All the world's a stage

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 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 third article, we use a theatrical analogy to summarise the notions of collaboration and concurrency from the previous two articles and introduce the idea of mobility.

> 'All the world's a stage – let's play a part.'
>
> Shakespeare and Elvis Presley

In my first article – *Getting your head round mozzarella* – our Tutor and Pupil explored a central feature of real processes in real organisations: the notion of *collaboration* in a process, the fact that the roles that play a part in a process not only act, but inter-act – they collaborate – and that the way we hand out dynamically created responsibilities (abstract roles) to static things like posts in the organisation (concrete roles) creates the mozzarella of collaboration.

In the second article – *Getting your head round spaghetti* – the Tutor and Pupil discussed another central feature, the massive 'flux of concurrency': the constantly changing flow of instances of processes, roles within process instances, and instances of threads within role instances.

In this third article these concepts come together in another conversation next to the water-cooler, adding the notion of process 'mobility' which is behind the idea that processes themselves are dynamic objects.

Tutor: We've been talking about activity in our organisation. I'd like us to think of it for a moment as a theatre – a rather twenty-first-century one with some unusual characteristics. Let's start with processes.

Pupil: Well, we've got processes – plays, I guess. Plays are written down in scripts ... process models, and ...

Tutor: Stop there for a moment. Think about instances ...

Pupil: Plays are performed ... they have performances ... process instances! So when we walk into the theatre we'll find performances of plays going on. Ah – the theatre seems to be a multiplex, because I have lots of

plays being performed at the same time! Some plays are being performed many times simultaneously. Hmmm, worse still, some performances start up new performances! And presumably they have to find a stage to operate on.

Tutor: Concentrate on one performance for a moment.

Pupil: Well, a play consists of a number of roles. Each role in a performance … ah, each role instance in a performance … is acted by an actor – who was cast in that part somewhere along the line. Things start getting a bit weird around here, I suspect, because during the performance some actors might be acting several roles each, rushing around the stage from one to another. Worse, I suppose they might be acting roles in more than one performance, so they'll have to run from stage to stage, changing costume as they go. In the worst case they'll be acting several roles in several performances.

Tutor: When we're talking about an individual performance, remember we need to talk about role instances.

Pupil: Yes, that's a bit weird too: how many Hamlets can you have in one performance of *Hamlet*? Only one, I guess; but we certainly see processes in which some roles are instantiated many times – mercifully Shakespeare was content with a single instance of the Prince of Denmark on stage at any one moment. But it appears that in some plays new role instances are created while the play is in progress – as new responsibilities arise. And an actor has to be found to play the role instance. In extreme cases, the actors might be writing bits of play, inventing new roles, instantiating them, and then casting actors as they go along. This is truly contemporary!

Tutor: Absolutely. You mention casting …

Pupil: Yeeees. Presumably the allocation of actors to role instances is just more process … so casting happens on stage, and probably during the performance as the roles get instantiated – as the responsibilities get created!

Tutor: Right. Any thoughts about props?

Pupil: Well … the props are the resources an actor needs to play a role instance. It might be a book or a newspaper in a real play, and an information system or a software application in a real business process. Their costume sounds a bit like the mind-set they need for the role!

Tutor: So we have a number of stages, each with a performance of a play going on. Performances are starting up and stopping all over the place. On each stage, role instances are being played by actors, who are possibly rushing from play to play and from role instance to role instance, putting costumes on and taking costumes off, and picking up and putting down props as appropriate. So far so good. What about the actions that the role instances carry out and the interactions between them?

Pupil: I'm not sure I want to watch one of these plays. Sometimes a role instance is doing an action: so the actor is giving a soliloquy; some-

times it's interacting with another role instance – having a conversation – or even with several others at the same time. There may be several soliloquies and several conversations all going on at the same time in the one performance . And occasionally proceedings will get held up for want of an actor because they're acting another role instance on another stage. It's madness.

Tutor: Hold that thought. You've only dealt with role interactions in a single performance, but we know that process instances interact. And in the theatre?

Pupil: Oh dear. Some plays are connected. A performance of one play has to interact with a performance of another. That means that role instances in the two performances have to interact – there must be communications systems between the stages – telephones or email or something. Or perhaps the actors rush to and from each other's stage, or perhaps they meet in the corridor? My head is starting to spin. At least the plays are fixed. That must be a constant in this madness!

