9 Discovering & defining processes

INTRODUCTION

In the simplest of situations, we want no more than to find out what is going on in the organisation. We're in the world of *discovery*: there's a process out there, but no-one can quite say what actually happens. People are doing things and perhaps the right results are being obtained. But our individual responsibilities – what we each have to do – are passed on by word of mouth, by rote, or by tradition. The fact that results are produced at all is perhaps a matter of luck rather than judgement.

Once we have been through the process of discovery, we shall have a process model that says 'This is how in practice we handle customer complaints.' We're now in a position to give each individual – perhaps for the first time – a helicopter view of how things work and how their contribution fits with that of everyone else – or doesn't: 'This is how your work contributes to customer complaint handling.' At last we're able to share that understanding across a group of people: 'So, this is how together we handle customer complaints round here.' This will be the first step towards change and small-scale improvement: 'If you could do this, my life would be made much easier; in return I can …'

As well as discovery, we might want to write up our process as a work instruction: 'This is how we will all handle customer complaints.' This sort of process *definition* can serve several purposes:

☞ to ensure we continue working together in a way that we all share and understand;

☞ to ensure that new people joining the organisation can get into the swing of things quickly, by following the process that has been laid down;

☞ to ensure that we deliver the quality our customers are looking for in our product or service by working in a way that guarantees – or at least increases our chances of delivering – that quality;

☞ to satisfy a regulatory body that governs how we go about our work, by carrying out our processes in the same way, day in and day out, so that we achieve the necessary quality.

In this chapter, we shall look at how best to use the concepts we have developed so far to discover a process that is out there somewhere, and then go on to examine the different ways in which we can present a process definition for the different purposes we might have. But first we need to touch on a subject that will crop up a lot in modelling work: the *boundary* of a model.

The boundary of a process model

When we draw a process model, no matter how much detail we put in it, or how 'big' it is, we are always drawing a boundary somewhere, a boundary that says 'For this model, I'm not interested in what happens beyond here.' We are making a modelling decision.

The black-box action as a boundary

The first and obvious boundary in a RAD is the black-box action. By representing an action as a black box we are treating it as an atom and saying that we are not concerned here with the detail of what goes on in that action. The black box is a sort of lower boundary to the detail about activity that we want to capture. We might wish at some point to examine what happens 'inside' the box – we dealt with this in Chapter 4.

Here are some examples of the sorts of things we might say when describing a single action. Note how they might describe a small, short-lived activity or something very large and very long-lasting.

> Prepare production line for new model.
>
> Complete hand-over form signalling acceptance of new production line.
>
> Agree pricing with dealerships.
>
> Issue Press Release.

The interaction as a boundary

Similarly, an interaction is an atomic thing: we have no detail about what goes on when the interaction takes place. It just happens at the appointed states in the participating roles and that's an end to it as far as the model is concerned. Another lower boundary of detail. Again, we have chosen not to look more closely in this RAD.

As with an action, the 'size' and duration of an interaction can vary enormously:

> Supplier demonstrates equipment to Potential Customers.
>
> Supplier and Customer negotiate price.
>
> Supplier and Customer sign agreement.
>
> Supplier maintains Customer's equipment for ten years.

The trigger as a boundary

Any trigger is clearly a boundary too. Suppose we have the event *Decision made to carry out an audit* as the trigger of a thread in a role. This says that the thread starts when this event occurs, and that – on this model – we are not interested in where or why or how it occurs, only that it occurs.

Pre-existing role instances as a boundary

We can consider pre-existing role instances as a form of boundary to our model. If we draw the role *Task Force* and mark it as having a pre-existing instance, we are saying 'However it came about, there is a single instance of the role *Task Force* when this process starts.' An instance might be there because it's a fixed post in an organisation, such as *CEO*, or because

some other role in some other process – we care not what on this RAD – has created the instance in order, say, to activate this process.

Boundary definition by omission

Finally, we implicitly define the boundary of a model simply by omitting things. The most common example of this is where we only draw those parts of a real-world role that are relevant to the process we are looking at. Figure 9-1 shows how we might wish to indicate that, for certain items of expenditure, the Finance Director needs to be told, and to that end there is an interaction with the *Finance Director* role. But we have chosen *not* to show just what the Finance Director does with the information – this is outside our interest for this RAD. Clearly, the Finance Director does more in life than is suggested by this RAD, but everything besides accepting notification of capital expenditure is outside the boundary chosen for this model.

