BPM, Anyone?

A Response to Paul Harmon's response to my response to the BPTrends Email Advisor of November 15, 2005

Keith Harrison-Broninski

First, let me say that it is a pleasure to engage in this important discussion with Paul in a public forum devoted to the business process field – it is certainly a civilized and practical way to drive out the key issues facing practitioners at this time.

For those readers who have forgotten by now what we are arguing about, Paul proposed in his email advisor of November 2005 that BPMN is a modeling notation sufficient for all business processes, to which I responded in December 2005 that BPMN is only appropriate for "mechanistic" processes – programmatic, repeatable sequences of activity in which human involvement is limited to key data entry and decision points. I believe that many core activities on which business depends (strategy, design, marketing, sales, and so on) belong, in fact, to another type of process entirely, characterized by the adaptive, innovative behavior of collaborating humans. Each instance of such a "human-driven" process evolves in practice, and, hence, operates differently from other instances. The techniques for defining and controlling such processes are known collectively as Human Interaction Management (HIM). For process modeling, HIM makes use of a simplified and enhanced version of the long-established notation known as Role Activity Diagrams (RADs) rather than "programmatic" notations such as BPMN or the UML.

In his counter-response of January 2006, Paul made essentially two points. He asserted:

1. That any notation is OK for high-level modeling where automation is not required. In his words:
   I first encountered "workflow diagrams" when I went to work for Geary Rummler in the late Sixties. We used simple workflow diagrams to help business people describe what they did. We didn't focus on processes as candidates for automation. We weren't interested in automation and neither were our clients ... I don't think the notational elements make much difference. I think the key is the rigor with which you attempt to define the processes. Business people normally seek to define processes in a more general, generic way. They don't focus on exceptions. They assume that a box that says "Complete Application" describes human activities and that the people involved will conduct whatever dialogues they need to undertake to complete the application.
2. That my RAD case study (of Business System Support in a global manufacturing company) looks very similar to a BPMN diagram rotated through 90 degrees. Again, in Paul's words:
   A case study developed by Harrison-Broninski has been posted on BPTrends this month. It contains a RAD diagram of a Business System-Support Process. To my eye, that diagram is just a complex version of a Rummler-Brache swim lane diagram, turned on end.

Let's take these in turn. Starting with the first point, Paul asserts that notation doesn't "make much difference" if you are not "interested in automation." I agree with this, at least to a point. Any old notation can, in principle, be used if you do not intend to implement computer support for your business processes – since you are unlikely ever to discover what you left out or simply got wrong. Business will proceed as normal despite a mismatch with the diagrams in your quality manual or policy book, although some readers might be wondering what the ROI then was on creating the diagrams in the first place. To remedy this, we have BPM. Any BPM practitioner must surely believe that computers can and should be used to facilitate business processes –

and in this case, you need to bring some formality to the way in which you address the problem, and particularly to the process notation you choose.

I set out in my response of December 2005 a core reason why traditional workflow notations do not match the reality of adaptive, innovative, collaborative human activity. So I will not repeat that particular argument here, and instead refer the reader back to the article concerned. I will just say that, contrary to Paul's experience, my own experience over 15 years of industrial process modeling is that using traditional workflow notation to capture human-driven processes simply provides business people with a false sense of reassurance. They come away thinking they understand something at a high level, but this "understanding" does not match the reality of the work carried out.

Further, I must disagree with Paul that business people "don't focus on exceptions" – it is generally accepted that the 20% of exceptional cases give rise to 80% of the overall costs. This business truism is often referred to as the **Pareto principle** – and mainstream process management methodologies such as Six Sigma that focus on reducing variation between instances effectively take their justification from it. Moreover, in human-driven processes *just about every case is an exception*, since most of what humans do in such processes is to use their initiative and intelligence to respond to changing situations as they arise; in other words, they redefine the process as they go along, which gradually moves even a normal case away from the pre-defined high-level process they are supposedly carrying out.

Turning to Paul's second point, he asserts that my example RAD looks very like a BPMN diagram turned on its side. Again, I agree with this, at least to a point. Yes, Roles in a RAD do look a bit like rotated swim lanes. However, the resemblance is purely superficial. The semantics underlying the notations are different in almost every respect.

