

BUSINESS PROCESS MANAGEMENT A RIGOROUS APPROACH

MARTYN A OULD



Copyright © 2018 Martyn A. Ould

Originally published in 2004 by the British Computer Society.

The text is now made freely available for general use and distribution.

Please respect this and do not republish for gain!

Each chapter (listed below) is made available as a separate PDF.

Contents

0 Introduction

Preface (2004)

Preface (2018)

Abbreviations

What processes are and why we need to model them.

1 Basic process concepts

What are the features of real-world processes that we want to reflect in process models?

2 Modelling a process

Provides all the 'vocabulary' necessary to represent a single process in a Role Activity Diagram (RAD).

3 Dynamism in the process

Highlights the levels of within-process concurrency that can be captured in a RAD.

4 Process relationships

Examines the two types of dynamic relationship that processes can have and how to represent them in RADs.

5 The three basic process types

Describes the three main types of process – the case, case management and case strategy processes.

6 Preparing a process architecture

Describes how to construct the process architecture of an organisation, a concept of central importance for any work with processes.

7 Dynamism in the world

Shows how the process architecture captures all the between-process concurrency in the world.

8 Managing the modelling

Provides guidance on running a process workshop and conducting interviews in order to prepare a model of a process, for whatever purpose.

9 Discovering & defining processes

Covers the practical use of the approach in determining what processes an organisation has, in eliciting those processes (onto RADs), and in the use of RADs in Quality Management Systems, tying into ISO 9001.

10 Analysing for process improvement

Covers using the approach at both the architectural and the process level for asking questions about processes and their performance, and for driving tactical process improvement.

11 Designing a process

Covers the design of a new process architecture and new processes starting from a blank sheet of paper.

12 Processes and information systems

Covers the role of the approach in constructing an IS strategy for an organisation, and in the design of traditional information systems.

13 Processes and process systems

Covers the role of the approach in developing Business Process Management systems in which agile and mobile processes replace static data structures.

Preface (2004)

Much has happened in the business process world since I wrote *Business Processes* (Wiley, 1995). The recent growth of interest in the important emerging technology of Business Process Management and the experience I have had using the approach described in this book have together conspired to force me back to my desk to write this current work about the *Riva* method.

My grateful thanks go in particular to Stewart Green and his colleagues (University of the West of England, Bristol), Tim Huckvale (Charteris), Dr Steve Hutson (Calcis Consulting Group), Derek Miers (Enix Consulting Limited, and Director of BPMI.org), David Perrin (Holonix Limited), Clive Roberts (Co-ordination Systems Limited), Howard Smith (Co-chair of BPMI.org and CTO Computer Sciences Corporation European Group), Professor Bob Snowdon (University of Manchester). Michael Jackson's work on system development has been very influential. Their help on the text, and conversations with them and many others over the years have helped me where private cogitation has failed. Many of their thoughts, ideas, and words have found their way into the melting pot, and it is hard now to identify, let alone acknowledge, all of them. Insights arise from many sources, not least from application to real problems, but also from the courses I have given on *Riva* and the challenges of course attendees. Any nonsense is of course mine.

I would also like to thank a number of people for permission to use material: Deloitte for allowing me to carry forward some material from my 1995 book; Howard Smith for quotations from his and Peter Fingar's influential book *Business Process Management: The Third Wave* (Meghan-Kiffer Press, 2002) and other articles; Tony Solomonides at the University of the West of England, Bristol for permission to use the Sentinel case study in Chapter 7; the University of the West of England, Bristol for permission to use case study material from work undertaken with the CEMS Faculty.

My website at www.veniceconsulting.co.uk contains free downloads of some resources to support the use of *Riva*. Questions, criticisms, and contributions are welcome and should be addressed to riva@veniceconsulting.co.uk.

Martyn Ould
Hinton Charterhouse, 1 May 2004

Preface (2018)

The paper version of this book, originally published in 2004, went out of print in 2017. No doubt copies will be available on the second-hand market, but I felt that it would be worthwhile for the entire text to be more generally available. I have therefore reformatted it for continuous reading (rather than facing pages) and prepared it as a PDF for the web.

However, I have resisted the temptation to bring the text up to date, to 2018 – it would only go out of date again in a few years. So some of the contextual material no doubt sounds odd today, if not old-fashioned. I make no apologies for that.

If you find what you read useful and would like to make a donation please do so via PayPal to mao@theoldschoolpress.com – £5 or \$5 would be a nice gesture!

I am now retired from this field but will generally be willing to answer questions addressed to mao@veniceconsulting.co.uk.

I hope you find what follows helps you in your work as it did in mine.

*Martyn Ould
Seaton, 1 May 2018*

Abbreviations

BPM	Business Process Management
BPML	Business Process Modeling Language
BPMN	Business Process Modeling Notation
BPMS	Business Process Management System
BPR	Business Process Re-engineering
CMP	case management process
CP	case process
CSP	case strategy process
ERP	Enterprise Resource Planning
IS	information system
PAD	Process Architecture Diagram
QMS	Quality Management System
RAD	Role Activity Diagram
RML	Requirements Modeling Language
SAS	System Attribute Specification
SOP	Standard Operating Procedure
SPML	STRIM Process Modelling Language
SSADM	Structured Systems Analysis and Design Methodology
STRIM	Structured Technique for Role Interaction Modelling (the precursor of <i>Riva</i>)
TQM	Total Quality Management
UML	Unified Modeling Language
WFM	Workflow Management

Introduction

PROCESSES ARE BACK ON TOP

Something very different is happening in the world of process. In recent decades business processes have ridden two 'waves'. A third is upon us.

In the first wave, processes existed purely in terms of custom – 'it's just the way we do things round here' – or, if they were lucky, lived a private life inside policy and procedure manuals. The emphasis was towards 'small' pieces of organisational activity such as claiming expenses, approving budgets, or perhaps standardised and often-repeated sequential activity such as the analysis of a laboratory sample. If processes were modelled it was with simple diagrams such as flowcharts. And perhaps flowcharts were enough for simple sequential processes. The pictures were stored in procedure manuals – and all too easily ignored. Sometimes those pictures would be drawn as part of an attempt to understand the process and how it might be improved, how things might be done more quickly or more cheaply: in other words for process improvement. Total Quality Management (TQM), with its Seven Tools of Quality, used simple models to examine processes, looking for those – typically small-scale – improvements.

In the second wave, processes initially became the unwitting victims of the information system developers: 'information' was on top. By cementing the allowable information flows in the information system, the information system engineers cemented the business process at the same time. The process could no longer be changed without expensive re-engineering of the underlying information systems. But worse was to come: processes found themselves captives of enterprise resource planning (ERP) systems. A side-effect of Business Process Re-engineering (BPR) was that many of the 'common' processes of organisations were re-engineered to what was (not jokingly) called 'best practice', or 'world class practice'. Recognising that the commonality of such processes meant that they could be made commodities, vendors of ERP systems defined processes for whole areas of organisational activity and built them into their systems – everyone could be offered the same 'competitive advantage'! The 'Inventory Management Process' could be bought by all and sundry. That's '*The Inventory Management Process*'. Once again, the process had become a victim, condemned to second-class citizenship, subservient to 'information'.