Tutor: I'm afraid not. It's about to get much worse. The scripts of plays can be rewritten. We might have liked *Hamlet* the way Shakespeare wrote it, but we might also consider changing it for modern times – strengthening the substance abuse angle at the end for instance.

Pupil: Eeeek. So who is it that is changing the script?

Tutor: That depends. In some situations the original author might come along and produce a new version, perhaps rewriting the final act to give it a different outcome, or refining some of the smaller parts for better characterisation, or removing unnecessary material. When a new performance of the play is about to start, the actors can use the new version.

Pupil: That's understandable in the real world: we have all sorts of reasons for wanting to change the way we do things – our processes. But presumably the whole business of changing a script happens outside the theatre?

Tutor: NO! The scripts are in the theatre – that's the only place you can change them. Not only are they in the theatre, they are part of the subject matter of the theatre: in other words, you can get hold of them. Now, you can only change a script by using the *process* for changing scripts. 'Handle a script' – let us say – would be just another process, indeed a case process, one with as many instances as there are processes/plays.

The whole point of this theatre is that it is where *everything* happens – there is no 'outside' – and one of the things that can happen there is that you work with your processes. Putting it another way: the theatre supports you *in* your processes, by managing all the collaboration and concurrency; but it also supports you *with* your processes, by giving you all the means you need to write new ones or change existing ones.

Pupil: I can accept that a script can get changed and new performances use the new script. But presumably any performance in progress is unaffected? Please?

Tutor: Why so? Why should a performance not switch to the new version as soon as it's available? Why should it be forced to carry on with the old one? In some situations sticking with an old script might make sense – for consistency reasons perhaps – but in principle we don't need to make performances stick with old scripts.

Pupil: I'm struggling … I have a picture of a 'master' script which could be changed. Any performance in progress might continue with the old version, or switch to the new version.

Tutor: Who said anything about 'masters'? Why shouldn't a performance use its own variation of a script?

Pupil: … because … they … OK, why not? So, they might start with the 'standard' script for Hamlet but decide to change it in some way for just this performance?

Tutor: Of course! They're doing a lunch-time slot, they're ten minutes in and realise that people don't have time for a full *Hamlet,* so they quickly do a rewrite and present a reduced version. There is nothing fixed or sacred about a process. Fitness for purpose, matching the process to the customer – those things are sacred.

Pupil: Now I'm getting concerned about the sort of chaos that will ensue if we let everyone tinker with processes as they please.

Tutor: Then don't let everyone tinker with processes as they please: you must script the 'Handle a script' process to control what can and cannot happen to a script. Everything is in the theatre, including control over use of the theatre.

Pupil: My head is hurting.

Tutor: OK. Let's take a breath. Why don't you think through the implications of the existence of the 'Handle a script' process.

Pupil: Well, I guess that for each script there is an instance of 'Handle a script' running. Which means there is a stage where that performance is going on, and that performance has the actual … the paper script on it. Presumably if someone wants to use the script they get it from that process, from that stage. They can take it away in some form and use it for a new performance on its own stage. So far so good?

Tutor: Yes. But I want to put what you said in a different way: *scripts can be handed around.*

Pupil: Oh dear. One performance can give a script to another performance?

Tutor: Of course. Processes are truly mobile. When an interaction occurs between two process instances, a process might be involved. In traditional computer systems, data was passed around, or messages were passed between objects. The object-oriented paradigm is only halfway to full process thinking: the unit of currency ought to be the process.

Pupil: One of the things that has really come home to me is that processes aren't – except in a few cases – simple static workflows. They evolve and blossom and die as the organisation 'runs'. And moreover they can be evolved and changed and developed as objects in their own right. I can see that instantiation is the key to this.

Tutor: Yes, and a good example is email. If you observe an email conversation – a process instance – you see it develop and spread and die back and stop. It can do this – in a way that a workflow cannot – because new participants can be brought in and introduced to each other, making new interactions possible: the process evolves.

Pupil: I'm beginning to feel that a workflow diagram with swimlanes is only going to allow me to describe the simplest of processes.

Tutor: You're right. We need appropriate concepts, we need a language built on those concepts, and then we need a method using that language, that together give us a handle on collaboration, concurrency, and mobility. That way we'll get a valid and stable understanding of our business, before we set off trying to move our processes onto a BPMS.

This article first appeared at www.BPtrends.com in November 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