

# 10 Analysing for process improvement

## THE IMPORTANCE OF SPRING-CLEANING

Two three-letter acronyms have dominated the world of business processes over recent decades: TQM and BPR.

This is not a book on TQM or BPR, both huge topics that have the concept of *process* at their heart, and that also contain many soft (but difficult) issues such as the management of change, visioning, motivation, culture and ethics. Our concern here is to look at how *Riva* can provide answers to process questions for both. In particular, I want to highlight how the concepts that *Riva* uses give us ways of looking for possible improvements.

Over time, our organisation and its processes become convoluted. A process that started out simple and clean has, somehow, become complex and messy. The business itself has changed, or the business environment has changed around it and it has not changed its process in response. Let's look at some of the ways that such complexities can arise.

### **'We won't let that happen again!'**

Bad experiences all too readily lead to extra twists being added to a process. A change might be made to plug a loophole, particularly if a mistake has at some time caused trouble, such as financial loss. Suppose that one day, we ship some equipment to a customer but the spares arrive late. Perhaps that has happened before but without any major repercussion – the spares have caught up a little bit later. This time the equipment failed soon after arrival and the spares were needed quickly ... but weren't there. Our customer sustained a significant loss of business and we had to compensate them. 'We're not going to let that happen again' vows our senior management, and extra steps and checks are added to the process to plug that possible loophole. Of course, on average each shipment is now held up that bit longer, just to ensure that the earlier mistake is never repeated.

Every time there is a process failure a new check, or double-check, or extra approval or sign-off is added to the process. Gradually the barnacles cover the once sleek hull of the ship, and friction builds.

### **'We won't give them another chance to mess us up!'**

Functional groups can become self-protective. Suppose group *A* provides materials to group *B*. And suppose *B* starts to get trouble with poor quality arriving from *A* and, as a result and too often, has to return material for reworking, having wasted its own materials and resources working on poor inputs. Group *B* is very likely to put in extra checking procedures at the interface with group *A*, and add elaborate hand-over and fault-reporting procedures, in order to protect themselves in the future.

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

Now everything is checked, everything is signed for, faults are recorded and reported formally, and there are procedures for tracking faults and cross-charging for the costs of faulty work. More complications for what should be a simple hand-over process. None of that new activity adds value, only cost.

### **‘The world has moved on!’**

The nature of our business can change. A process that was adequate under one set of market conditions can become quite ineffective when those conditions change, and, if the change occurs slowly, we shall add work-arounds bit by bit to try to overcome the increasing inadequacy of the old process.

Suppose we are in the business of making fitted kitchen furniture. Some years back when life was slower and customers less demanding, our order-filling process relied on a simple flow of manufacture, with orders and changes to orders being dealt with on a cyclic basis: on the first Monday of each month, the production team would look at the recent orders and adjust the production schedules. The whole thing satisfactorily revolved around that monthly cycle.

But our end-customers started demanding quicker delivery, and some stores started promising faster delivery to certain customers. Some orders couldn't wait for the monthly production meeting and so had to be handled as exceptions: 'short-circuits' and 'fast tracks' were added to the simple monthly cycle to handle the exceptions. But as time has passed and things have become more hectic, we have added more and more such work-arounds to the main cycle, and the work-arounds have become institutionalised, with the result that our monthly meeting is now more about finding out what is happening and handling the big bush fires, rather than planning. Planning is effectively being done out there in the work-arounds.

We need to replace the old cycle-driven process, which is now unwieldy and unresponsive, with an event-driven process, one which is clean and responsive. Instead of piling things up to be handled, we need to deal with things as they come in and make the necessary changes to production schedules on the spot. The current role structure and the interactions in it, as defined by the monthly cycle, could continue to exist in order to *monitor* the process from a managerial standpoint, but the day-to-day *control* of the process needs to become the focus.

### **‘We’re in another world!’**

Have we changed what we do? As well as teaching courses to full-time students, do we now offer remote-learning packages to part-time students? Have we decided not to outsource chemical assays to a service company any more, and instead to bring that function back in house? Have we passed all our invoicing to a finance house? Have we added generic pharmaceutical drugs to our product line as well as under-patent drugs?

In each of these situations we might have tinkered with existing processes to deal with the change in business. But knowing how we can characterise business through its EBEs and then go on to derive a process architecture in terms of its UOWs, we immediately see that we really should be thinking of changes at an architectural level, rather than at a process level. We may well have entirely new case, case management, and case strategy processes. Or we may need to delete some processes. Relationships between our processes may have changed and we might have made those changes simply by bodging.

If our business has changed in its UOW characterisation, we must start by reworking the process architecture.

### STARTING POINTS

There are several questions we need to answer before we can start work on the process architecture or any individual processes.

#### **What improvements are being sought?**

Our process or organisational improvement work will not reach its goal if we do not know what the goal is. It might be about reducing cycle times so that we can get to market faster, or close the sale more quickly, or answer a customer's query more quickly, or be able to offer a faster response than our competitors. It might be about reducing the cost of dealing with a repair, or putting a new PC on an employee's desk, or recruiting a new member of staff, or preparing a bid for some work, or managing the flow of call centre calls.

How we decide on our improvement goal is a topic outside the compass of this book – any book on process improvement from a high-level point of view will provide guidance. (Chapter 11's guidelines for designing a process are also relevant when improving a process.) So let's assume that we start with a clear idea of what improvement we are looking for.

#### **Where shall we focus our changes?**

##### *Big changes or little changes?*

At one end of the scale, we can completely rework the fundamentals of how we do our business: in *Riva* terms we would, for instance, be examining designed units of work and asking whether they are strictly necessary, and whether a different approach using different units of work would streamline things. We collect money for goods by issuing invoices. But why? *Invoice* is a designed unit of work. What happens if we delete it? We decompose our software development projects into work packages based on the stages of a development lifecycle – what would happen if we were to decompose them into work packages based on a functional breakdown of the system? (Software developers will recognise this as the step that was taken in moving from the SSADM development approach to the DSDM approach.)

In the middle of the scale, we might reallocate responsibilities amongst roles to reduce the number of hand-offs in preparing an invoice, thereby reducing the work involved and the time it takes to issue one.

At the other end of the scale, we might reduce the number of times a rework action in **Handle an invoice** is carried out by improving the layout of the invoice we send to customers.

We can look at process improvements right across this scale.

##### *Intent or mechanism?*

We can concentrate on *intent* or *mechanism*.

Are we doing the right things (intent)? The process might be highly efficient in what it does ... but it does the wrong thing.

Are we doing the things right (mechanism)? It might do the right thing ... but it does it inefficiently.

We shall need to decide whether we want to work with an abstract model (which is about intent) or a concrete model (which is about mechanism).