But we are now seeing a Third Wave – a term first coined by Howard Smith and Peter Fingar in their book, *Business Process Management: The Third Wave* (Meghan-Kifer Press, 2002) – a wave driven by the new technology of *business process management* (BPM). Finally, the business process has become a first-class citizen, the one that now determines what information will be kept. Process first, then information. Instead of being buried in the rules of a

INTRODUCTION

relational database, or the settings of an ERP, invisible but cast in concrete, the process is now visible, changeable and, potentially, back in the hands of the organisation, with the result that the organisation now has the power to change how it wants to do its business, and to change it when it needs to.

At the heart of BPM is a different understanding of business processes. Part of that understanding is that our process is not something that could perhaps be ‘deduced’ from the way our information system is set up or from what our ERP allows us to do. It is not ‘implied by’ the information system. Our process has its own separate existence in a form that – given to a ‘process enactment engine’ – can be executed or ‘run’, that can be changed on the fly, that can be evolved as our business evolves, that can be monitored in real-time, and that can be deployed at will through the organisation. A computer system that supports our organisation no longer simply helps us to manage our information: it now helps us, first and foremost, to manage our processes. It is a business process management system (BPMS).

This third wave needs appropriate methods for thinking about processes, for working with processes, for defining, designing and analysing processes in a way that positions us to use those new BPM systems. If you are familiar with my previous book *Business Processes* you will know that such a method has been around for some time, originally in the form of STRIM, and now in its updated form *Riva*. This book describes *Riva*.

Riva has its roots in the *IPSE 2.5* research project carried out as part of the UK Alvey Programme in 1986. As part of the project, Clive Roberts and I undertook to develop a language that, firstly, could be used to describe a process, and, secondly, was defined to a point where a process model written in that language could be given to a computer system which would then ‘enact’ the process, thereby supporting the group who would collectively carry it out. The solution we developed was a combination of Anatol Holt’s Role Activity Diagrams (RADs) to which we made some important adaptations, and Sol Greenspan’s *Requirements Modeling Language* (RML) from which we developed the *STRIM Process Modelling Language* (SPML) in particular by adding the concept of *role* to its formal semantics. Central to *IPSE 2.5* was the idea that a process model could be changed *on the fly* by the process users and while the process was running. (References to relevant literature can be found at the end of this chapter.) *IPSE 2.5* was an early BPMS, but it has taken the intervening decades to see the development of technologies necessary to make it a reality, not least the internet, web services, and much of the object-oriented application infrastructure now available.

When Clive and I developed SPML we had to provide formal semantics for the language so that it was executable and supported on-the-fly changes. We also defined transformation rules that allowed us to translate a RAD into SPML; this meant that a RAD had strong semantics itself, and it is this feature that is important in the area of modelling to enact a process. SPML has now been taken forward and the new version is called *CoSpeak* – it is a language of coordination. To the everyday user of *Riva*, the presence of an XML-variant called *CoSpeak* is not important. But to the analyst and to those interested in building BPMSs, the presence of a formal language with full semantics, underpinning RADs, is vital. Without those semantics we cannot say unambiguously what a given RAD means in terms of the behaviour of the process it describes.

Holt himself describes the application of his Role/Activity Theory to *coordination systems* in terms of a conceptual model based around what people as a group *do*, rather than

INTRODUCTION

around the data they operate on. It is hard to over-emphasise this point. A lot of process modelling has come from the software engineering world where, historically, data and information have ruled. That sort of process modelling has therefore concentrated on things and data about things. But processes are about dynamics, activity, collaboration, cooperation. So the way we think about process must have these at the centre. We must put processes back on top. Riva does this.

WHAT A 'PROCESS' IS AND WHAT IT ISN'T

I'm a stickler for good definitions. I believe that we can get a long way in solving many problems by defining our basic concepts properly. By striving for clarity in those basic concepts we force ourselves to deepen our understanding of our subject. Moreover, those definitions are the foundations of our work: poor definitions mean poor theories, and poor theories make poor practice. I don't want to assemble a lot of shoddy ideas and call them a method. I want to build a *principled* theory and practice of business processes. The more principled our theory is, the more powerful it will be, the more quickly we can get to results. I also want it to be as simple as possible, but not too simple, as someone cleverer than I once said.

This leaves me wanting to start this book by defining the concept 'process'. If we don't know – from the outset – what the word means, then we won't get very far.

Every organisation does things in order to achieve its objectives. For instance:

We handle orders for goods.

We recruit staff.

We design new products.

We run an investment portfolio.

We develop new pharmaceutical drugs.

Something that characterises all of these is they are 'quite large things'. They probably involve more than one person, take more than a moment, and might be carried out in different ways in different situations. We can appreciate that they are different from something like 'Fill in expenses form'. When we use the word 'process' we are thinking of a *coherent body of organisational activity*: activity that goes on in the organisation and that in some sense comes as a unit. Typically 'comes as a unit' means 'is all focussed on a certain outcome'. For instance, in the above cases the outcomes might be

To respond to a customer order by shipping the requested goods and invoicing the customer for payment.

To respond to the staffing needs of the organisation by engaging staff of the right type and capabilities on appropriate terms and conditions.

To answer a gap in the market place with a product that can be manufactured, marketed and sold profitably.

To decide how available funds will be allocated to financial instruments in order to realise gains of the right value at an acceptable level of risk.

INTRODUCTION

To develop and bring to market new pharmaceutical drugs that are efficacious and safe.

Let's look at some of the essential features of a process. A process involves *activity*: people and/or machines *do* things. A process also generally involves more than one person or machine working together: a process is about *groups*; in particular it is about *collaborative* activity. And a process has a *goal*: it is intended to achieve something. The group collaborate to achieve the goal.

We can also characterise the concept 'process' by what it is not. It is not the same as a 'functional group', e.g. Personnel, Manufacturing, Finance, Goods Inwards, or Credit Control. These are parts of the organisation which have responsibilities, staff and resources; but they are not processes, though they take part in processes. In fact, we shall see later that the relationship between processes and functional groups can be complex and indeed that the efficient operation of processes can be hindered by an organisation's structures. In a re-engineering context we shall want to explore that relationship between the organisation and the process. So, we shall resist all thoughts of the 'Finance process', or the 'HR process' – these are meaningless phrases.

This discussion begs a question: 'How do we sensibly chunk all of the organisational activity into things we can call processes?' This is a hard question, and one that is often badly – even wrongly – answered in business process management projects. Key to it, I believe, is that the method we use must give the answer appropriate to the business of the particular organisation we are looking at. And if *you* use that method and *I* use that same method, we should surely both end up with the same answer. If we reach different answers, what value can we give to either?

Suppose I walk into the dissecting room at the teaching hospital to lecture to medical students on how the human body is constructed. Awaiting my arrival is my assistant, ready with a thick marker pen. There, on the table, is the cadaver. I have brought with me an axe. With a deft overhead blow I lop off the lower part of one leg. My assistant labels it 'The A bit'. The students dutifully record the name against their sketches of the body. Some aren't sure quite where on the leg the axe fell, but choose a point anyway. With no more ado, another blow removes the top of the skull, which my assistant labels 'The B bit'. More scratching in notebooks. Further blows yield bits C, D, E, the remainder on the table being labelled 'The F bit' with the marker pen. 'The F bit' is still quite large, so four swipes render it into five pieces, which my assistant labels 'The F1 bit', 'The F2 bit', and so on. The students take notes, increasingly unsure about exactly how much corpse each bit is made up of. Never mind, I now take bit F2 and, putting down the axe in favour of a small meat cleaver, I cleave it into bits whose names, I tell the students, are F2a, F2b, and F2c.