Figure 9-1 – Omission as boundary
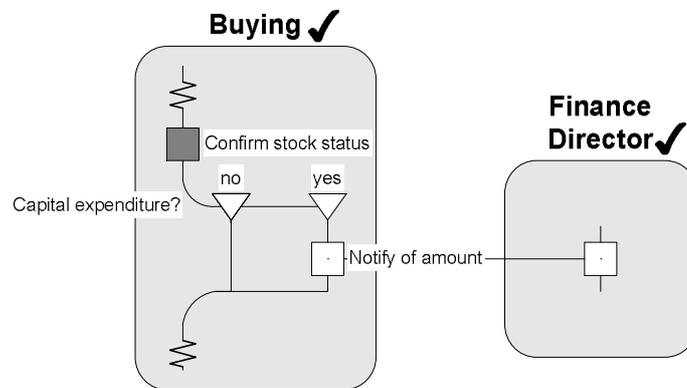
DISCOVERING A PROCESS

Let's begin by looking at how we go about discovering (uncovering?) a process. As ever, we shall start with a process architecture. The important thing about a *Riva* process architecture is that we don't need to know anything about how things are being done. In other words, we don't need to know how things are done before we structure our understanding of how things are done. The process architecture is deduced only and wholly from an analysis of the business the organisation is in, its 'subject matter', which we examined in Chapter 6. By going through the EBE and UOW analysis we reach an architecture which says 'If you are in this business, which has these units of work, then you must have these processes with these dynamic relationships.' That's a very powerful place to start the challenge of discovering how individual processes in the architecture are done. The chunking we achieve ensures we build on sound foundations.

So, when we start on the discovery of a particular process within the process architecture, we already have a name for it, and we know where it starts and where it ends. Let's remind ourselves of the basic scheme from Chapter 8:

9 – Discovering & defining processes

1 Decide on the objectives of the modelling: here we are concerned with discovering the process.

2 Brief ourselves by getting an overall picture, no matter how coarse, from a variety of sources.

3 Run one or more interactive workshops of those involved to draw up a RAD that meets the objectives of the modelling.

4 Use other appropriate sources of information.

5 Review, revise and validate the model using other inputs.

6 Use the model.

What adjustments do we need to make to this general approach? We shall listen in …

Tutor: Firstly, no managers are allowed to this workshop!

Pupil: They won't stand for that! Why don't you want them there? They feel exposed already because the process isn't defined, which presumably is their fault. And even if they have been delinquent, they'll want to know what is happening out there.

Tutor: The problem is that the people doing the work might be – shall we say – shy about revealing how they do it. So, by all means let's interview the managers as part of step 2, finding out what the managers think is happening or expect to be happening, or even know is happening. But we are more likely to get the 'truth' from the coal-face, where the dirt is!

Pupil: So, at the workshop we'll have in front of us a group who should not feel intimidated, and who will tell us how things really happen, good or bad, appropriate or inappropriate.

Tutor: Right. Secondly we'll draw a concrete model.

Pupil: If we want to understand the process shouldn't we be thinking abstractly rather than worrying about footling detail?

Tutor: We want to know what happens in reality. We might choose to stand back a little to understand the purpose of the actions and interactions. But it won't help to get too abstract too early.

Pupil: OK, but if we want to extract what is really going on, what sort of questions should we be asking?

Tutor: Easy: 'Do you really?' 'Why do you do that?' 'Why in that order?' 'Does everyone in that role do it the same way?' 'Has it always been done like that?' 'Where do things go wrong, and what do you do then?' 'What's behind doing it like that?' 'Why does the same person do those two rather different sorts of things?'

Pupil: So we're standing in front of some 'coal-face' workers; we know we are going to draw a process model. But where do we start?

Tutor: At the beginning of course! Suppose we're working on a case process. When we looked at the general scheme for running a process modelling workshop, we drew the main triggering event for the case at the top left-hand corner of the white-board and put it into the role where the process starts. And we drew the goal or commonest outcome of the process as a state somewhere in some role at the bottom right of the whiteboard. Once the trigger is on the board, we can just ask 'And then what happens?' and we draw what we hear. There will be discussion; there will be questions about whether we are getting too detailed or not detailed enough. As the analyst we must help the group make that decision at each step. Once a new blob is on the model, we ask the same question: 'And then what happens?'

Pupil: How do we handle the fact that processes are very rarely just a simple sequence? There will be all sorts of threads.

Tutor: Yes, there will. The trick is not to forget that we are leaving a trail of unfinished threads behind us – just use that little spring symbol to say 'We have to come back here and pick up this thread.' When we started the workshop we brainstormed all possible triggers and all possible outcomes, and shall need to check through those lists for more process before we finish.

Pupil: If this is true process discovery, there will be lots of surprises and it will be hard to do that pre-workshop briefing you talked about.

Tutor: It will. It means that we have to listen very hard at the workshop itself to what people are saying, and not to let anything go. It's why I suggested that there should be a separate note-taker present to catch all the points that might slip away as discussion continues.