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

### *To-be or as-is?*

We can start straight into modelling the process as it should be after improvement, the to-be model. But if we know the process is bad, why draw a picture of it? Let's just start designing a new process. In some cases, we can safely ignore the existing process and our process modelling workshop will design the new one from scratch.

Alternatively we can start by examining the process as it is and look for improvements. If we expect the changes to be at a more detailed level, an as-is model might be the right place to begin. In yet other cases, we might not even know what is going on currently, and have to find out through an as-is model before we can start to look for opportunities to improve the process. And we should not assume that the current way of doing things is all bad – there might be very good reasons for certain things being the way they are and it is good to understand them and not make a mistake by reworking them.

### **Where should we be measuring?**

Measurement is a central part of business process management. We need measurement to know how we are performing, and to get indications of where things are not as they should be. Our process architecture gives us a vital separation between the process for dealing with one thing, and the management of all the things currently in progress. In particular, the time that elapses between the activation of a case process and the moment it reaches the desired outcome is easier to identify: it is obvious where we should insert measurement probes. More, the case management process is where such measurements can be collected, trends detected and responded to.

#### **KEY POINTS**

Before we begin we must decide

- ☞ what improvements are being sought;
- ☞ whether we can consider big changes and/or little changes;
- ☞ whether we are concerned with intent or mechanism;
- ☞ whether we need to start with the to-be or the as-is process;
- ☞ where measurement comes into play.

## **ANALYSING THE PROCESS ARCHITECTURE**

Let's begin at the beginning. Do we have the right process architecture? Do we have the right designed units of work? In a sense, this is the central question of BPR: are our processes aligned to our business and our customers? So our starting point is to build the process architecture for the part of the organisation we're concerned with and to examine that.

### *Focusing the searchlight*

Our first step is to decide what we mean by 'our organisation'. This team? The sales force and its customers? Production Engineering and R&D? The branch network? We saw in Chapter 6

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

that we can adjust the searchlight according to our purpose. Since the essential business entities characterise the organisation we are talking about, when we discuss which essential business entities are units of work for 'our organisation' we shall be moving the searchlight until it illuminates the right thing. When we have finished we shall have a process architecture that covers precisely the organisation we want to talk about. As a rule of thumb we can expect to extend across the boundary by one process simply to recognise the points where our processes interact with those outside our organisation, or its customer and suppliers.

The process architecture should – if we have done it properly – only include processes for *essential* units of work. We now add *designed* units of work: units of work that we have decided to have and use in order to do our business – in reality of course we may have spotted some of these when looking for the essential ones and have bracketed them out. Now we can add them back in, but – importantly – we distinguish them from the essential ones. We can think of this as adding mechanisms to an essential model. We saw in Chapter 6 how we can choose to include smaller and smaller units of work, as if we are seeing fainter and fainter stars between the brighter stars in the night sky. We must ask 'Is this unit of work too small to be worth bothering with?' Our answer will be driven by whether we think there is mileage in questioning its reason for existence.

### *Deleting designed units of work*

Knowing which units of work are essential and which are designed, we now examine each designed unit of work and ask 'Are we happy with this designed unit of work as the way of achieving our goal?', or 'What happens if we do away with this unit of work? What could we do to achieve the same goal, but without incurring the work?'

This sort of questioning makes us realise that *Invoice* is not the only way of getting payment from a customer: we could simply require them to pay electronically into our account against the delivery note when the goods arrive. Instead of batching delivery requests and treating a batch as a unit of work, doing all the deliveries in the batch together, what would happen if we simply dealt with each delivery when it arrived?

### *Outsourcing and insourcing designed units of work*

We can also ask 'Would this unit of work be better more cheaply/quickly provided by someone else as a service to us?' – the classic outsourcing question. If we have let the boundary of our process architecture cross into our supplier's domain, we can ask the opposite question: 'This unit of work that we currently get from an outside service, could we do it more cheaply/quickly if we did it ourselves?' – the classic insourcing question.

Units of work are precisely the things that we can consider out/insourcing. Roughly speaking, if we decide to outsource a unit of work, its corresponding case process, case management process and case strategy process all go to the outsourcing company. If we insource a unit of work, we must take those three processes back.

### *Task force or service function?*

In Chapter 4, we saw the difference between a unit of work that is provided by a service and one that is provided by a task force. Suppose we run a product company. For each product we prepare a user manual which needs to be printed. Clearly, *User Manual Print Run* is a UOW which will have a case process and a case management process. We now have a choice.

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

We can set up a group that will do manual printing as a service. This group will operate the processes **Do a User Manual Print Run** and **Manage the flow of User Manual Print Runs**. We shall demand that any product team that needs a print run for its user manual must use that service. Our goal is to make cost savings by having one set of facilities dedicated to printing manuals. But it does mean that a product team must knock on the door of **Manage the flow of User Manual Print Runs** and ask for a print run, and then take its turn in the queue.

An alternative is to allow each product team to arrange its own printing. It must have its own **Do a User Manual Print Run** and **Manage the flow of User Manual Print Runs**. The case management is probably nil. The team must set up a task force to do the printing. It does not need to join a queue. But it does have to solve the problem of facilities itself, perhaps by going to a local print shop or perhaps by buying its own and staffing them up itself. Our goal is to stop the printing of manuals becoming a bottle-neck in getting products onto the market, and we are prepared to pay some extra for that. In essence, we are building a process architecture in which those that use the outputs of a process operate that process – a classic re-engineering strategy.

Asking this question – shared service or not? – becomes very easy with a *Riva* process architecture.

### KEY POINTS

Using EBEs, we draw a boundary round our 'organisation' to include just the areas where we think change is necessary.

Starting with those EBEs we prepare a process architecture.

We examine *designed* units of work and ask if we are happy to continue with them in the future.

Task force and service functions are interchangeable.

We judge a unit of work 'too small' for the process architecture if we are content to leave how it is handled out of the study.

## PREPARING A PROCESS MODEL FOR ANALYSIS

There is no single way to prepare an as-is process model as a basis for analysing an individual process. Instead we have several options.

Our first model should be a concrete model. 'This is what we do.' 'These are the mechanisms, we use.' 'This is what happens on the ground.' As we prepare the model, we shall see the problems, some staring us in the face. Above all we need an honest model. We want to see it as it is, warts and all. In fact it's the warts we are looking for. We do not want a cleaned-up version.

We might then choose to derive an abstract version from the concrete model: 'This is what we are trying to do.' Comparison with the concrete model will give us further messages. The convolutions of the process on the ground will be all the more apparent when we compare it with the simple thing we are trying to achieve.

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

If we plan to take a quantitative view of the process, we shall need details about the duration of the individual actions and interactions and about the amounts of resource they use: people and people's time, quantities of input materials, and any other resources needed. In some cases we might already have this information from measurement, or we can estimate it. If we find a significant spread in the values we measure, we might find it useful to record that spread in some way: perhaps a sketch graph of the probability, or something along the lines of 'It normally takes two to three days, never less than one, and in rare circumstances up to six.' Once again we simply annotate the action or interaction on the RAD with whatever information we wish to keep.