What understanding do the students now have about the way the body is constructed and how it works? Has the chunking been guided by an understanding of what a human body is all about? Would each student have the same understanding of exactly what constituted each bit? If I gave the same lecture next week, would the student's drawings be anything like those of this week's?

You would, I know, prefer that I had taken a scalpel with me to the dissecting room, together with an understanding of what a human body is all about – the fact that there are 'natural cleavage lines' that separate the central nervous system, the gastro-intestinal system, the

INTRODUCTION

skeleton, the musculature – and of how those systems are connected. We look for these things because we know that a human body is ‘in the business of’ feeling and sensing, nourishing itself, standing and moving.

When we chunk organisational activity we shall need a similar scalpel that will allow us to cut along the natural cleavage lines of that activity, to separate out the processes using an understanding of what the organisation is all about, an understanding in particular of what business it is in. In the early chapters we shall look at how to model an individual process, without in the first instance worrying too much about how we know that *that* pile of activity constitutes a process, dissected along natural cleavage lines, rather than a chunk that has been as-good-as-randomly hacked out of the whole. In Chapter 6 we shall discover how to do that dissection, how to divide all the activity going on in the organisation into a set of processes that has been rigorously derived from an understanding of what business the organisation is in. And we shall do it in a way that yields the same answer whoever does the analysis.

KEY POINTS

A process is a coherent set of activities carried out by a collaborating group to achieve a goal.

The chunking of organisational activity into ‘processes’ must be driven by an understanding of the business the organisation is in.

Before we go further, I must say how we shall be using the words ‘business’ and ‘organisation’. We shall use the neutral word ‘organisation’ to mean any group we are interested in: a team, a department, a company, a group of companies, a company and its customers, a nation, whatever. I shall (try to) restrict my use of the word ‘business’ to mean ‘what the organisation gets up to’:

This organisation is in the business of making and marketing furniture.

This organisation is in the business of providing shared services to local hospitals.

This organisation is in the business of managing building programmes for the city.

This organisation is in the business of collecting fines imposed by the courts.

Deciding on the ‘organisation’ we are concerned with, can itself be an important decision. Gone are the days when we would only worry about the efficiency of individual activities. Today we are concerned with the efficiency of, say, our entire supply chain – from one end to the other. Gone are the days when we would only worry just about our patch, ignoring what happens the other side of the wall, in the world of our suppliers or our customer or our partners. Today we need to ensure the way we work with them is fully thought through and

INTRODUCTION

integrated. Gone are the days when we could get on with our business in private. Today we are increasingly required to make our end-to-end processes visible.

WHY WORRY ABOUT PROCESSES?

There have been a number of business 'movements' over the last twenty years that have made people recognise that they have processes and that these processes are what the organisation is about. The central notion in each is that of *process* and there is a need to be able to picture a process, through a *process model*. Like all models, a process model will capture just those things that we need for our purpose. To understand the needs of the process modeller we must look at the various situations we might find ourselves in where process is important. We can readily identify the following seven. They are not completely separate but it is useful to consider them separately for now. They are:

- ☞ Situations where there is a need for a *shared understanding* of what the organisation does and how it does it.
- ☞ Situations where a *common approach* to doing something is to be adopted and perhaps mandated, for instance through a Quality Management System (QMS).
- ☞ *Incremental improvement* programmes, such as might be run under the banner of Total Quality Management.
- ☞ *Radical change* programmes, such as might be carried out using the principles and techniques of Business Process Re-engineering.
- ☞ Situations where traditional data-based *information technology* (IT) systems need to be aligned with the needs of the organisation.
- ☞ Situations in which *workflow management systems* are to be used on a computer system to control the flow of work.
- ☞ Situations where new forms of *process technology* such as Business Process Management Systems are to be applied to give active support to the management and enactment of business process.

Let's examine each in a little more detail.

Understanding your organisation

It has to be said that not every organisation recognises that it operates processes, even though it knows perfectly well how it is structured into functional groups and what each of those functional groups is responsible for. Whilst people might appreciate in some abstract way that the organisation can only work through their collaboration, they might individually have very little idea of how the collaboration actually works – they each do their bit, but how do the bits fit together? When I have modelled a process within an organisation, people will often remark 'You know, I've never really thought of things in terms of a *process* that starts there and ends there.' People know about what they do, who they depend on, and who they pass things on to. But they might not be aware of the larger, end-to-end process in which they, along with many others, play a part.

Simply modelling the process can provide individuals and groups with a perspective on the organisation that transcends parochial views and, as a result, can promote a more

INTRODUCTION

collaborative spirit. 'Now I know why you want that, I can make sure you get it reliably.' We are interested in helping people to 'get out of the functional silos'.

A model that makes the process *visible* to the parties concerned can in itself bring great value.

Standardising processes in Quality Management Systems

The emergence and development of the ISO 9000 series of standards led to an increased concern with how an organisation goes about its business in a way that ensures quality in the products or services that it delivers to its customers. ISO 9001 (ISO 9001:2000 *Quality Management Systems – Requirements*) sets a standard for a Quality Management System. Central to the standard is the notion that key processes should be defined in some way so that they are repeatable, measurable, and improvable. The details are unimportant here, but the message is: if you are concerned with ensuring the quality of your product or service, you must concern yourself with the processes that deliver that product or service.

Typically, therefore, an organisation will describe how its processes are carried out in a way that

- ☞ communicates the processes to those who must carry them out ('How should I do this piece of work?');
- ☞ provides the opportunity for independent assessment of the organisation's conformance to the process it has laid down for itself ('Are these people doing what they said they would do?');
- ☞ acts as a basis for future improvement of the process ('We do this now, but how could we do it better?').

Such descriptions tend to be lodged in some form of Quality Manual and they can take many forms, typically a mixture of text and diagram. A good description – model – of a process will be one that *communicates* in sufficient detail to those that must carry it out, that is *precise* enough to permit an assessment of conformance, and that is appropriately *detailed* to be a basis for analysis and improvement. (Note the careful use of the words 'sufficient', 'enough' and 'appropriately'.)

Incremental improvement

It has long been recognised in the disciplines of quality management and TQM that the cost-effectiveness and profitability of a process are determined by the quality of the goods or services it produces, and that that quality is itself determined by the process as well as the inputs and the workers. In particular, if we want to reduce wastage (of materials, or resources, or time) we need to address the 'common causes' (to use the jargon) of defects, and this means removing systemic errors, i.e. those introduced by the process itself.

To do this, we need a way of exploring at an appropriately detailed level just what happens between the customer making a request and the customer going away satisfied with the goods or services we have provided, and, within that flow, we need to understand where defects and/or unnecessary delays are introduced so that we can adjust the process to remove their cause. Central to that exploration is a model of the process, through which we can ask where improvements can be made.

INTRODUCTION

The aim is that, bit by bit, we refine the process and gradually eradicate those systemic causes of poor quality: we are in the arena of incremental improvement.

Radical change

In the world of Business Process Re-engineering, incremental improvement is not enough. Here we are looking for major breakthroughs, an 80% reduction in cycle time not 10%, reducing staff levels to one-fifth not by one-fifth. And to do this we are prepared to make radical changes, not just tinkering with the fine detail of our processes but making major changes in our organisational structure, and questioning the very need for doing things the way we have done them for years, or even why we have those processes at all. We are prepared to ask questions like 'Can we operate without a central Purchasing Department?', or 'Is tendering the only way we can ensure the best price for bought-in goods and services?', or 'What would happen if suppliers were paid by the recipient of the goods rather than Accounts?'