One fundamental difference between BPMN and RADs is, in fact, concerned with swim lanes. In BPMN, the swim lane concept is implemented by two types of container, **Pools** and **Lanes**. A Pool may contain several Lanes. Quoting the BPMN specification (v1.0 of 3 May 2004),

> "A Pool represents a Participant in the Process. A Participant can be a specific business entity (e.g., a company) or can be a more general business role (e.g., a buyer, seller, or manufacturer). Graphically, a Pool is a container for partitioning a Process from other Pools when modeling business-to-business situations, although a Pool need not have any internal details (i.e., it can be a "black box")" (P.103).
>
> "Lanes are used to organize and categorize activities within a Pool. The meaning of the Lanes is up to the modeler. BPMN does not specify the usage of Lanes. Lanes are often used for such things as internal roles (e.g., Manager, Associate), systems (e.g., an enterprise application), an internal department (e.g., shipping, finance), etc." (P.106).

In other words, a BPMN swim lane is at most a suggestion that certain activities should be carried out by the same agent. A Role box in a RAD, by contrast, is a much more powerful concept, and carries with it various constructs of real-world importance.

In particular, a RAD Role is not just a grouping of activities, but a "private information space" – it contains data used by the activities, and not shared by other process participants (technical people might think of this as a **context** or **scope**). This data may be structured or unstructured. It may be entirely contained within the Role or a pointer (e.g., a filename or URL) to data held at a separate location. It may conform to a definition laid down at the time the Role was defined or be extended with other data items during the operation of the process. The concept is very rich, and corresponds directly to how humans work – we all need to maintain our own data stores in order to accomplish anything useful – and may be leveraged by the process modeler to make diagrams that reflect the reality of human behavior.

In the example case study referred to above, the Interactions between Roles pass data items from one Role to another – not only to request a response, but sometimes without even requiring any action to be taken, as when the Production release notice is sent out. The need for all process participants' private data stores to be updated, as and when necessary, is an aspect of human behavior to which other notations do not naturally cater, or even suggest the need.

Private data within a Role is also used in RADs when defining *enabling* and *validating* conditions on activities, as well as when creating a *terminating* condition for the Role itself. A further common use of such data is to represent the information required in order to start and stop Role instances as necessary. All these concepts are missing from BPMN, yet represent normal and vital parts of human collaborative activity in which flexible but powerful control over activities is required, and where process participants typically come and go during the life of the process. A human process participant's private data is not ancillary to how they do their work, but the foundation of it so a suitable notation must deal with this explicitly.

Another weakness of BPMN semantics compared to those of RADs for the description of human collaboration is the representation of communication. Quoting the BPMN specification again,

> "A Message Flow is used to show the flow of messages between two entities that are prepared to send and receive them" (P.116).

Only *two* entities? In typical human interactions there may be any number of people involved – think of a conference call, team meeting, document distribution, group notification, or any other everyday business interaction. RADs permit this to be modeled via a simple horizontal line joining any number of Roles; for example, many Interactions in the example case study referred to above are multi-party. To depict this typical communication pattern with BPMN you have to fabricate complex webs of interaction, generally using artificial intermediaries to collate and deliver messages, which runs the risk not only of confusing the user but also of introducing sequence errors into the diagram – all because the BPMN notation does not match the reality of human behavior.

What BPMN *does* match is the reality of fully automated computer behavior, which is its origin and natural purpose. Unlike humans, program language objects typically talk to only one other object at a time, which is why BPMN follows their example. BPMN is a powerful notation for representing programmatic sequences of activity, with occasional human involvement for data entry and decision-making. It was not intended for, and is not suited for, anything else.

There are various other important differences between BPMN and RADs that make RADs a more viable notation for depicting human-driven processes – the ability of RADs to model provisional decision-making; a transaction concept in RADs that closely reflects human behavior; and so on. However, in this document I have tried to provide just an overview of why BPMN is unsuited to modeling human collaborative activity, and how RADs remedy its deficiencies in this respect. I have kept things as simple as possible for readability, focusing on a few core notational issues rather than providing an exhaustive comparison.

If this public debate continues, I look forward to discussing and exploring other important issues at a later date. For now, though, you can find more information about RADs at www.human-interaction-management.info and in my recent book, *Human Interactions: The Heart and Soul of Business Process Management* (Meghan-Kiffer Press, 2005). The theory of **Human Interaction Management** starts from a description of core process modeling techniques based on RADs, and goes on to provide a powerful methodology not only for developing but also for *managing* human-driven processes and supporting these management techniques with software.

My position with regard to BPMN can be summarized as follows: If you want to apply BPM to human-driven processes, you cannot get away with a notation that only vaguely captures the process. Yes, you need a notation that is clear enough to suit business people as well as technical people, but the notation must also be strong enough to permit facilitation of human-driven processes via software. After all, take computerized process support away from BPM, and what have you got left?