We want to know where people don't use the prescribed process and why they don't; we want to know where the work-arounds have become necessary and perhaps even institution-alised; we want to know where rework has become a way of life; we want to know where the big hold-ups occur and why ... all the dirt. This is one of the reasons it can be important to exclude managers: we want a safe environment in which the unpleasant facts can emerge.

### ANALYSING THE PROCESS MODEL

If a process is in trouble or inefficient, the symptoms will often be recognised and understood by its actors. No process model can of itself provide the answer to a process problem: it can only act as a sort of searchlight on the process. The purpose of the process model must be to *reveal*: to reveal the process, the roots of its problems, and potential ways of attacking the trouble. Sometimes a RAD can reveal the nature of the problem and suggest a solution very quickly; something in the RAD is like a flashing light saying 'Here's your problem.' In other situations, revelation comes more slowly, perhaps as the right – revealing – perspective is homed in on.

We shall want to analyse the process from two points of view: *qualitative* and *quantitative*. To set the scene for a description of the sorts of quantitative and qualitative analysis that are possible on a RAD, let's first look at the different styles of process improvement that are possible.

There are four ways in which we can improve matters in a process:

- ☞ by 'point-wise' improvements to individual actions or interactions;
- ☞ by 'flow-wise' improvements;
- ☞ by restructuring roles;
- ☞ by realigning the organisational structure and the process structure.

In any given BPR or TQM programme, some mixture of these will be used. The first three are generally the domain of the TQM disciplines, concentrating on incremental change and incremental improvement; the last two are where BPR looks for radical change, in addition to the architectural level we looked at earlier in this chapter.

We take these in turn before considering the sorts of analysis of a RAD that would lead us to answers at these four different levels. It is also worth reminding ourselves once again that which of these levels we want to consider will very much determine the perspectives – and hence the RADs – we elicit and draw.

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

### Making point-wise improvements

This is the finest level of granularity in process improvement. Here we are concerned with increasing the efficiency (use of resources) or effectiveness (reliability and quality of result) of individual actions in the process.

We might choose to help the individual carry out an individual action by giving them tools to do their work. In an office environment, helping the individual to do their work increasingly means using information technology with personal productivity tools such as spreadsheets, organisers, and word processors for example. We might improve the way that interactions are carried out by providing e-mail, better-equipped meeting rooms, video-conferencing facilities, on-line discussion groups, or even an arbitration service. Which actions and interactions we choose to concentrate on when searching for point-wise improvements depends on where we are seeking the benefits of improvement.

#### *Reducing overall cost*

Any action consumes resources and some interactions do too. Suppose our concern is to optimise our process in its use of resources and hence its cost, and suppose we have annotated the RAD with the resource usage of each action and interaction. Deming identifies a number of ways in which a flowchart description of a process can be analysed to check for potential improvements in the process. There are straightforward corresponding analyses that can be carried out on a RAD:

- ☞ Work on the big hits first. Look at each action and interaction in turn to see which have significant resource usage and which therefore might yield the biggest savings.
- ☞ Rework equals waste. Look for case refinements concerned with checking for poor quality. Look at the frequency with which the thread that deals with faulty material or errors in previous work is followed. Trace back through the process and identify where that fault is introduced. What can be done in that action to reduce the likelihood of poor quality? And can the fault be detected earlier in the process, so that the cost of rework is reduced? These are all traditional TQM-style questions which the RAD can help us answer.
- ☞ Duplication equals waste. Do different roles do the same thing?
- ☞ Look for opportunities for error. Remove them.
- ☞ Examine the potential impact of supplier inputs. Poor quality will mean increased cost to fix. See what can be done to improve them.

If you are familiar with activity networks for planning purposes, you will know that one of the important ideas when working with them is the *critical path*, that sequence of activities whose total duration is the duration of the process. If an action on the critical path takes ten days longer then the whole process takes ten days longer. The critical path determines the overall duration of the process. We saw above how a RAD can be annotated with the duration of each action and interaction. It can be a simple manual activity to find the critical path. The presence of loops in our process makes the situation more complicated of course, but the principle remains. In complex cases, assuming the loops can be compressed or ignored and all but the

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

longest thread on each case refinement discarded, the (reduced) RAD can be transferred to a traditional project planning tool which will find the critical path automatically.

Actions – especially those on the critical path – can be externally focused, i.e. focused on the customer of the process; or they can be internally focused, i.e. supporting some internal function that is not directly relevant to delivery to the customer. We can go through the RAD classifying each process element (action or interaction) under one of three headings:

- ☞ It delivers value directly to the client of the process (mark these in green).
- ☞ It delivers value only to the organisation: internal reports, measurement, delegation, approval, etc (mark these in blue).
- ☞ It represents the existence of waste in some form: quality control, handling exceptions, correcting mistakes, apologising etc (mark these in red).

Ideally only green process elements should be on the critical path. Blue elements should be moved off the critical path if necessary. Red elements need to be eliminated of course, though this will require change to the other parts of the process to make it right-first-time.

### *Shortening cycle time*

All actions and interactions take time. We might be seeking a reduced time-to-process for an individual case: we want to get the result – the service or the product – to the customer as quickly as possible (this might be different from optimising the throughput of an individual worker). Critical path analysis will be a key tool here:

- ☞ Look at each action and interaction on the critical path. If we can reduce the duration of any of those, we shall, by definition, reduce the duration of the whole process (up to the point, of course, where another path through the process becomes the critical path).
- ☞ Look for the actions and interactions on the critical path whose duration has the greatest variation. In some situations, the perceived quality of a service can be increased as much by making the service *reliably* of a certain duration as by making it shorter in most cases. 'I don't mind if it takes three days, providing I can be sure it doesn't take any more.'

This works quite well except for situations where the process has loops, or repeated action in it, or threads that are only traversed under some circumstances, for instance for rework. Standard project planning tools cannot deal with loops and case refinements, so we must remove them first if we intend using such tools. We can do that in one of two ways:

- ☞ By adding some 'overhead' to the part of the process that, on occasions, needs to be redone.  
Suppose we have a reviewing cycle that is repeated until the thing being reviewed – a document say – is deemed acceptable. We might, for the purposes of critical path analysis (CPA), simply assume that the reviewing cycle is always done twice, and replace the loop by two reviewing threads in sequence. Or we might collapse the cycle into a single action.
- ☞ By treating alternative paths in some proportional way.

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

Suppose we are processing forms, and suppose type A forms take six days while type B forms take twelve days. About two-thirds of the forms are type A. We might say that the processing of type A and B forms takes eight days on average, thereby removing the case refinement.