In this context, detailed maps of our current processes are largely irrelevant. We've decided that we shall only consider big changes – detail is simply not interesting. But an architectural view of our processes, and broad-brush models of the way our organisation operates, of what processes we have and how they traverse the functional silos, could give us clues about the sorts of radical change we might imagine. We could question whether we could remove entire processes by thinking about how we do things in a more radical way.

And when we have decided how we want our new organisation to land when we have thrown the existing one in the air, we shall need some way of designing the new processes, ensuring that they fit with those that survive and with each other, and that they make sense in our new flattened or process-oriented organisational structure.

Process design means, again, being able to model the process, this time the *new* process.

Building on database management systems

I once worked with an IT department that had built a system designed to support the information needs of a particular group within the organisation. The system was designed to provide that group with a way of recording and tracking progress on the items they were processing in real-time. But, at the time that the system was designed, no-one had recognised that the work of that group was inextricably linked with the work of another group who had their own IT system; the new system quite simply 'clashed' with the process by which each work item had to be handled. As a result the users resorted to inputting information *once the process had finished*, and as a result the system failed to provide the real-time support which it had been intended to provide.

Rightly or wrongly, past IT systems have often been considered to have been failures in that they have not brought the benefits to the organisation that were promised. There has been an assumption in the minds of those that build IT systems (and I have been amongst them) that data and information are central: that if we start with an analysis of the information needs of the individual in the business process we shall build a system that supports the organisation effectively. Unfortunately, this ignores one important feature of organisations: they do not work simply by ensuring individuals have information at their finger-tips; they work by having processes in which groups collaborate effectively. Good IT follows firstly from an understanding of the way that the organisation does its business with the structures it has, and only

INTRODUCTION

then from an understanding of the information that the organisation needs because of the way it chooses to do its business.

Process precedes information. Once we have decided how we shall carry out our business – our process – we can identify the information needs and hence the information-based systems needed to support them.

Building on workflow management systems

In the 1990s, a new class of software infrastructure products emerged: *workflow management systems*. These provide active support to a simple business process by controlling the flow of a work item around the organisation, routing it and its supporting information and (electronic) documents and images from person to person in the process, from workstation to workstation.

Such systems clearly need some model of the process, a model that describes the path the work item takes from role to role, the decisions, the alternative paths, exception handling, escalation paths, and so on. Once again, we see the need to be able to model a process in terms of roles, activities, decisions, and flows from role to role (what we shall generally refer to as *interactions*).

A key feature of such infrastructure products was that the process was pretty much set in stone: once programmed into the workflow system, changing it became a major undertaking, on a par with changing the structure of a database supporting a process: not something to be undertaken lightly, and something only to be done by the experts in the IT Department. Alas, workflow management systems have also tended to use proprietary models (i.e. they are all based on different process constructs), and are not general enough to express all the sorts of process we might wish to run.

Building on business process management systems

More recently, we are seeing an entirely radical class of products: *business process management systems*, or BPMSs.

A BPMS takes a description (model) of a set of processes, and *enacts* it: we might say that it ‘executes’ it, or ‘carries it out’, or ‘runs it’. In the same way that a normal computer ‘carries out’ a software program, so a BPMS ‘carries out’ a set of processes. This much they have in common with workflow systems. But in a true BPMS, a radical step is taken: in the parlance, processes become ‘first-class objects’; so, within a BPMS, a process can be itself managed, revised and passed around. In this new paradigm, processes come out of the shadows and become true and visible business assets.

We can express this succinctly by saying that a BPMS supports us working *in* a business process, and supports us working *with* a business process. In the first case we enact the process, in the second we manage the process. The moment we recognise that the management of a process is itself a process, we realise that the BPMS is a new sort of world where the unit of currency is the process not its data, a world in which that currency is both minted and used – processes are defined and enacted. To use more of the parlance, we say that processes have *mobility*: rather than being something static – statically defined and statically followed as in traditional workflow systems – we allow them to be the subject matter of other processes, we allow processes to be passed around for enactment, and we allow a process to grow as a network of interacting and collaborating parts.

INTRODUCTION

The modelling situation here is an order of magnitude more complex – the problem is the same as that addressed by the IPSE 2.5 project that I mentioned earlier and that contains the seeds of *Riva*. As a result, *Riva* has the necessary concepts and machinery for those modelling processes using languages such as BPML and BPEL.

THE RIVA METHOD

Whatever our reason for taking an interest in our processes, we shall need a way to define, record, discuss and analyse them, and we shall need a language for talking about processes. This is where *Riva* comes in.

Riva is a method for the elicitation, modelling, analysis and design of organisational processes.

It uses two languages for talking about processes:

- ☞ The *Process Architecture Diagram* (PAD) is used to describe the overall chunking of the organisational activity into individual processes.
- ☞ The *Role Activity Diagram* (RAD) is used to describe an individual process. In a modelling project, we shall always have one PAD and one or more RADs.

The *Riva* method includes techniques for

- ☞ determining what processes an organisation must have in order to be in the business it is in (chunking);
- ☞ 'discovering' and modelling an existing process;
- ☞ defining an existing process;
- ☞ designing an intended process;
- ☞ qualitatively analysing a process once a model has been produced;
- ☞ using process models for requirements definition for information systems and workflow systems;
- ☞ developing process models for BPMS development.

In any process modelling *method* we shall want to find more than just a notation, more than a way of drawing pictures: we shall want to find 'intellectual machinery' that helps us to think about our processes and get to answers. *Riva* is a rigorous and rather uncompromising method: I take the view that a sloppy model cannot be relied on; a sloppy model can give the wrong answers; a sloppy model does not give real insight; a sloppy model cannot be used as the basis for any sort of computer system that is intended to support the organisation, whether it be data-, workflow- or process-based.

But before we look at the notations and the techniques themselves, we must pause and examine our motivations and needs in a little more detail.

Chunking

When we set to work on an organisation's processes, our first problem is getting our arms around the whole thing. Walking into the building where the business is done, we find a mass of activity going on. How on earth do we start? And where do we start? Presumably all this

INTRODUCTION

activity falls into some sort of chunks, chunks that make some sort of business sense. But how do we chunk it in a way that makes business sense?

Let's go a step further. If our organisation changes its structure, surely we wouldn't expect to end up with a different set of chunks: it is still in the same business after all. How it does those processes might change, but it still must have those processes. And surely to be in a particular business we need a particular set of processes. And if our organisation changes its culture, surely it will still have the same processes. How the processes get done will – of course – change if we change the culture. But the existence of the process will not – it remains there, in one shape or another, as long as we are in the same business.

We shall be looking for a method of chunking that gives us a 'process architecture' with this property of invariance. What we are looking for is a chunking that is derived solely from an understanding of what business the organisation is in. We would like to say 'If the organisation is in this business then it must have these processes.'

When pharmaceutical drug compounds are being developed, small batches are made for clinical trials using general purpose pilot plants. Such plants are expensive to build and run. There is a queue of batches waiting their turn for use of a pilot plant. If that queue is not managed properly, the effect on the drug pipeline can be significant: it doesn't matter how quickly each batch goes through if the important batches are held up because of bad planning. Turn to the Procedures Manual of any organisation, and I can almost – almost – guarantee that it will only cover 'coal-face' processes, processes that deal with a manufacturing batch, or a purchase order, a sale, a customer, a clinical trial. But what about that flow management? When individual cases compete for resources, management activity is necessary to decide priorities, allocate and schedule resources. We can draw two lessons. Firstly, when we chunk all the organisational activity, we must not forget that that management activity is out there and deserves equal recognition with the coal-face processes. Secondly, we cannot mix the description of how that management activity is done with the description of how an individual case is done – they are different processes.