We'll still also use all the other means we can to find out what is happening: all the methods we listed in Chapter 8 are at our disposal: interviews, observation, documents, terms of reference, personal objectives … anything that might suggest how things are actually being done. It all gets assessed for the model.

Choosing perspectives for communication

If we are modelling to communicate, the scope and perspective of our model need to be chosen so that the model tells people what they need or want to know, and omits everything else, or perhaps relegates it to 'the edges' of the model. One strength of a RAD is that we can decide at each point of the boundary how we want to draw it: with a black-box action, with an external event, or with an interaction, as we saw above.

At each point in the modelling we can ask 'Is it important to us to know this, or to know beyond this point?', or 'Does it help our understanding to get into this area?' In some instances we might say 'definitely not' and simply terminate that thread; in other instances we might say 'We don't need to know the detail but it would be useful to have some context' in which case we might 'summarise' the process beyond the area we are interested in. Again, on a RAD it is quite simple and natural to mix the level of coverage to suit your purpose.

We might have more than a 'neutral' aim of communication. We might want to scare. The process is a mess and we want people – management in particular perhaps – to realise that it is a mess. The keynote of our perspective here is *honesty*. Whatever it looks like we will model it: if it's a muddle, we will model that muddle. Drawing a tidy model of a messy process makes no sense if our aim is to reveal the mess.

KEY POINTS

When modelling a process for discovery

☞ keep the managers away;

☞ keep the model concrete;

☞ keep challenging;

☞ draw whatever is interesting and helpful;

☞ draw muddle if there is muddle.

PRESENTING A PROCESS DEFINITION

Suppose we want to use our model as a *formal* description. Perhaps we plan to use it as some sort of work instruction, in a Quality Manual for instance; perhaps it is to be used as the keystone of a Process Standard or a Standard Operating Procedure. We want to *prescribe* how a process is to be carried out: 'This is how we do things around here.' As with process discovery, we shall start with a process architecture to chunk the organisational activity and decide what processes there are. The question then is 'What perspective do we want to take of the process given that we want the model as a definition, to guide/instruct/require of people?'

Choosing perspectives for prescription

Let's look at the features we shall expect of our model:

☞ Our model will need to be sufficiently precise and detailed for us to be able to 'dictate' what we expect of people where we want to dictate to them, and to leave room for discretion where we want discretion to be used. Warning: most people prescribe too much and their process definitions stray into areas where people could and should be left to their own devices, discretion and initiative.

☞ Our RAD might need to be precise and testable enough to allow independent auditing of the way the process is actually being carried out in order to check for conformance. Remember that 'precise' does not mean 'excessively detailed'; 'testable' means there is a way of deciding afterwards whether it really happened that way.

☞ We shall need to take care that we do not say more than we need to. If we are going to be audited against the defined process, then the definition should only say what we want to be audited against. It is all too easy to over-egg the definition and then find we have prescribed more than is necessary for our purposes and the auditors are demanding we do things that we don't always want to be held to.

☞ The roles that appear on such a RAD will probably be *concrete* roles – either posts or groups – and the descriptions of actions and interactions will involve concrete mechanisms. Conformance is easier to establish if definitions are close to the real world.

☞ Independent verification or validation plays an important part in highly regulated industries, and in a RAD we have the mechanism – through roles – for showing this explicitly. We will expect to see separate roles that have responsibility for authorising actions, approving actions, and signing things off, and we will expect to see interactions with them that express the execution of that responsibility.

Using RADs in an ISO 9001 context

The international standard ISO 9001:2000 (*Quality management systems – Requirements*) is commonly used as a specification by organisations intending to manage the quality of their products or services through the management of their processes. In support of the quality management system (QMS) approach, the standard says that it

> … encourages organizations to analyse customer requirements, define the processes that contribute to the achievement of a product which is acceptable to the customer, and keep these processes under control. A quality management system can provide the framework for continual improvement to increase the probability of enhancing customer satisfaction and the satisfaction of other interested parties. It provides confidence to the organization and its customers that it is able to provide products that consistently fulfil requirements.

So, in an ISO 9001-compliant QMS we can expect to find the relevant processes defined in some way. RADs offer a useful way of presenting such definitions. *The TickIT Guide*, which is used for ISO 9001 certification in the software development arena, recommends RADs for modelling processes in a QMS (see *The TickIT Guide*, details at www.tickit.org).

Figure 9-2 shows ISO 9001's process view of quality management in an organisation. Customers, in whatever form they appear, have requirements which are realised by the organisation's processes and presented to the customer in the form of a product or service, hopefully to their satisfaction. 'Product realization' will be where our case processes will typically sit. 'Resource management' will be where our case management processes will typically sit. In Chapter 10 we shall touch on the question of where measurement sits so that process analysis and improvement can be done.

