there is a process that checks for the existence of a held return for the same taxpayer/s. If there is a held return, and that held return is being held for a credit elect, the current process sends a message to...

This process can get very complex, very quickly. You would need notation that business people don't normally use to capture the "circularity" and the waiting involved in this process.

The author proposed a BPMN solution that was awkward and asked if others could propose something
better. Two or three BPMN gurus jumped in and suggested alternative models that were only marginally less complex than the author's initial model. (You might pause and sketch out how you might model the problem before reading on.)

Up to this point I wasn't paying much attention—just another complex BPMN modeling problem that most business people would leave to IT to solve. Everything changed, however, when Ron Ross joined the discussion. Ron is a co-founder of Business Rules Solutions, a consultancy, and also the chair of the Business Rules Forum (the leading business rules conference in the US. See www.businessrulesforum.com). Ron is active at the OMG because the OMG has several Business Rules standards that Ron has contributed to in a significant way, but he is not normally active in the BPMN exchanges. And, of course, Ron Ross also writes a Business Rules Solutions Column that appears regularly on the BPTrends website.

When I noticed that Ron had joined the discussion, I quickly opened his email to see how he would solve the problem. I reproduce a portion of his response below:

"I can't tell you whether your pool and message solution is correct. What I can tell you is this:

Even if you can find the correct solution, the business intent of what you end up modeling will be hopelessly lost.

You have a rule problem, not a process problem, and a relatively simple one at that.

A much better solution would be to externalize the rule from the process (in a form that people can read, understand, and change), as well as from the state information ("tax return that is held")."

Here is a rule to illustrate the solution. [1]

Rule 1 (Version 1): For a tax return
submitted for a given year...

**If** the return claims a payment that has not been received, or A credit elect for a tax return from the same taxpayer for a previous year has not been posted, **Then** the tax return will be held (not processed).

Rule 1 (Version 2): To deal with a timeout:

**If** the received date of the tax return is less than 6 months in the past, and A return claims a payment that has not been received, or A credit elect for a tax return from the same taxpayer for a previous year has not been posted, **Then** the tax return for a given year must be held.

I am embarrassed that Ron's response had not been my own first response. I certainly said, "Yes! Of course!" In the BPTrends Associates methodology we put quite a bit of emphasis on the balance between process models and rules, but I had, without thinking about it, considered the problem—in the context in which it was presented—as a problem in BPMN notation.

All business process analysts who work with business managers—as opposed to those who simply model processes in order to generate software code—owe it to themselves to consider this problem and the solution very carefully.

Reframed, the BPMN model is as follows:

<table>
<thead>
<tr>
<th>Customer</th>
<th>Submit Tax Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tax Agency</td>
<td>Evaluate Tax Return</td>
</tr>
</tbody>
</table>

The solution is not to make the BPMN model more complex to handle the complexities involved in evaluating tax returns. Instead, the solution is simply to indicate that Evaluate Tax Return relies on a decision rule like the Rule 1 (Version 2) presented above.
The simple BPMN, with a page attached listing the business rule used by Evaluate Tax Return, will communicate the situation to any business team that is interested in the process.

A much more complex BPMN diagram will not only not help but it will go a long way towards convincing the business managers involved in the effort that they want nothing to do with the process analysis effort or with process modeling.

This is not, of course, an argument against complex modeling for software specifications, but it is an argument for using tools that help business managers understand their processes. To put it more strongly—the goal of BUSINESS process analysts is to help business people improve their processes. We should not use process models as an end in themselves, but as a communication tool. And clear communication is usually facilitated by simple models. In most cases a judicious combination of simple process maps and worksheets with business rules cross-referenced to the activity/ies in which they are used will provide the most useful solution.

I just hope that the next time I encounter a similar situation, I can answer, as quickly as Ron Ross did. The solution isn't a complex process diagram—it's a simple process model and an easy-to-understand business rule.

Till next time,

Paul Harmon

Note: Business Rules and Business Process people have often conceptualized their domains independently and met to talk about their concerns in isolation. Increasingly, however, BPM conferences are including good talks on the appropriate use of Business Rules and vice versa. Business process practitioners would, for example, find lots of useful talks at this year's Business Rules Forum, which is held on October 26-30 in Orlando.

[1] Ron’s original solution was expressed in RuleSpeak, an SBVR-compliant language created by his company. RuleSpeak avoids If-Then syntax for reasons not important here, but since If-then syntax is more familiar, I’ve rewritten the rules that way for this Advisor.