So, of all that activity in the building, we must be sure to cover both 'coal-face' processes and management processes. But what about all those people in the conference room, standing around the whiteboard discussing how they should respond to recent developments in their competitors' offerings; the finance folks sitting with their bankers looking at their exposure to currency fluctuations and deciding whether they have the right mechanisms in place to deal with them; the Board discussing possible take-over targets with their management consultants. None of these people are carrying out the day-to-day business processes, or the processes that manage those processes. They are standing back and taking a longer view of the business. They are involved in what we might categorise as 'strategy work': the results of their deliberations will change what happens and perhaps how it happens. When we put our arms round the organisational activity and chunk it up, that strategic work must also appear if we are to have a complete picture: we shall expect to find strategy processes.

In summary, we can expect our chunking to expose processes in three flavours: coal-face, management, and strategic. We shall examine these more thoroughly in Chapter 5, before describing in Chapter 6 how to use them when developing a process architecture.

INTRODUCTION

Describing, designing, analysing, enacting

A recurrent theme in this book is the idea that what is included in a model for an individual process and what is not will be determined largely by the perspectives we choose to take, which in turn are determined by the reasons we have for modelling in the first place. We have already seen that there are a number of reasons for taking a process-oriented view of our organisation, and that we have many reasons for modelling the processes within it. 4Ds and an E: discovery, definition, diagnosis, design, and enactment.

Modelling to discover and define a process

This is what we do when we want, amongst other things,

- ☞ to discover a process that has not seen the light of day: ‘This is how we apparently handle customer complaints’;
- ☞ to define a process: ‘This is how we will all handle customer complaints’;
- ☞ to communicate it to others: ‘This is how your work contributes to customer complaint handling’;
- ☞ to share it across a group of people: ‘So, this is how together we handle customer complaints round here’;
- ☞ to negotiate around it: ‘If you could do this, my life would be made much easier; in return I can ...’

It is surprisingly common for an organisation not to have a clear idea – or sometimes any idea – of how certain things are done. They do get done, because people fit in, work things out, develop their own patterns of behaviour, and pass them on. There is a sense in which this could be regarded as ‘good’: it promotes flexibility, discourages slavish subservience to a set of rules, allows things to develop and shift as the environment changes, and so on. On the other hand, it becomes hard to promote good practice, makes life hard for new recruits, doesn’t permit steady improvement, allows reversion to bad practice without anyone noticing, and so on. Process modelling often has a role to play in revealing to the organisation how things are, perhaps how bad things are.

We might call this ‘process discovery’. It is about building a shared understanding of the process as it is today, an understanding that we can communicate to others.

Such discovery might be a first step to agreeing on a common way of doing things. Suppose a group is seeking ISO 9001 certification for its Quality Management System; it will want to define its processes in its Quality Manual. We shall probably find a descriptive process model in some form in such a Manual. The model acts as a work instruction to people in the organisation. Text is very often used to describe how things are done, but the serial nature of text makes it impossible to adequately describe – let alone prescribe – something that has possibly many threads, decisions, concurrent activities and so on. A diagram is a traditional way of dealing with this.

A software house had a number of its processes described in its Quality Manual in the form of Role Activity Diagrams. Those diagrams told people what was expected of them when carrying out the processes; in particular they specified in quite considerable detail the key business processes of planning and reporting projects, of bidding for new contracts, and

INTRODUCTION

purchasing – all processes that had an important and direct financial impact on the company and that it therefore wanted to ensure some degree of conformance to. Other processes were about delivering product quality. RADs allowed them to define all these processes to an appropriate level of detail.

Summarising our needs for process description is difficult: we can have so many different reasons to describe processes. Certainly our model must say what we want it to say. So the notation must allow us to say those sorts of things, and our method must help us do that. We shall look more closely at the practicalities of modelling for discovery and definition in Chapter 9.

Modelling to design a process

It is relatively rare for people to have cause to design a new process or set of processes from scratch. It probably means that either a new organisation has been created or an existing one has undergone very radical change of business.

One group of managers in a multinational product company wanted to define a planning process that they would use, involving their management and other corporate functions such as 'HQ', 'Finance' and 'Audit'. They already had a process in place but it was not well articulated and they were fairly certain that they did not all have the same view of the process. The process was that of deciding on the portfolio of products in which they would fund investment, and the process of developing a new product within that overall planning process. They already had a short verbal description of the proposed new process and we used the methods described in this book to capture those processes completely. In addition to giving them greater understanding through the analysis, the method gave them a clear diagram which then went into their Procedures Manual. It also filled in many gaps in the textual description which only became apparent when the process was explored using Riva's rigorous notations and method.

With the fragmentation of industries such as utilities, and the consequent construction of new regulatory frameworks and bodies, entirely new processes become necessary to ensure fair competition between the fragments. These can be truly green-field sites for process design. Nothing similar exists in the current organisation; entirely new objectives arise, and new roles and responsibilities must be created, and so on. A new utility organisation found itself having to set up a new group with responsibility for reporting to the regulatory body. Nothing like this had been done before: what processes would they need? Never having had the need before, there was no organisational structure to support it. How should they design those processes in the absence of the 'comfort' of an existing organisation?

We shall look more closely at the practicalities of modelling for process design in Chapter 11.

Modelling to diagnose a process for improvement

Once we have a model of an existing process, we may well want to use the model to analyse the process itself. Some of the questions we shall want to ask will be quantitative: 'What is the average cycle time?', 'How much will the cycle time be affected by changing the process in this way?', 'Where are the bottlenecks?' Other questions will be qualitative: 'Do we have the optimum division of tasks across the people involved?', 'Why does this paperwork flow back and

INTRODUCTION

forth?', 'Are the right decisions being made at the right level in the organisation?', 'Are we over-doing the financial oversight?'

Such analysis is a common precursor to improving the organisation by, for instance:

- ☞ changing the order of activities,
- ☞ changing responsibilities for activities or decisions,
- ☞ changing the way things are scheduled,
- ☞ increasing or decreasing the amount of parallel activity,
- ☞ removing or inserting buffers or stores for materials between steps in a process,
- ☞ restructuring functional groups to align them better with the process.

Any organisation involved in TQM or other process improvement activity will want to model its processes and analyse them for weaknesses or inefficiencies on paper, before trying out improvements in real-life.

We shall look more closely at the practicalities of modelling for process improvement in Chapter 10.

Modelling for traditional information systems

Traditionally, when considering computer support for the activity of an organisation, we have done some sort of analysis of the organisation in order to identify where automation can best be applied. Since the mid-1970s that analysis has concentrated on data, and into the twenty-first century data analysis continues to be the cornerstone of software development: object analysis, data-flow analysis, entity relationship modelling, and entity life history modelling in particular have featured large and continue to do so. The move towards object-oriented conceptualisation has changed that scene little.

Unfortunately, too little attention has traditionally been paid to understanding the business *process* that is being supported. Even an otherwise well-thought-through system development method such as SSADM (see for instance *Business Systems Development with SSADM*, Office of Government Commerce, UK, 2001) has taken the view that business activity is relatively unstructured, or at least that any structuring is not of great interest. With the advent of the Unified Modeling Language (UML), we have seen an increased interest in the nature of organisational activity, with the idea of so-called *use cases*.