This is not very satisfactory. We would prefer to leave the full process structure in place and deal with it intact with loops and alternative threads. The quantitative analysis approach of *System Dynamics* can help here (see for instance *An introduction to computer simulation: a system dynamics approach*, N Roberts *et al*, Addison-Wesley, 1983). In System Dynamics, a process is seen as a set of 'flows' of material between 'stocks'. The flows form a network which can include feedback loops and alternative paths for materials. PC-based tools for animating such models allow the flow rates to be specified as formulae and in particular to be made dependent on each other, on stock levels, and on external variables, on the passage of time, and on the time of year or time of day; those relationships can involve statistical probabilities, so we end up with a model that can be animated allowing us to determine the cycle time as a probability distribution itself: a much richer representation, especially if we are trying to understand the variability of the time a process takes. Such a model also allows the process's long-term behaviour to be explored: 'Is the process stable over time?', 'What happens to the throughput and cycle time over the seasonal rush?'

A RAD can be converted into a System Dynamics model by turning states into stocks, and actions and interactions into flows. Case refinements become split flows whose rates sum to one, whilst part refinements become joint flows in which the material is replicated on each flow. In practice, the RAD must be greatly simplified first as it contains a great deal of detail that does not need to be carried into the quantitative model.

Discrete simulation models permit similar quantitative representation of a process in terms of flows of 'stuff' from one action to another, but that stuff now takes the form of discrete objects with attached properties which can be handled differently by an action according to their attributes: green widgets are packaged in tens, red widgets in twenties.

To complete a quantitative model we will generally need to collect information about other 'influencing factors' that affect the quantitative behaviour of the process such as:

- ☞ the rate at which cases arrive for processing;
- ☞ the time of year (the rate at which work arrives might be seasonal);
- ☞ staff morale, which in turn might affect ...
- ☞ staff productivity;
- ☞ the numbers of staff available to carry out different actions;
- ☞ the availability of tools, machinery and other resources needed;
- ☞ consumer confidence;
- ☞ interest rates;
- ☞ weather patterns;
- ☞ population movements.

We can start to see that a quantitative model deals with factors which are strictly outside the sort of model we have prepared with *Riva*. Moreover, experience has shown that a quantitative model is most often beneficial if it is kept at a fairly high level. So we should not expect to

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

find quantitative solutions from a qualitative model, nor qualitative solutions from a quantitative model. Whilst there is some overlap between, say, a RAD and a System Dynamics model, we are better off thinking of them as complementary tools, each with its own things to tell us. That said, a process architecture can be a better starting point for a Systems Dynamics model as it operates at a more appropriate level than a RAD.

### Making flow-wise improvements

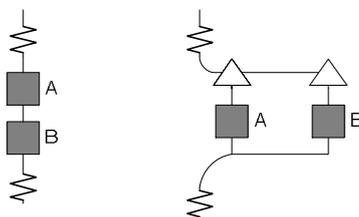
So much for dealing with individual actions, interactions and decisions. Given a set of roles and responsibilities, how can we improve the *flow* through the process? What changes can we make to the *order* of actions and interactions within a role in order to reduce the overall case processing time, or reduce resource requirements?

#### *Increasing parallelism/concurrency*

One obvious approach to reducing the overall elapsed time that it takes to process a case is to increase the overlap of activity, especially where this reduces the length of the critical path. This is an approach well known to those who plan projects using activity networks.

To reduce the overall elapsed time of a project, the planner looks for the critical path through the project and looks for ways to increase the amount of concurrent activity so that actions that were once done sequentially are now done in parallel. The RAD equivalent is to move from the process fragment on the left-hand side of Figure 10-2 to that on the right-hand side. The assumption is of course that *B* does not depend on *A* and that there are actors available to do *A* and *B* concurrently.

Figure 10-2 – Increasing parallelism in a role's actions



The effect of this is to change the elapsed time of this fragment from the sum of the time it takes to do *A* and *B* to the maximum of those two times.

Our inspection of the critical path might suggest that there is no need for *B* to be on the critical path at all and moving it off will be our immediate response.

It is a good discipline to ask 'Do we really have to do these things in sequence?' whenever we see more than two black boxes in a row.

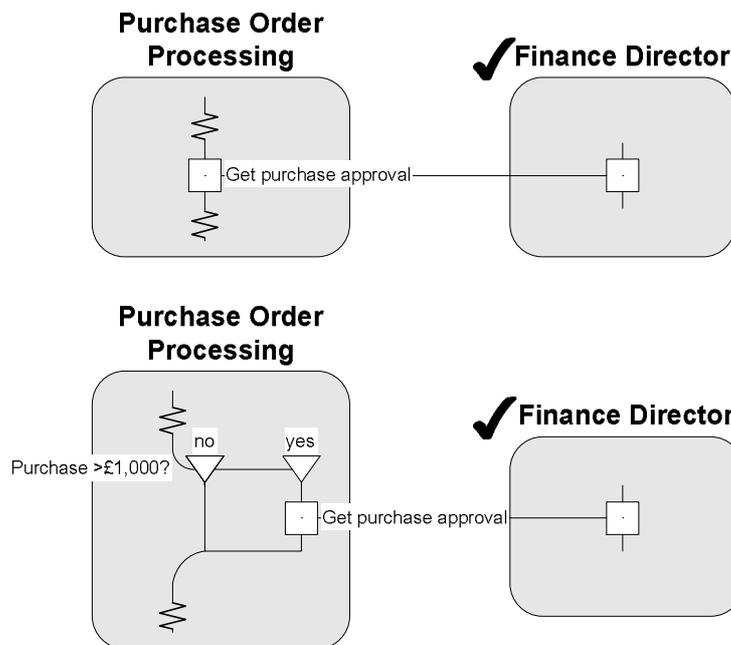
#### *Applying the 80:20 rule – from generalists to specialists*

Does every case that goes through a process need to undergo the same processing? Does every purchase order need to be seen by the Finance Director? Could we limit the ones needing the FD's approval to those over a certain value? If we can do this, we can reduce the *average* time it takes for a case to be processed, if not the maximum. It might look as though we are

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

complicating the process, and indeed we are, but the effect can be positive, assuming that the extra decision-making requires very little time. We might replace the top process fragment in Figure 10-3 by the bottom fragment.