Author Bio

Keith Harrison-Broninski (keith.harrison-broninski.info) is a consultant, writer, researcher, and software developer working at the forefront of the IT and business worlds. He is

- Author of the landmark book *Human Interactions: The Heart And Soul Of Business Process Management*, described by reviewers as "*[a key component of] the overarching framework for 21st century business technology – a must read for Process Professionals and Systems Analysts alike,'* www.human-interaction-management.info

- Contributing "thought leader" to the BPMG book, *In Search Of BPM Excellence* (BPMG, 2005, www.bpmg.org)

- Author of numerous articles on the topic of Human Interaction Management (HIM) and the Human Interaction Management System (HIMS) "*There are a number of collaborative activities that go beyond workflow and knowledge management, which I call human interaction management, and that is going to be the next envelope pushed in the whole BPM space."* –Peter Fingar, interview for CIO Magazine Australia, September 2005

- A founder member of the Process Modelling Group (www.process-modelling-group.org).

Along with his research and consulting work, Keith is now the CTO of Role Modellers Ltd (www.rolemodellers.com). The company mission is to develop understanding and support of human-driven processes across industry, a field that Keith has pioneered. Their new product, humanedj (www.humanedj.com), will be released in 2006.

In the November 15, 2005 BPTrends email Advisor, Paul Harmon writes:

> *BPMN is currently the best available notation for business managers and we ought to rally round it and put aside our favored idiosyncratic notations, for the common good. If our goal is to get business people to focus on processes and to learn to express ideas about processes in graphic form, we need a common process language. It will make everything easier. I suggest we all use BPMN.*

Paul describes the enormous amount of work that has gone into achieving consensus on BPMN, and questions why any other notation should be necessary. As he rightly says, standardization is for the common good.

But standardization on what, in this case? BPMN is certainly the best available notation *for the type of business processes for which it was designed* – the processes that act like what computer scientists call "automata". But there are other kinds of business process. Bear with me here, reader, since we need to understand what an automaton is before going further.

In plain language, an **automaton** is a machine that has different "states" at different times – a state can be thought of as a set of values for the machine's properties. Further, an automaton is limited in its behavior, in that it cannot freely move from any state to any other state – from each particular state, there are only certain other states that it can go to next.

This notion of automaton is the basis of much of computer science. In particular, it underpins both the "process formalisms" known as Petri nets and pi-calculus – and these are in turn the basis of BPMN. So BPMN has (as Paul says) good claim to be the best graphical tool available for depicting business processes that act like automata.

But do all business processes act like automata?

I have been arguing for some time now, both online and in print, that *human collaborative activity* is not like this at all. There really are two kinds of business process: "mechanistic" and "human-driven".

Mechanistic business processes are largely implemented by machines, with human involvement limited to key decision and data entry points – they are not unlike computer programs in their essential nature. Most processes involved in order-to-cash and supply chain are like this, and even some service processes (such as insurance claim handling) can be treated in this way.

Human-driven processes, on the other hand, are quite different. They are fundamentally collaborative, dynamic and innovative. Consider such everyday business activities as product design, IT outsourcing, complex sales, marketing, company growth/merger/divestment, health care, human resources and so on – such processes do not act like automata, since they are full of sudden jumps from one place to another, for two reasons:

1. People take action based on a variety of uniquely human responses to situations – if you do not recognize this you are burying your head in the sand. People change their mind, make provisional decisions, are affected by impulse and emotion, and allow one thing they are doing to affect other things they are doing. It's just the way we are, all of us.
2. Human-driven processes are subject to continual change. This change is not a superficial aspect of such a process, but embedded right into it. Much of the work humans do is "deciding what to do next" – we change our processes as we go along, in other words. The ability to do this well is often what makes a person successful. It may represent a large part of what they are employed to do.

Hence, trying to model human-driven processes as automata is like going to sea in a sieve. You spend your whole time baling out, since the equipment you are using is profoundly unsuited to the task at hand.

Actually, to deal properly with human-driven processes you need a lot more than an appropriate modeling notation – you need patterns for using that notation, and techniques for process *management* that support the continual process change endemic to human collaborative activity. For more on this, see www.human-interaction-management.info. However, let's stay with notation for now.

The simplest way I have found to represent collaborative human activity is a version of the notation known as Role Activity Diagrams (RADs) - stripped down to remove unnecessary complexity, and reinterpreted to match how people really behave. So I use such RADs to depict human-driven business processes, build software based on these diagrams, and have always found them to have the following advantages:

- Business people can understand the notation immediately without training
- You can generally reduce a complex set of documents to a single 1-page RAD.