We can look to our process modelling method for two things in the IS context. Firstly, we can expect it to reveal more effectively, reliably and efficiently the use cases of the business. Secondly, we can expect it to make the information needs of the individual clearer in the context of the entire process. We shall look more closely at the practicalities of modelling as a precursor to defining requirements for traditional information systems in Chapter 12.

Modelling for workflow management systems

Traditional information systems have for the most part had a very simple architecture. In essence, they provide the individual with a peek'n'poke facility into some central database – most importantly *they only give explicit support to the individual*. Every individual is plumbed into the data. But organisational processes are not just collections of individuals operating independently. Organisational processes are carried out by groups of people acting collaboratively to achieve a goal. If our computer systems are to reflect this they must support groups

INTRODUCTION

more directly. In the last two decades, software packages have appeared under the title *work-flow management* (WFM). Their architecture recognises that people who connect to the system are working jointly on a case, and so that architecture must support the flow of a case from person to person. They have recognised that organisational activity is more than a set of individuals manipulating data, and is better seen as a flow of work items between collaborating individuals.

Now, traditional data-oriented systems analysis and design have served us perfectly well when we build data-oriented systems supporting individuals. But WFM products require different analysis and design methods. Data-oriented analysis methods provided us with various sorts of data model. WFM products need process-oriented methods providing us with process models.

We shall be looking for ways of identifying workflows, and then understanding the nature and content of that flow, in particular who gets to do what when, what information they need to do it, what information flows with the workflow, and what is static. We shall cover this in Chapter 12.

Modelling for business process management systems

Given a data model for a data processing system, any database management system will allow us to store this in a database and use it directly to generate forms and reports that the individual can use to add, amend or present data. There is a thin sense in which we could say 'The data model is executed.'

What's the analogue for process models? Suppose we could give our process model to a computer system and have it *enact* that model, i.e. 'run' the model, supporting the participants in the process as the process proceeds, handling their agendas, supporting their interactions, and perhaps playing its own part in the process. Systems that provide this sort of support are termed *enactment systems* and they provide us with our final motive for process modelling: they require a process model whose 'meaning' is sufficiently well defined to allow them to enact the process without further human intervention.

Such an enactment system will go beyond handling simple workflows, and will deal with the network of processes that can be in progress at any one moment. Processes themselves will become a sort of currency in the system: they can be changed and passed around. This use of process models will have important implications for the process modelling method and its notations. But we can imagine the step beyond that we touched on at the start of this chapter: the idea that our processes, 'living' inside the enactment system, can themselves be manipulated in a variety of ways: small changes, big changes, new processes even. As well as supporting our use of our processes, such a system would support our *management* of our processes. It becomes a *Business Process Management System*. Not surprisingly this introduces a whole new raft of requirements on our modelling notations and method.

We shall look more closely at the practicalities of modelling for BPMSs in Chapter 13.

KEY POINTS

Our modelling method must enable us to

- ☞ take an architectural view of our processes;

INTRODUCTION

- ☞ expose our processes as a discovery activity;
- ☞ define our processes appropriately in a regulated environment;
- ☞ use our models to diagnose our processes;
- ☞ design new processes from scratch;
- ☞ expose the information needs of the participants in a business process as a prelude to building an information system;
- ☞ design workflows that can be supported by a workflow management system;
- ☞ produce the process definitions that can be enacted in a BPMS.

EIGHT PRINCIPLES FOR PROCESS MODELLING

To finish this chapter, I'd like to introduce a Tutor and a Pupil in discussion about some of the requirements of a process method. Their dialogue – the first of a number throughout the book – highlights some important principles that underlie *Riva*.

Principle 1: If we must have abstractions, let's make them meaningful

Tutor: When we model something we describe it, and to describe something we need a language. A model uses a limited language – a limited number of concepts – that allows us to say the things we want to say, to describe the things we want to describe. If we wanted to model a nation's economy how would we model it?

Pupil: we might capture it in terms of the flow of money between the various places it can reside: the Treasury, people's savings, money in circulation, investment instruments of various sorts, and so on?

Tutor: Yes. Note how we'd be working with two abstract concepts: a 'pool', of which each of those residences is an example; and a 'flow' – the movement of money between two pools. By varying the rates of flow we can investigate the behaviour of the model and perhaps deduce something about the behaviour of the real economy. An early economic model used precisely these abstractions, and made them concrete by using plastic containers of water to represent pools, and piping and pumps between containers to represent flows. The pools and flows were represented as plumbing, and money by water.

When we model processes – what people do – we shall need a small number of concepts that represent real-world things, but they must be concrete enough for someone looking at our model to readily understand the model and what it is telling them.

Principle 2: The real world is messy

Tutor: At a suitably high level of abstraction, any process can be made to look neat and tidy. Figure 0-1 summarises the process of collecting taxes from you and me and giving them to the government to spend on our behalf. Tidy isn't it?

INTRODUCTION

Figure 0-1 – Tax collection: pure simplicity



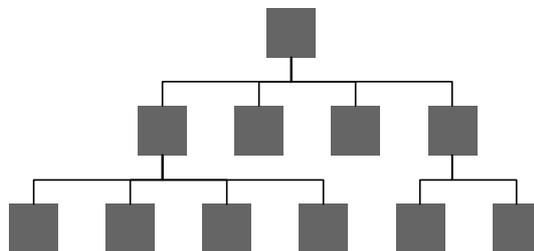
Pupil: Well, it certainly summarises what tax collection is about: namely that there is some relationship between The People and The Government that involves The Tax Collector.

Tutor: So how is it that tax collection causes so much grief and there is so much money to be made out of helping people with their tax affairs? The fact is of course that the tax collection process is complex and involved. A summary diagram like the above hides all that – and can make it next to useless if we want to answer any questions.

Pupil: Could we ‘decompose’ this neat picture in some way? We could still keep it neat by ensuring that we restrict the number of boxes on the diagram to, say, no more than seven. After all, someone did once say that people have trouble with handling more than seven items of information at any one time. And, if we want to, we could allow ourselves to ‘decompose’ any one of those boxes in turn onto its own diagram with another seven-ish boxes to describe it; and so on.

Tutor: You seem to be saying that our process model can take the form of a hierarchy, consisting of a set of diagrams, each of which (with the exception of the topmost one) expands a single box on its ‘parent’ diagram. Something like Figure 0-2 perhaps? A neat hierarchical model? OK, then answer me this: if this is a ‘good’ model, then it must be ‘like’ the real world it claims to model. Is it? In particular, does the structure of the process in the model reflect the structure of the process in the real world? When we look at the world, do we find neatly ordered processes, with each activity in a process neatly decomposable?

Figure 0-2 – Are the world’s processes really this neat and tidy?



INTRODUCTION

Pupil: I guess not. I suppose if we are looking at something statically – a city for instance – we could draw a model of its static nature and say that the city is made up of quarters, and each quarter has districts, and each district has streets and so on ... But we wouldn't be saying anything about the way that the city works.

Tutor: Exactly, and your point about the difference between a static and a dynamic view is very important. When we model organisational activity we are interested in dynamics. Dynamics are not hierarchical. Organic systems are networks, often networks of networks. And they are changing networks: things are born, establish relationships with other things, and then die. The network is a dynamic thing, constantly changing. There is a dynamic flux of activity ... not some static hierarchy of activity.

Pupil: Is there another sense in which models can be complex: things can simply get ... messy? In fact I guess we should be very surprised if something like the process of an organisation that has grown organically over time were so neat and tidy.