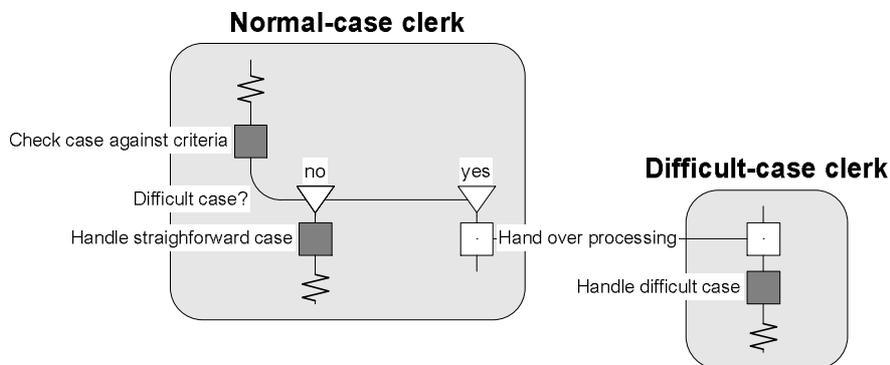
Figure 10-3 – Complicating a process for improved average speed



We can take this further. Rather than routing both simple and complex cases through the same case workers, thereby requiring all case workers to have the same high level of skill so that they are able to handle any case of any degree of difficulty, we can consider filtering out difficult cases at some point during their processing and passing them to a smaller number of expert personnel. The personnel who handle the run-of-the-mill cases no longer have to have the same degree of skill and could therefore be a less expensive resource. Once again the process becomes slightly more complex to incorporate the filtering, but the benefits could outweigh the costs. The equivalent in RAD terms is shown in Figure 10-4.

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

Figure 10-4 – Splitting cases by difficulty



As ever, we can also ask the opposite question: rather than having specialists and filtering mechanisms and hand-offs to serve them, can we not have generalists who offer a one-stop-shop?

### *Planning for success*

'Planning for success' is a technique that we can use where shortening elapsed time is of paramount importance and where we can countenance wasting resource if the potential time gains are great enough. A product development process is a typical example. Getting to the market-place earlier can mean increased product life and earlier revenue flow. A business might consider the possibility of doing work that has to be thrown away if development is abandoned, if it means it can get the product to the market-place earlier in the cases where development is successful. A pharmaceutical drug company might well build a production plant for a new drug before they have obtained regulatory approval to sell it.

Suppose that the 'sensible' way of doing things is

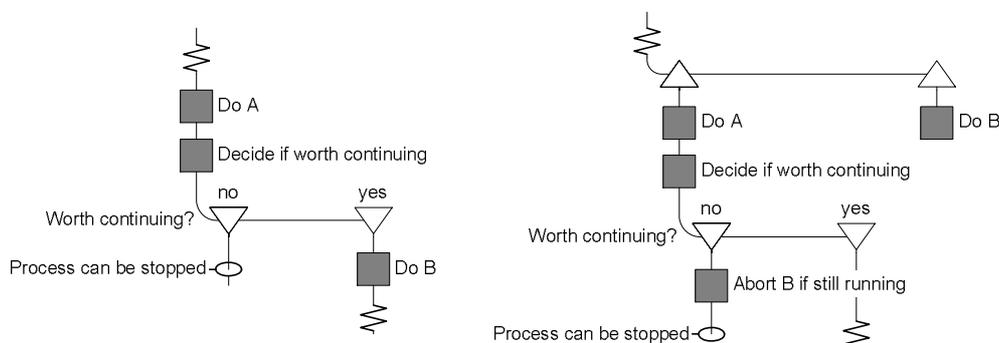
- 1 Do action *A*.
- 2 Decide whether it is worth continuing.
- 3 If it is, do action *B* and continue with the case; otherwise abandon the case.

If we decide at step 2 that it is not worth continuing with the product, we have not wasted effort doing action *B* to no avail. The time it takes to get through this is of course the sum of the time to do action *A*, the time to make the decision (typically small in comparison), and the time to do action *B*.

If we plan for success, we start actions *A* and *B* at the same time. When *A* finishes we make the decision. If the decision is 'go' we let *B* continue, otherwise we chop *B*. This is equivalent to replacing the left-hand fragment in Figure 10-5 by the right-hand fragment. (Remember that an action can be terminated for a number of reasons: in this case either *B* finishes naturally because it was allowed to run to completion, or it was aborted. How we proceed at the end of the part refinement depends on whether *B* was aborted or not.)

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

Figure 10-5 – Planning for success



The upside of this change is that, whenever the decision is 'go', we save the elapsed time of the shorter of doing action *A* and doing action *B* – we get to market that much faster. The downside is of course that, whenever the decision is 'no go', we have wasted whatever has by that time been spent doing *B*. Whether the upside is considered bigger than the downside is clearly going to change from one situation to the next, but the RAD provides a way of exploring the possibility.

The pharmaceutical industry offers many examples of the potential benefits of planning for success. Each extra day that a successful drug is on the market can mean a considerable amount on the bottom line; this can justify risking wasted effort on actions that prove to be unnecessary.

### *Checking for coherent flow of 'stuff'*

We can annotate the actions and interactions in a process model with the 'stuff' – materials or information – they need and the 'stuff' they produce (dare I say their inputs and outputs). Given that annotation, we can check that the flow through the process is *coherent*, i.e. that stuff gets around the process in one way or another from the roles that produce it to the roles that use it. It is not unusual for the work-arounds that we find in a process to be there simply to cope with the inadequacies of the formal process when it comes to moving stuff between the roles. 'Why is that interaction there? Well, I normally end up going back to the originator because, for some reason, the information asked for in section 5 of the report is rarely enough for me to do my bit.' Those work-arounds, as is often the case, give us clues for improvement of the process.

I've lost count of the number of times that someone in a process modelling workshop has said 'I always wondered why you sent me that stuff – I've always binned it.' With any luck they will then say something like 'If it came with the summary sheet, it would save me having to reconstitute the figures myself every time, which is all I want – can you do that easily?'

We can ask such questions as:

- ☞ Is anything generated but never used?
- ☞ Are all received grams used by the receiving role?
- ☞ Is all stuff needed generated?

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

☞ Is all stuff generated before it is needed?

### *Catching faults earlier*

If someone makes a mistake somewhere in the process, it might not be found until later on, and then correcting can involve unwinding things, and tracing back to the source to get the fault corrected – complications in the process and delays in the processing.

Earlier fault detection can reduce the likelihood of faults getting through to later stages and can reduce the cost of correcting them. We can check the RAD for places where faults are detected and see if the detection can be moved to an earlier point, nearer the source.

Look for iteration, where a passage of process is repeated until something is right, or of the desired quality. See if the number of iterations can be reduced by inserting more up-front, preparatory work, for instance.

#### KEY POINTS

Examine the critical path.

Look for places where there is rework, work-arounds, or hold-ups.

Consider ways to improve the effectiveness and efficiency of individual actions and interactions.

Consider ways to shorten timescales by adjusting the flow of activity within roles.

Look for ways of catching faults earlier.

