Getting your head round mozzarella

MARTYN A OULD
VENICE CONSULTING LTD
The Old School, Hinton Charterhouse,
Bath BA2 7TJ, UK

phone: +44 (0)1225 723 822
fax: +44 (0)870 056 7443
e-mail: mao@veniceconsulting.co.uk

In a series of short articles, Martyn Ould explores three central features of real-world business processes – collaboration, concurrency, and mobility – and how we can get a handle on them. In this first article, we'll look at the first feature: businesses as complex collaborative processes, rather than simple workflows.

Process modelling seems stuck in the Dark Ages. It's as if in the last three decades we had never moved beyond using assembly language on our computers. The languages we use to talk about processes are poor and mean. In particular, we cannot talk about the things that really matter in the real world. Conventional process modelling discussion has got stuck somewhere around 'What shape should a box for an activity be?' when the real question is 'What *concepts* should we base our notations on in order to capture the collaborative and highly concurrent nature of real-world processes?'

When we watch an organisation at work, we see a mass of collaborative and concurrent activity, a flux of dynamic activity, and a flux of micro- and macro-responsibilities. We see processes starting up, evolving, developing, convoluting, dying. Within processes we see responsibilities being created and carried out. Where in our languages are these things talked about? How do we describe that dynamism? Our business process management systems are there to support that dynamism, so there is a real imperative for a process modelling language and method that help us truly understand collaboration and concurrency in the real-world.

In a series of three articles we shall explore some of the characteristics of the real-world that I believe must find expression in our process models and hence in our BPMSs. In the first, we shall examine businesses as collaborative processes. In the second, we shall examine businesses as active flows of concurrent responsibilities. In the final article, we shall look at the way processes evolve dynamically: their *mobility*.

Let's listen in on a conversation next to the water-cooler.

Tutor: Have you ever drawn a model of a process?

Pupil: Yes, I've drawn flowcharts showing workflows.

Tutor: And do your workflow diagrams show who does what?

Pupil: Yes, if you add swimlanes and show the work moving from one person to another.

Tutor: What do these diagrams mean by a swim-lane? And what do they mean by a line that crosses from one swim-lane to another?

Pupil: I hadn't thought about it much. I guess a swim-lane represents a job title or perhaps a department. And a line represents where the activity moves from, say, one department to another. Sometimes we add the flow of stuff – data, materials, whatever.

Tutor: That's a very impoverished view of the world. It suggests that these swimlanes are little islands and stuff moves from island to island. There must be boats travelling from one to another carrying batons that say 'your go now' and perhaps some stuff to go with the baton. Are organisations really just islands with batons and stuff moving between them? What are we doing here, now, for instance?

Pupil: Well, we're discussing the nature of processes.

Tutor: Right, we're having an interaction. We're doing something together. And in this case there's no stuff 'changing hands'. In a business process we collaborate – act together, inter-act – in a whole variety of ways:

> We discuss something.
>
> We negotiate.
>
> I contract with you to do something.
>
> I pass you some information.
>
> I delegate a task to you.
>
> I ask you for something.
>
> I give you authority to do something.
>
> We agree on an action.
>
> We jointly approve something.
>
> You report your status to me.
>
> I oversee something you are doing.
>
> You pass me the results of your work.
>
> I instruct you.
>
> You and I work on something together.
>
> I wait for you to do something.
>
> I chase the progress of your work.

We aren't islands with boat traffic between. I'd hope that our process modelling approaches recognise that very important fact: when we interact, when we collaborate, we do things together – we don't simply send the boat over with a baton and some stuff on board. Quite the opposite: in business terms, interactions aren't just about the locus of activity moving from one role to another: real, business-oriented,

value-adding things can happen in an interaction. Interactions are just as vital to the process as the actions that individual roles carry out on their own islands. Ineffective interactions can be as damaging to a process as ineffective actions. Slow interactions can affect cycle times as much as slow actions. So let's forget swimlanes as islands and think about rich interactions.

Let's go back to the question of what the swimlanes actually were.

Pupil: Well, as I said, they're ... job titles, say.

Tutor: I'm bound to ask why you're starting with job titles. Let's stand back. When a process runs, *responsibilities* come and go. When a new customer order arrives, it creates a responsibility to deal with it. (In the jargon, the responsibility is *instantiated*, an *instance* is created) This in turn generates (instantiates) further responsibilities that will contribute. The process is all about the coordination of all these contributions, all of these micro-responsibilities.

Pupil: I guess I can understand that: when we define someone's job or the job of a department, we effectively bundle up whole classes of responsibilities and say 'These are the ones you look after ... in this job.'

Tutor: Right. And when the process runs, new instances of these responsibilities are created, carried out by the nominated post or department, and disappear. The posts and departments are pretty much static things, but the responsibilities are dynamic: they come and go.

I use the term 'role' when talking about those bundles. If we take an abstract view of the world, a role is simply a single responsibility which can be created (instantiated) dynamically. If we take a concrete view of the world, a role might be a post on the organogram to which we have allocated a bundle of responsibilities.

Pupil: You're drawing a distinction here: our design of the process determines what responsibilities are created when, and how they interact; and our design of the organisation determines which roles are tasked with which classes of responsibilities.

Tutor: Right. So, if we were to design a process in terms of pure responsibilities it would probably be quite simple. But then – for a variety of reasons, some valid and some not so – we design our organisation and carve up that neat process and hand pieces/responsibilities out to the posts and groups we have invented or which we find around us. Have you ever shared a pizza? All those strands of mozzarella between you and your friends – that's the result. And it's the same with processes. When we allocate responsibilities to different posts and groups we have to create interactions between them to make sure all those responsibilities remain coordinated.

For me, a line between two roles is an *interaction*. That's an *interaction*. It's not that the locus of activity moves from one role to another. It's that the two roles do something together: agree, decide, discuss, exchange, inform, request, report. These are more than data flows. The roles must interact in order to coordinate their responsibilities. Once their interaction is over they can continue carrying out

> their separate roles until the next interaction. This isn't data flow. This isn't a flowchart. A line from one swimlane to another is a poor thing, inadequate for capturing the interaction and coordination of responsibilities that actually takes place. Moreover, a swimlane can be all too static a thing – responsibilities are dynamically created.
>
> In a sentence, getting our heads round the mozzarella (and not the other way round) means getting to grips with the responsibilities, their grouping into roles, and the interactions that coordinate their activity. With these concepts, we shall have a deep understanding of our process, rather than the skin-deep view we've become accustomed to – and all too easily satisfied with.

In this article we have looked at *collaboration* and how to think about it. In the next we shall look at *concurrency* and how we should get our heads around the mass of parallel activity that is going on in the building at any one moment.

This article first appeared at **www.BPtrends.com** in September 2004.