Tutor: True. We'll see that – when they happen – real-world processes can be complex, gangling, even muddled or messy (these aren't my words – they're words frequently used by the people whose processes I work with). They wander here, there and everywhere; the hierarchical picture fails to show that: it replaces convolution by neat (but irrelevant) hierarchy. The message for our process modelling approach is that we can't let the notation dictate to us that our models must be neat hierarchies. Things are much more complex.

Unfortunately the IT world has for years been using hierarchies as a way of designing synthetic things like computer programs, and software structures of all kinds. The great mistake has been to assume that we can use such hierarchy-based notions to describe *organic* things like businesses. Businesses aren't the product of tidy design activities over which some designer has had full control. They're the product of time and many uncoordinated changes by many hands. Traditional structured approaches from the IT world just won't work.

Principle 3: A model must mean something, but only one something

Tutor: What does figure 0-1 mean to you? What do the two boxes mean? What does the arrow mean?

Pupil: Errr ... from their names, the boxes seem to be collections of people .. or one is a collection of people and the other is an organisation ...

Tutor: I haven't told you what a box *means*, of course. 'Money' doesn't appear anywhere in the picture – are you surprised?

Pupil: Well, I assumed the arrow represented money.

Tutor: It could. But it could also represent 'paperwork'. Suppose we accept that the arrow represents money flowing from The People to The Government. Can we

INTRODUCTION

assume from the single head on the arrow that the money only flows one way? In fact it doesn't. I recently got a tax refund. Would having a two-headed arrow be more accurate? Let us assume that, because the arrow has the label 'The Tax Collector', the Tax Collector is the agent for the flow, the thing that makes it happen on behalf of The Government. Would that mean then that the Tax Collector sometimes makes the flow go from The Government to The People? Does the Tax Collector act as your agent for giving money to you?

Pupil: It was such a simple diagram – two boxes and an arrow – but it does seem to raise more questions than answers.

Tutor: Right. And the reason for that is that I have not given a clear meaning to the symbols I've used in the diagram. 'A picture says a thousand words' – a different thousand words to each of us, unless we agree on the semantics of our notation. If I can't tell what my process model means, I can't tell what needs doing. When I choose a notation for my models, I need to be sure that I have clear semantics for them: I know what a box means, I know what a line means, I know what each little symbol means. The model is unambiguous.

Principle 4: Process models are about people, and for people

Pupil: I can see a danger here. When we model a process we are describing what people do. (I suppose machines of various sorts might play a part in a process, but they don't have to be asked their opinion so I guess we can concentrate on people.) To find out what people do we'll need to ask them and watch them, we'll do some sort of 'process elicitation'. I'm sure we'll then want to put our model of their process in front of them and have them discuss it, correct it and improve it and ask whatever questions of it they want answered.

So the concepts we use in our model must be concepts people relate to in daily life. The notation has to make sense to people. If we can't explain it in ten minutes, it doesn't make sense. But on the other side of the coin, any well-founded notation will have subtleties and potential complexities, and so there's a danger of opacity. A newcomer to a model must be able to understand quickly what the model is saying. If the model needs extensive interpretation by a skilled analyst before it can be understood by the person on the shop-floor, the battle is half-lost, surely.

Tutor: Yes, there is a dilemma there. We want a notation that is powerful – that allows us to say all the complex things we want to say – yet is accessible. The acid test is this: at their first modelling session, can an ordinary person go to the whiteboard and correct a mistake in the model so that it correctly describes what happens?

Principle 5: There's what people actually do, and there's what they effectively do

Pupil: I have another problem. When the Accounts Department raises an invoice against a customer's order, what we see them doing is involving themselves

INTRODUCTION

and others in a paper-chase, in which the flow of paper is embellished with activities of transcription, checking, updating, copying, chasing, phoning, and so on. But what they're *effectively* doing is extracting payment for the goods and services supplied. How do we reconcile these two views of a process?

Tutor: The answer is that we won't try to reconcile them: we'll recognise that these are two different views of the same thing. When we look at a process in concrete terms, we see what people actually do, and such a model might be in terms of the *mechanisms* they use. But, when we want to understand how the process might be re-engineered, or what the possibilities for technology support are, or how the organisation and roles and responsibilities might be changed to improve things, we shall need to get to the bottom of the process, to understand what it is *effectively* about. We would draw a different model that expressed the *intent* of their actions.

Our modelling method and notation must allow us to prepare both sorts of model, what we might call *concrete* and *abstract* models.

Principle 6: People work in functions, but they do processes

Tutor: If you ask someone what they do in an organisation what sort of answer will you get?

Pupil: 'I work in Accounts,' or 'I work in Product Design,' or something like that.

Tutor: And if you press for detail?

Pupil: They'll tell you how they contribute to the work of the department, where the work comes from, and what happens as a result of their work.

Tutor: Right. The lessons from TQM and BPR have been that what people really do is play a part in one or more *processes*, and that these processes invariably cut through the boundaries between departments. Getting the organisation to recognise the existence of these 'cross-functional' processes and to deal with the conflict that occurs at the boundaries between the 'functional silos' is a major part of improvement and re-engineering.

There's a big message here for our modelling approach: do I have ways of modelling the process from the point of view of organisational structures and how work is allocated across the functions, as well as from a pure responsibility point of view? Can we separate out responsibilities from organisational functions? We have to be able to do this if we are considering re-engineering through change in the functional structure or in the responsibilities allocated to functions. Remember that we can have a functional group called, say, Administration, and then have to decide what responsibilities we want to give them.

Principle 7: It's what people do, not what they do it to, that counts

Tutor: What do you do?

INTRODUCTION

Pupil: I ... carry out activities: I write an article; I facilitate a workshop for a client; I elicit a client's process; I give a training course.

Tutor: Just one thing at a time?

Pupil: Well ... I often have a number of activities going on in parallel: I start a new piece of work for a client; I put one activity down to do another; I stop everything to fill in my timesheet; I resume one of the activities I put down; I finish an activity.

Tutor: Right, so you have a number of balls in the air, a number of plates spinning, a number of activities you could be doing, or are in the middle of doing at any one moment. Anything else?

Pupil: I make decisions: I decide that the text for the brochure need not go for another review but can go straight to the graphic artist; I decide that a request for new computer hardware is acceptable; I decide that an invoice is correct and can be sent to the client.

Tutor: Yes, you're choosing among alternative courses of action depending on the current circumstances. More?

Pupil: I interact with other people: the graphic artist and I agree on the final layout of the brochure; I send the manuscript of my book to the publisher; I delegate responsibility for hardware procurement to my facilities manager.

Tutor: OK. Let's pull this together. In all these things that occupy your day, you're doing things, sometimes on your own, sometimes with others. You're playing a part in a number of different processes at any one time, and in each you may have a number of actions in progress. It's reasonable then to expect that any notation we use to model those processes should allow us to capture activity: concurrent actions, collaborative interactions, and decision making.

We must focus on what people do, collaboratively and individually. What they do it to is of secondary importance.

I raise this because, here again there has been an unfortunate influence from the IT world on our process modelling (and I speak as a software engineer of many years). Hitherto the IT world has concentrated almost exclusively on *data*. On what people do things to. This is not surprising. Computers gave up computing pretty soon after their invention to concentrate on looking after people's data: memory became a cheap, voluminous commodity and that memory was as good as permanent. We've exploited that. When IT people build information systems they principally design databases and ways of getting the data in and out. The traditional development methods (SSADM, IE, Yourdon, and so on) rightly concentrate on the data aspects of the business being supported, and the notations are about data: Data Flow Diagrams, Entity Models, Entity Life Histories etc. But we should not expect their hammers to be good for driving home our screws. We are in the process business not the data business.