References to articles and books about this use of RADs can be found at the link above. For the purpose of this article, the point is that RADs, reinvented in this way, provide a means of showing human activities that allows the people who will carry them out to understand what is going on, and that is amenable in principle to software support. To achieve this, the notation must *correspond to the processes in question*. And since humans are not automata, BPMN doesn't fit this criterion.

So what about standardization? The issue is easier to resolve than it may seem. What we need is not a single notation for both types of process – it is the ability to use the same software tools to design (and possibly support) both types of process. We must standardize on *process exchange* - and fortunately, we already have the ability to do this. For some years now the OMG has been working on a common means of representing any objects, both types and instances, whatever their provenance (known as **MOF**). There is even an XML dialect available for importing and exporting such objects between software tools (known as **XMI**).

So any process that can be saved in XMI format can be opened in any MOF-compliant tool – of which there are many, Rational Rose being only the most well-known. Further, since such tools typically integrate *process design* with lower-level *program design*, this is a very productive and practical route for process support via enterprise IT. The fact that there exist 2 diagrammatic representations of the process objects, according to whether they belong to a mechanistic or a human-driven process, then doesn't matter. What matters is the process content.

To finish, it is worth noting that in principle, BPMN could be bent and twisted to represent human-driven processes as well as mechanistic ones. Just as any "Turing-complete" programming language can in principle express any conceivable program, any process modeling notation containing a minimum set of constructs can express any possible process. However, this is not to say that one programming language - and one process modeling notation - is all that we need.

With respect to programming, it is taken as read by all system developers that different languages are appropriate for different circumstances – some are higher level but less efficient, some are customized with special constructs for mathematical calculation, some offer the means to build dynamic Web sites with minimum effort, and so on. IT is an engineering discipline, in which the professional chooses their tools to match the needs of the situation, rather than applying the one-size-fits-all approach of a hobbyist.

Similarly, with process modeling, BPMN may well be a sensible choice for any process that genuinely deals with automata: i.e., computing systems with human involvement at key points only. However, applying BPMN to collaborative human activities is uncomfortable at best. Humans are not automata, as shown above, so modeling their collaborative activities in BPMN is tortuous and long-winded for any real-world process, especially given the continual change that is inevitable once the work starts.

It is easy to overlook the fact that simple "textbook" process examples do not in any way illustrate the true nature of human collaborative activity. As discussed in Appendix B of the Process Modeling Group Workshop Proceedings from June 2005 (http://tinyurl.com/84bsj), a description of a human-driven process may appear to be realistic while actually being far too simplistic. Once you start including the true complexity of the activities that are involved in even quite everyday processes (the example in the document is a Travel Agency), the modeling effort can quickly spiral out of control unless you adopt an appropriate approach.

In real-world practice, as opposed to textbook examples, modeling human-driven processes with BPMN is so cumbersome as to be effectively unusable. One team that attempted to model a Systems Engineering process in such a way persisted for 6 months then simply gave up in frustration. A person responsible for re-engineering IT support in a global manufacturing company spent several months drawing up process diagrams which no-one used, or could even understand, before turning to RADs and finding that everything they had written so far, and more, could be expressed in a single 1-page diagram – further, a diagram that elucidated the processes concerned so clearly that all involved were able to agree on a re-engineered version within 2 meetings. The diagram in question is available online as a Case Study at http://www.human-interaction-management.info.

In fact, using an inappropriate notation for human-driven processes is more than difficult – it is dangerous, since such an approach if implemented in process support software can only lead to systems that are unnecessarily restrictive. The people charged with doing the work will then be forced either to bypass the system or to do their work badly. Through no fault of their own, people's work will be compromised by the method used to describe it – as happens all the time with existing workflow systems, when it comes to processes requiring genuine innovation and collaboration.

I listed above some examples of the many fundamental business processes that are human-driven rather than mechanistic. And there exist many more such processes that are not specifically commercial but nevertheless lie at the very root of our society: political/social negotiation, natural disaster prevention/management, crime solving, epidemic control, government policy implementation, running an election campaign, military action, and so on.

We got to grips with the management of mechanistic business processes in the 20th century – and now it's the turn of human-driven processes. Perhaps if we were to deal with such processes more efficiently with the aid of IT, the world would be a better place to live in. We can make a start by settling on the right tools for the job.

Keith Harrison-Broninski (keith.harrison-broninski.info) is a consultant, writer, researcher and software developer working at the forefront of the IT and business worlds. He is the author of *Human Interactions: The Heart And Soul Of Business Process Management*, (Meghan-Kiffer Press, 2005), a founding member of the Process Modelling Group (www.process-modelling-group.org) and CTO of Role Modellers Ltd. (www.rolemodellers.com).