Figure 9-2 – ISO 9001's process view of quality management in an organisation
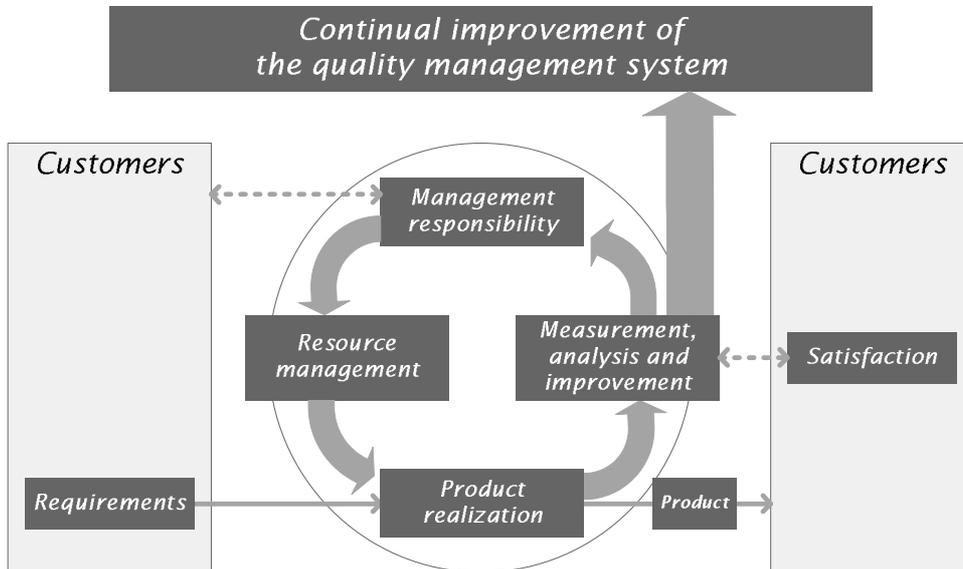
In an ISO 9001 context we will expect to see an emphasis on the control of the process, especially quality control and corrective action, in other words answers to questions such as: 'At what points in the process are checks on quality carried out?' and 'If a fault is discovered in the product or service what action is taken (a) to correct it, and (b) to ensure it does not happen again?'

I have seen RADs presented in a number of ways as part of the definition of a process:

☞ A RAD on its own.

One of the great secrets of RADs is that captions to symbols are drawn *next* to small symbols and not inside big symbols. This is such a small thing and yet it is a massive modelling convenience. Whoever decided that the lozenge should represent decisions in flowcharts deserves a hundred lashes: it's the most impractical shape you could choose. In a RAD the structure of the process is carried by the ordering of the blobs and becomes a great deal more visible. The captions are free to take up as much space as needed, alongside them. We are at liberty to put a small paragraph against, say, an action blob if we find it useful. Thanks to this remarkable property, a well-constructed RAD has little need of additional supporting material and yet it is very compact.

☞ A RAD with roles expanded in text.

There are, however, those who simply cannot work with a picture – something to do with left- and right-handed brains perhaps. An organisation that was preparing a Quality Management System to be used by many hundreds of people prepared the procedures in their Quality Manual in the form of a RAD followed by text structured by role. Each action, interaction and decision in a role was elaborated in text in a section following the RAD. Someone

carrying out a role could therefore see everything about that role in one place, both in the text and in the RAD.

☞ Text supported by a RAD.

I have also seen procedures in the form of traditional text description, 'supported by' a RAD. This is generally unhelpful. It is notoriously difficult to express a complex concurrent process using something as serial as natural language. I once took a 24-page text description of a process and reworked it as an A3 RAD, with practically no loss of information but with a huge gain in visibility … and during that reworking a host of problems were revealed that the text hid.

☞ A RAD on an intranet with hot links to ancillary material.

This is much more interesting. A RAD fits naturally in the world of web pages. Where supporting text is required, hot links to separate pages carrying that text can be provided from the relevant blobs on the RAD. Or links can lead to pro formas to be used, data systems to be accessed, and more.

☞ An enactable RAD.

This is perhaps the ultimate in process definition: the process model is loaded on a process enactment system that is able to 'drive' that process and all its actors. This is a major topic and an important one in the world of Business Process Management and we must leave it to Chapter 13.

KEY POINTS

When modelling a process for definition/prescription

☞ make the model a concrete model, especially if it is a work instruction;

☞ say things just once;

☞ only say things you want to be held to;

☞ use the RAD as the hub of the definition and hang other material off it.

SUMMARY

Figure 9-3 shows the general scheme that we use for discovering and defining existing processes. As with any process work, we start by preparing a process architecture to get that ideal chunking into processes. We can then take the process we are interested in and start work on that, confident that we are starting from somewhere appropriate. 'Observation of the process' covers all the different ways that we might decide how an process is being done.

Figure 9-3 – The general scheme for discovering and defining processes