INTRODUCTION

KEY POINTS

If we must have abstractions, let's make them meaningful. Any process modelling notation must deal in business-oriented concepts that people relate to. Otherwise how can they tell if a model's right?

The real world is messy. The notation must be able to model mess when necessary. Muddle modelling is perhaps the norm, not the exception.

A model must mean something and only one thing. If our model is ambiguous, how can we tell what it is telling us or others?

Process models are about people, and for people. The notation must make sense to people. If we can't explain a model in ten minutes, it doesn't make sense.

There's what people actually do. And there's what they effectively do. These are different and we must be able to model both.

People do processes, but they work in functions. These two can be in conflict. A model must capture both – and the conflict.

It's what people do, not what they do it to, that counts. A process is principally about doing, deciding, and cooperating, not data or things.

THE STRUCTURE OF THIS BOOK

The book falls naturally into two parts: you can think of part 1 as theory and part 2 as practice.

When we want to think about a particular subject in life we need an appropriate vocabulary – words to describe the subject – and a grammar – ways of arranging words to convey meaning. If we choose the right words and the right grammar we shall be very expressive. Major branches of mathematics were held up until a notation was developed that not only represented the subject being worked on, but also worked as a notation. The same will be true for our subject: processes. If we choose the wrong vocabulary or the wrong syntax, we shall not be able to say things we want to say, and we might even end up saying things that are just plain wrong; we shall be describing processes that don't exist; or we won't be able to answer the questions we want answered.

So part 1 is very much about getting the right vocabulary and the right syntax so we can describe business processes in a way that meets our needs, whatever they may be.

- ☞ Chapter 1 – *Basic process concepts* – gets us thinking about just *how* we look at a process. What are the features of real-world processes that we want to reflect in process models?
- ☞ Chapter 2 – *Modelling a process* – provides all the 'vocabulary' necessary to represent a single process in a *Role Activity Diagram* (RAD). We shall examine the notation and the underlying concepts in detail.

INTRODUCTION

- ☞ Chapter 3 – *Dynamism in the process* – highlights the levels of within-process concurrency that can be captured in a RAD, and shows how we can exploit that richness when we model real-world situations, current or planned.
- ☞ Chapter 4 – *Process relationships* – examines the types of dynamic relationship that processes can have and illustrates how we represent them on RADs. The relationship types will be central to the construction of our process architecture.
- ☞ Chapter 5 – *The three basic process types* – describes the three main types of process – the *case process*, *case management process*, and *case strategy process* – that underlie the construction of a process architecture.
- ☞ Chapter 6 – *Preparing a process architecture* – deals with the construction of the *process architecture* of an organisation, a concept of central importance for reengineering, for overall process design, and for steering any process work.
- ☞ Chapter 7 – *Dynamism in the world* – shows how the process architecture captures all the between-process concurrency in the world.

Part 2 puts part 1's theory into practice.

- ☞ Chapter 8 – *Managing the modelling* – provides guidance on running a process workshop and conducting interviews in order to prepare a model of a process, for whatever reason. We shall concentrate on how to make appropriate modelling decisions, the need for fitness for purpose in process models, and how to get results quickly. Subsequent chapters customise this general approach for different purposes.
- ☞ Chapter 9 – *Discovering & defining processes* – covers the practical use of the approach in determining what processes an organisation has, in eliciting those processes onto RADs, and in the use of RADs in Quality Management Systems, tying into ISO 9001 with its emphasis on process.
- ☞ Chapter 10 – *Analysing for process improvement* – is about using the approach at both the architectural and the process level for asking questions about processes and their performance, and for driving tactical process improvement.
- ☞ Chapter 11 – *Designing a process* – covers the design of a new process architecture and new processes. As the processes do not exist today, we shall start from a blank sheet of paper.
- ☞ Chapter 12 – *Processes and information systems* – covers the use of the approach in constructing an IS strategy for an organisation, and in the design of information systems.
- ☞ Chapter 13 – *Processes and process systems* – covers the use of the approach in using the new wave of *Business Process Management Systems* (BPMSs) in which agile and mobile processes replace static data structures.

WARNINGS

Some warnings are in order.

INTRODUCTION

Riva is not just diagrams. We shall be drawing diagrams – three sorts in fact – but they are only half the story. The other half is the underlying concepts and how they can be used to get a real understanding of complex human organisational activity. Without them, the diagrams end up just being sequential flowcharts and the point is missed. If you feel tempted to simply check out the different sorts of blobs on a Role Activity Diagram in Chapter 2 and start drawing you will be missing a great deal.

When you come to the section on the Role Activity Diagram notation, you will find yourself reading (what I hope are) very precise definitions of things, with apparently simple concepts being teased apart mercilessly. You will ask yourself why all this is necessary. If you want rough and ready models, you will draw rough and ready models. You will misuse the *Riva* notation, ignore the subtleties, add new blobs and arrows of your own. In my defence, I must tell you that one reason for drawing a model of a process may be that *we want to execute the model*: we want to give the model to a computer and have it run the process for us, with human beings playing the roles and carrying out the activities and interactions, all under the control of the machine that has the process in front of it. Drawing a few blobs and arrows on a whiteboard can be a rewarding experience on its own: things become clear, relationships are exposed, and we can come to a shared understanding (we like to think). But those blobs and arrows do not capture the process with the *precision* required to give that ‘model’ to a computer to execute for us: a goal of Business Process Management Systems. Chapter 13 will take us into that more refined world, where slap-dash will not be sufficient, and where precision rules. When you use *Riva* you have the opportunity to be very precise, whether or not you choose to take that opportunity.

Riva is a method for the analyst. This being so, we shall not be afraid of using precise and specialised terminology between us as analysts. This book introduces a number of detailed technical ideas and terms, essential to the analyst for real understanding of a process, and for the accurate capture of a process in a process model. But one person’s terminology is of course another’s jargon, and the analyst needs to be careful when working with ‘ordinary’ people. My (good) experience is that, whilst I have these technical ideas and terms in my head, ordinary people can work with *Riva* process models happily and productively without them. My (bitter) experience is that the effect on an ordinary person of hearing the word ‘instantiate’ for example is akin to a sharp blow between the eyes with a heavy club: it switches them off. Exercise caution.

Like all methods *Riva* does some things and not others. It is a set of ideas. There are many ideas for different situations. There is no obligation to use all of them at the same time, only to pick the ones that you need. You can use *Riva* in a rough and ready fashion, or you can exploit all the subtleties and precision it offers.

Riva is not a cook-book. There is no recipe and there are no cooking instructions. There are concepts to be used. You choose.

SOME EARLY REFERENCES

Holt’s original exposition of Role/Activity Theory was presented in ‘Coordination system technology as the basis for a programming environment’, A W Holt, H R Ramsey & J D Grimes, *Electrical Communication*, 57, 4, 1983.

INTRODUCTION

Greenspan's thesis on his Requirements Modeling Language (RML) is *Requirements modeling: a knowledge representation approach to software requirements definition*, Sol Greenspan, Technical Report CSRG-15, Computer Systems Research Group, University of Toronto, 1985.

Clive Roberts's and my paper on the IPSE 2.5 work is 'Defining formal models of the software development process', M A Ould & C Roberts, *Software Engineering Environments*, ed. P Brereton, Ellis Horwood, 1987.