
Getting your head round spaghetti

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

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



In a series of 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 second article, we examine the many levels on which we can observe the concurrency of activity in the organisation.

Once again we're eavesdropping next to the water-cooler.

Tutor: The last time we stood here we talked about how *collaboration* is central a process. I didn't point out that we begged a question: how do we know that this bundle of organisational activity is a 'process' and not just some random chunk?

Pupil: I can't see why this is a problem. Surely we just need a chunking of some sort.

Tutor: If you and I walk into the same organisation's building together and, separately, go off and draw up a list of the essential processes that the organisation operates and their dynamic relationships – what I would call a *process architecture* for the organisation – what sense would it make if we came up with a different picture?

Pupil: But even if we drew the same picture, things will change tomorrow.

Tutor: I disagree. If the organisation decides to change how it's structured, why shouldn't the process architecture remain the same? Why should there be different essential processes if it's still in the same business?

Similarly, if the organisation decides to change its culture, why would the process architecture change? Why should there be different essential processes if it's still in the same business?

Clearly, how those processes are carried out will change if the chunking into responsibilities changes, or if the culture changes and the styles of interaction change with it, but the chunking will remain constant.

Pupil: I have this strange feeling that you're going to suggest that there is a process architecture for an organisation that is a sort of ... invariant for it. Things are the way they are, the organisation has those processes, simply because it's in that particular business. Am I right?

Tutor: You're getting to know me. Yes, if we could characterise the business of the organisation in a way that is independent of *design decisions* such as the organogram and the culture and the technology it has deployed, then the 'chunking' of organisational activity into processes that we arrive at will be a firm rock on which to build all our other analyses of the organisation's business, for building BPMS-based systems for instance. Put simply, for an organisation in a given business there is a given process architecture. I'll go further and say that we can decide what that architecture is very quickly. We need to cover that some other time, but for now let me just say that in the architecture we'll find two sorts of process: *case processes* and *case management processes*.

A case process is one that deals with one something – so we might have a process called 'Handle a customer order', or 'Handle a clinical trial'. As you can imagine, at any moment there will be many instances of each case process running, one for each customer order or clinical trial.

A case management process is one that deals with the flow of cases: so we would have processes called 'Manage the flow of customer orders', and 'Manage the flow of clinical trials'. As you might expect, given that it is all about scheduling, resourcing, prioritisation and so on, there is only one instance of each case management process at any time, overseeing the changing set of instances of the case process.

Pupil: OK, so when we walk into the building we know that there is a particular set of case processes and case management processes, and that we will have potentially many instances of the former and one instance each of the latter.

Tutor: Right, and all of these processes are interacting. Case process instances obviously interact with their case management process instance – they report on progress, for instance. But process instances are interacting all over the place, and indeed causing new instances to be created. When I run a drug development programme (a case process) I need to get going a number of concurrent clinical trials (another case process), and once those clinical trials are operating I shall need to interact with them – to guide them, get their results and so on. Our process architecture will give us the principal interactions that are there because of the business we're in. But more interactions will arise when we design the individual processes – remember the mozzarella.

Pupil: Let's see if I have this right. At the topmost level, what we actually observe in the building is a flux of interacting process instances, to use your jargon. And the process architecture would tell us what processes there are to be instantiated, when they are, and how instances subsequently interact?

Tutor: Exactly. Now let's open a box on the process architecture – let's look inside a single process, one that could perhaps have multiple instances at any moment. What do we find?

- Pupil: If I remember right from our last chat standing here, we find a network of activity involving what you referred to as roles ... which might be single responsibilities or might be bundles of responsibilities in the form of posts or departments?
- Tutor: Right. And we said that these roles interact – collaborate.
- Pupil: Something tells me you're going to use the 'instantiation' word again.
- Tutor: You're getting the hang of this. Of course I am. Responsibilities come and go: a customer places an order and a new responsibility is created to deal with it; you make a claim for expenses and a new responsibility is created to check and approve it, a responsibility that I as your line manager might be allocated. So these responsibilities come and go – yes, they're instantiated. To generalise: such abstract roles are instantiated during the lifetime of a process.
- Pupil: So we're not talking swim-lanes here?
- Tutor: Absolutely not. The real world is dynamic. Swimlanes are static. When a new instance of a case process is created to ... let's say ... carry out a clinical trial, a whole set of new responsibilities are generated related to that process instance: there's the responsibility for managing the trial, for developing the protocol for it, for recruiting patients for it, and so on. These responsibilities are created dynamically. Sure, they're handed out to people in specific posts or with certain job titles, but those are static things that could change tomorrow. Summarise for me.
- Pupil: Well, our process architecture ... our invariant process architecture for the organisation gives us a set of processes that can be instantiated, and as a result we see a constantly changing network of interacting process instances. And then within a single process instance we have a set of roles that can be instantiated and hence a constantly changing network of interacting roles.
- Tutor: Yes, it's what we might call a flux of concurrency. But I want to add one more level: concurrency within a role instance.
- Pupil: I almost suggested that. When I'm playing a part in a process I could have several things I could be getting on with: one bit of my brain might be dealing with getting the goods from the supplier and another bit could be sorting out the paperwork ... and I suspect there are more instances about to pop up?
- Tutor: Right: you might be a programme manager (a concrete role – a bundle of responsibilities) and you might start collecting status data about all the separate projects in your programme. You email all your project managers at the same time, and then – in parallel – collect data from them, question them, and so on.
- Pupil: So a role instance is a network of possibly interacting instances of threads of activity. I seem to have spaghetti in spaghetti in spaghetti.
- Tutor: Spot on. Concurrency is as rich a thing as collaboration. To really get our head round it at all its levels we have to capture the instantiation

of processes, the instantiation of roles within process instances, and the instantiation of threads within role instances. Miss any one of these and we miss a chunk of the real world.

This ... terrifying ... flux of instances - of processes, of roles and of threads in roles - is what concurrency in the organisation is all about. And the networks of role instances and their interactions - within and across processes - is what collaboration is all about.

Collaboration and concurrency - get your head round them!

This article first appeared at www.BPtrends.com in October 2004.

Martyn Ould's new book *Business Process Management – A Rigorous Approach* describes a business-oriented method for describing, analysing, and designing business processes for BPMSs and for traditional information systems.

The book is endorsed by BPMI.org.

For more details visit www.bcs.org/books/bpm.

Copyright © Martyn A Ould 2004