### **Restructuring roles and interactions**

As we map the process we reveal a structure that is the result of perhaps decades of change in people's job descriptions, in the way the organisation is structured, in the business of the organisation, in how the organisation likes to treat its staff, in the principles the organisation holds to, in the use of technology, in unionisation, ... a host of factors for change. The precise content of each role – particularly where it is defined as a post or group – will not be wholly rational if we view it dispassionately. But from our role-centred RADs we can look for ways of rationalising the structure of roles, in particular by moving actions between roles, combining roles and so on. We might explicitly try to reduce the number of interactions that are necessary to make a process work, and this would typically mean restructuring the roles and what they do.

In summary, the RAD becomes a way of challenging both the culture and the allocation of roles and responsibilities (both fixed and dynamic) to different roles in the organisation.

How can we detect these sorts of possibilities? With abstract models.

### *The process as pizza*

If we could design our process without worrying about who was to do what, or how the process would work with our particular organisational structure, we could imagine coming up with the perfect process in the form of a nice circular pizza. It would be a tidy, simple structure with only the absolute essentials. But in real life we have to take that nice circular pizza and

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

cut it up between the different people we employ in the different parts of the organisation. When we divide up the pizza and pull the sections apart we find we have a mess of strands of mozzarella on our hands. The more pieces we cut, the more the strands.

Each strand is an interaction. It does not add value. It is only there because we have cut the pizza that way. And the obvious observation is that by cutting the pizza another way – by changing our organisation so that the pizza pieces are fewer when the pizza is cut over it – we can simplify things and have less of a mess.

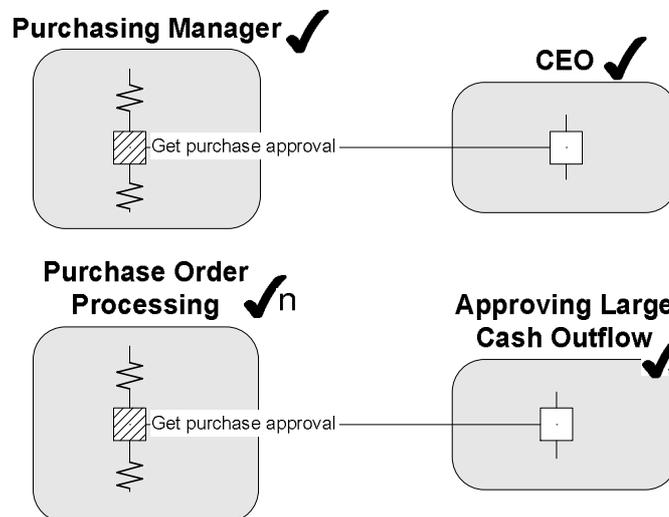
To understand that key relationship between the process and the organisation – hierarchies, management structures, job titles, etc – we need to use concrete and abstract process models.

Suppose we run our as-is modelling session using a concrete model. We are in effect showing how the many responsibilities in the process are divided up amongst the posts and departments. Division induces interactions. Now suppose we examine the concrete model and draw the equivalent abstract model, where we only concern ourselves with intent and we ignore mechanisms. The difference between the two models will point up where the real-world partitioning into posts and departments has unnecessarily complicated the simple – and ‘natural’ – division of responsibility.

### *Moving responsibilities between roles to reduce hand-offs*

Take as an example the process **Purchase an item of capital equipment**. We have found out that the CEO is asked by the Purchasing Manager to approve purchase orders. To label the role ‘CEO’ as shown in the top part of Figure 10-6 would give a correct model of the concrete process, of the ways things are actually done.

Figure 10-6 – Can we move this responsibility?

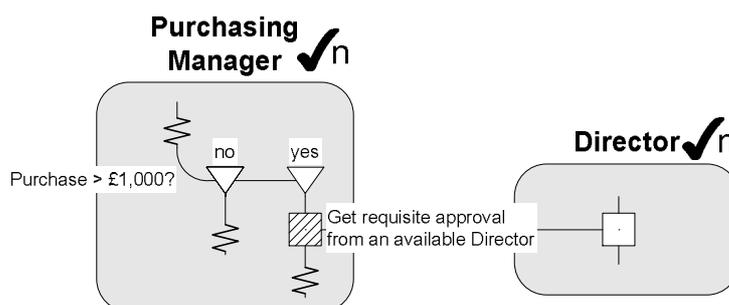


## 10 – ANALYSING FOR PROCESS IMPROVEMENT

But of course the CEO box on the RAD does not capture everything done by the CEO, only how the CEO contributes to the process under consideration. We might then look at that contribution and try to characterise it in some way. We might decide that what the CEO is actually doing is giving approval on behalf of the Board, taking into account the cash position of the company and the priorities for contending calls on that cash; the CEO has access to the knowledge of what else is going on in the company and the priorities necessary to make the decision. It just so happens that the CEO is the post currently designated to make that decision. But in our desire to speed up decision-making we might be prepared to move that responsibility around.

By concentrating on the responsibility – the abstract role *Approving Large Cash Outflow* in the bottom part of Figure 10-6 – rather than the current holder of that responsibility, we allow ourselves to think more radically about whether, for instance, *any* Director could undertake the decision if the necessary background information were available to them too. This gives us a candidate for restructuring. We might end up with something like Figure 10-7 in which we have also allowed the Purchasing Manager to commit to purchase orders up to £1,000 without getting approval.

Figure 10-7 – Fewer and easier hand-offs



Let's take a look at what happens in the Reception area of the building. It might be tempting to regard the work of the staff in Reception as a process in itself: there are clearly some people there doing things during the day, the same people in the same area of the building. But this would be to fall into the trap of associating a group of people with a process. In fact, of course, the Reception staff undertake a variety of roles that contribute to a variety of processes in the company. For example, they contribute to our *Purchase something* process by acting as Goods Inwards for certain sorts of goods, receiving deliveries, signing for them, determining who in the company the delivery was for, notifying them, matching delivery with the purchase order, and so on. They also contribute to our marketing processes insofar as they greet visitors and operate the telephone switchboard. Moreover, they contribute to the training processes in the company by gathering the names of those wishing to attend internal seminars, arranging food, AV equipment, rooms, etc. We might label these roles as *Small goods receiving*, *Visitor greeting*, *Telephone answering*, and *Seminar logistics handling*. Doing this makes it easier for us to see how our mapping of such roles onto functions – Reception – can be analysed and perhaps changed.

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

Let's generalise this. There are basically four steps:

- ☞ Draw up the as-is process using concrete roles.

We make the roles match the actual posts or functional units that exist in the organisation. We are going to assume that the organisation will remain the same – the same posts and functional units – but that we want to explore how we can divide the process over them in a way that reduces the number of mozzarella interactions – hand-offs – that are necessary. Remember that every interaction is a potential waste of resource, a potential delay, and a potential buffer and conflict point.

- ☞ Deduce the abstract roles underlying the process.

The actions and decisions have, over time and for a variety of reasons, ended up being the responsibility of the various posts and functional units shown. Once we have the as-is concrete model we look for the abstract roles within the process. We can then either redraw the process in its fully abstract form, or simply mark them on the as-is RAD by drawing lines around actions and decisions, grouping them into abstract roles.

- ☞ Identify ways of re-allocating actions and decisions in the abstract roles to the concrete roles.

We decide where actions and decisions could be moved between roles in order to reduce the number of interactions necessary. This is not a mechanical process: it requires experimentation, and it can also require the organisation to rethink some of its policies, particularly in areas to do with delegation.

- ☞ Define the new concrete roles.

Depending on the criteria we are using to choose the 'best' re-allocation of actions and decisions, we leave ourselves with a new, restructured process in which we might still have the same roles as we started out with, but now with their responsibilities changed. Alternatively, we might have created new posts that combine responsibilities more efficiently.

The key to all this is that the concrete and abstract models are helping us to look at the relationship between the process and the organisation.

### *Relaxing/strengthening approval and authorisation*

A RAD is excellent for revealing the approval and authorisation mechanisms that the organisation has put in place. They can all be questioned, with a view either to strengthening the mechanism or relaxing it. 'Does the Finance Director *have* to see all purchase requisitions?', 'Would it be better to introduce the requirement for senior management approval at this point, rather than waiting till further down the process?', 'Should this sort of situation be escalated to a higher management level than it is now?'

We can recognise five levels of relationship or delegation between manager and managed (William Oncken, *Managing management time: who's got the monkey?* Prentice-Hall, 1987 – an excellent book):

- 4 Wait until you are told.
- 5 Ask what to do.
- 6 Recommend what should be done.

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

- 7 Act but advise at once.
- 8 Act on your own, reporting routinely.

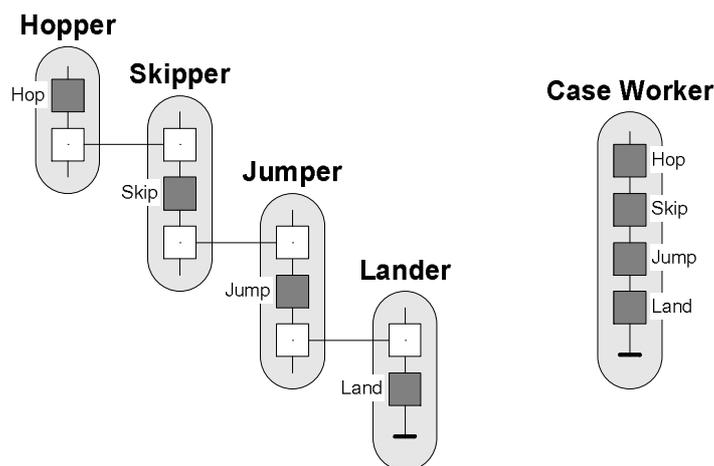
We can ask what the current interactions tell us about the current relationships, and ask whether they are over- or under-strict.

### *Specialists to generalists – the ‘case worker’*

When we divide a single task over two people we generate a need for interaction between them, across which the task will flow. When a case or gram moves from one role to another via that interaction, we will often find a *buffer*. If the respective roles process their own cases – units of work – according to their own cycles, some way is needed of ‘decoupling’ the cycles at the point where they intersect. That’s a buffer – it’s what happens in a case management process: requests arrive and go into the planning melting pot, possibly to be batched with others, possibly to be put at the back of the queue. Concretely, it can be folders accumulating in an in-tray, unread messages in someone’s electronic mailbox, or all the other ways we have of piling up work to be done. Buffers introduce delays, break the flow of processing and make tracking and monitoring difficult.

The flow of work for a single case can appear very simple, perhaps of the kind shown in the left hand-process fragment in Figure 10-8. Apparently each role makes its contribution to the processing of a case and passes it on to the next role down the production line – rather like a bucket chain at a fire: each person in the chain turns to their left, grabs the bucket and swings it to their right. Provided everyone is synchronised it works fine. The bucket (i.e. the case) moves smoothly down the chain. Most case processes are not like this; the roles take different or varying amounts of time with their contributions and different numbers of people are put in to deal with each stage to even out the flow. In other words we end up introducing case management at each interaction, and each case management process introduces its own buffer to smooth the flow: as a result buckets pile up between certain individuals and some people then hand on three buckets at a time.

Figure 10-8 – From specialist stream to case worker



## 10 – ANALYSING FOR PROCESS IMPROVEMENT

As each role (often a department) tries to optimise its performance, so the overall flow can start to become uneven as the compartmentalisation works against optimisation of the overall process. Many traditional production line industries have, for many reasons including motivation, moved away from the bucket chain approach and introduced 'case workers' who take responsibility for the entire handling of a case, as suggested by the right-hand side of Figure 10-8. In automobile manufacture for instance, a production 'cell' might take a car through from chassis to final inspection, working with it right down the line.

*Is the problem with the case or the case management?*

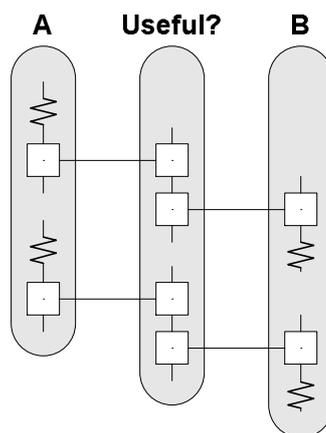
A case process itself might run more or less well, but if the corresponding case management process is ineffective – or even totally absent – then individual cases can conflict with each other, priorities can be poorly understood and ineffective, and people might have to make do to get by. One would expect to see ways of resourcing individual cases, managing changes in the case load, handling sudden changes in the priority of individual cases and changes in the nature or content of individual cases. If these are not adequately dealt with there could be trouble.

That said, the word 'adequately' is an important one: it can also be the situation that a case management process is *too* cumbersome and is unnecessarily bureaucratic. The modelling will help us to decide whether things are under- or over-managed.

*Is everyone doing something useful?*

In the most extreme cases, we might observe roles which have few or no actions of their own, and which seem to be only third parties in other people's interactions. These roles may be redundant, adding no value, and only slowing things up. I have seen a role which seemed to sit between two organisational units and passed stuff between them. It was quite hard to see what value was added en route, and some hard questions were asked about that role. Figure 10-9 suggests the sort of thing we might see.

Figure 10-9 – Is this intermediary role adding anything?



## 10 – ANALYSING FOR PROCESS IMPROVEMENT

### *Analysing interactions – hand-offs*

The interactions in the process are there because our roles – organisational units – need to collaborate, and the way we define our organisation determines the way the collaboration works: we get more or fewer mozzarella strings. By looking at the interactions we can therefore get some clues as to how the organisation might or might not be getting in the way of the process.

Auramaki *et al* examine office communications using discourse analysis, and their analysis of ‘illocutionary acts’ is a useful way of examining interactions in RADs (see ‘Modelling offices through discourse analysis: the SAMPO approach’, Auramaki *et al*, *Computer Journal*, 35, 4, August 1992).

We can think of an office as a communication network which creates, maintains, and fulfils commitments. In a RAD, commitments happen in interactions, so we should find it useful to analyse these to see what sorts of commitments are being made. (One interaction might involve several separate commitments.) If we refer to the role that initiates the interaction as the ‘speaker’ and the other roles as the ‘hearers’, we can identify five types of commitment:

- ☞ *assertive* – the speaker is noting an actual state of affairs;
- ☞ *directive* – the speaker is asking the hearer to do something;
- ☞ *commissive* – the speaker is committing to do something;
- ☞ *declarative* – the speaker is bringing about a new state of affairs;
- ☞ *expressive* – the speaker is expressing attitudes or feelings about the state of affairs.

An interaction carrying an *assertive* commitment would be one in which roles are being informed about the completion of some action: ‘The plan has been issued,’ ‘The invoice has been paid,’ ‘The budget has been finalised.’ We can ask ‘Do the hearer roles really need to know?’ Is *B* interested in the assertion?

When the commitment is *directive* we would expect to see the hearer role doing something as a result (or else why are they being told?), and at some later point reporting back in an assertive commitment to the speaker role, to confirm completion. We can check this: is the interaction *Request report on fault statistics* matched by a subsequent interaction *Deliver report on fault statistics* in the other direction? We can also check that each hearer is able to carry out the requested action (i.e. has the authority and resources, knows what is to be done, and is willing to accept instruction from the speaker). And finally we can check that the speaker has the authority to give these instructions.

When a speaker makes a *commissive* commitment (i.e. promises to do something), there should be a later interaction in which the speaker makes an assertive commitment to the hearer, confirming completion of the promise. We can check that this interaction occurs: is the interaction *Agree to supply necessary resources* matched by a subsequent interaction *Confirm necessary resources are available*? And we can check that the speaker has the authority to make the commitment in the first place.

An interaction carrying a *declarative* commitment might be hard to distinguish from one carrying an assertive commitment, but we would expect in the declarative case that all the roles involved would see this interaction as a trigger to get on with some new action: they already know *what* to do; this interaction is giving them permission to proceed. We might find the

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

interaction *Production can now start*, or *The specification can now be relied upon*, or *Approval to proceed has been given*. We can check whether all roles know what to do.

(*Expressive commitments* are outside the sort of modelling we undertake with RADs, so we shall not examine them further.)

In Chapter 2, we looked at how any interaction could be regarded as a ‘conversation for action’. Any interacting pair of roles in our process model will be engaged in such a conversation, possibly several. We can use the template in Figure 2-38 to analyse the relationship between two roles by identifying the separate interactions and mapping their components onto that framework. Are there conversation components that are missing? If so, should they be present? If they aren’t, would the process be improved if they were added?

Finally, we can ask the following general questions to test whether we have the right interactions between roles, and, possibly, whether we have the right roles:

- ☞ Is there a pair of roles with a mass of fine interaction? This might indicate a poor division of activity between the roles or a confusion over objectives.
- ☞ Are there roles which have the same type of interaction with many other roles? This might reveal a pervasive function that should be dealt with separately.
- ☞ Is the concrete form taken by an interaction ‘long’ in some sense? It might be inefficient.
- ☞ Does an interaction have a buffer in its concrete form? Buffers slow up interactions. The existence of a buffer might reveal a designed unit of work and its case management process that we had not recognised before.
- ☞ Does all the checking, authorisation, referring back, copying for comment, input, or approval, etc help in the achievement of goals?

### KEY POINTS

Draw the as-is process using concrete roles.

Deduce the abstract roles underlying the process.

Identify ways of re-allocating actions and decisions in the abstract roles to the concrete roles.

Examine the interaction structure to see what it tells us about the division of responsibilities across concrete roles.

Consider restructuring roles to improve the interaction structure of the process.

Check for buffers and the hidden units of work whose flow they manage.

## CASE STUDY

In organisation Q, a sizable mechanism had evolved over the years to deal with requests made to a service group in the organisation. Dealing with these requests – which involved making and delivering specialist goods to a hard and fast timescale and specification – was a top priority for the organisation and hence for the service group. When the requesters said ‘Get these

## 10 – ANALYSING FOR PROCESS IMPROVEMENT

supplies to this location on this date' it had to happen. To make life that little bit more interesting the requesters were, for perfectly valid reasons, in the habit of changing the details of their orders at any time, and making new requests at short notice. Arms and legs were broken by the service group to make sure the requests were correctly satisfied, and invariably they were. But the stress was increasing and it was not clear that the mechanism, or the people, could stand a strategic change in the business that would lead to larger numbers of smaller requests. What was to be done to reduce the stress? Point-wise improvement – increasing the productivity of the actions required to make the supplies – was not the issue. Some other solution had to be found.

We modelled both the case process – how an individual request or change to a request was satisfied – and the case management process – how the stream of requests was handled, prioritised and so forth.

The main issue was the fact that satisfying a request required contributions from four main teams, each struggling to solve difficult technical problems to fulfil its contribution. Each new request or change to a request meant getting all four teams to rejig their schedules whilst keeping all previous requesters happy.

In the old days, when life was slower, this 'negotiation' of changed schedules could be dealt with by one of the regular meetings held by different committees: there were four altogether, dealing with the management of the process from the day-to-day tactical level to the long-term strategic level. But as time pressures had increased, people had found that the formal process was too slow, and during the modelling of the case management process we found many work-arounds where people would try to sort out a solution and then get it signed off at the next appropriate meeting. This became evident from the RAD which showed, buried in a mass of work-arounds, the original four time-driven cycles. What the RADs also revealed was a mass of bilateral interactions by which the four teams attempted to negotiate changes to their separate schedules, outside the formal meetings.

These insights into what was happening led the service group to look for a new way of running the case management. What was needed first of all was an event-driven process rather than a cycle-driven process: things could no longer be held up until the next cycle – they had to be dealt with as and when they came in. The need for a quick resolution was now the norm not the exception. The unit of work was not the *Monthly meeting*: it was the *Request* and the *Change to a request*. Indeed, the latter had not been recognised at all and was treated as an irritation. Moreover, the negotiation of schedules between the four teams needed to be made the responsibility of a single new body (role) that would replace all those bilateral interactions with a single negotiation, recognising that many bilateral negotiations led invariably to whirling around in circles ... and stress.

The change of process pattern and the introduction of a new role simplified the case management considerably, resulting in less stress, continued satisfaction of the requesters, and increased ability to handle future work-loads.

SUMMARY

Figure 10-9 – The general scheme for process improvement

