

The *How To* of Essential Modelling

Derrick Brown, Course Designer
IRM Training Pty Ltd ACN 007 219 589
Suite 209, 620 St Kilda Rd, Melbourne, Vic. 3004, Australia
Tel: +613 9533 2300

Introduction

Also called *abstract* or *business modelling*, essential modelling can be an extremely valuable tool for the business analyst. Instead of modelling *how* things are done (the current system), or *how* they might be done (a proposed system), we model *what* is done, or *what* might be done. For example the purpose of a Customer Service Department is to provide customers with a level of service they expect (or the company defines). Things like call centres and customer relationship management systems are the *how* of customer service.

This switch in thinking is not always easy as we have to ignore the very practical matters of procedures, methods, people, technology etc. The more involved we are in the system that we're looking at, the more difficult it may be to look at things conceptually. We have to look at what business objective we are trying to achieve. The business analyst who can do this - and explain it to clients and management - becomes a most valuable asset to the business.

Background

The term "essential modelling" was coined by software modelling gurus of the 1970's and 80's. McMenamin and Palmer in their book *Essential Systems Analysis* were among the first to explain it¹.

The reasoning behind essential modelling was sound then and remains so today: if we are to improve or re-design a system then we need a way of looking at it which will help us focus on what it does, what business function it performs. Essential modelling evolved as a technique to help analysts and business users understand the *purpose* of a system. It can be used both in analysis - to understand the purpose of a current system - and in design to describe what is needed. It is an especially useful technique for tackling business process re-engineering as it helps us with an *as is* and *to be* perspective of the process.

Diagramming techniques for essential modelling

Originally the tool of choice for essential modelling was the data flow diagram (DFD) as we were primarily focused on modelling processes. Equally valid today are UML diagrams (e.g. use case) which allow us to model the behaviour of a system without specifying how it will be implemented. A typical UML project is iterative, that is we go through a number of cycles which add more levels of detail. We can apply an essential modelling perspective for our first use case diagram by including only business functions.

At this point we should be mindful that the main purpose of a DFD and a UML diagram is to help us build better systems. They are design tools that help us communicate with developers. In essential modelling however we are using them as analysis tools, to help us conceptualise business processes. You won't find essential modelling defined as one of the 13 UML diagram types but if you already use UML then you'll find it an equally versatile essential modelling tool, alongside the DFD.

¹ Essential Systems Analysis, McMenamin S.M. & Palmer, J., 1984

How do we go about it?

Let's suppose that we are starting up a new business and we want a system to support our main activity. We are planning to have a number of mobile sales staff who will be visiting customers, selling and also supporting our products. They will need access to information – customers, products, prices, delivery - and management will need to know about their activities - customers seen, orders obtained, faults reported.

We can model the key elements of this, starting with a clean sheet, by asking ourselves "what information do we want to obtain, what information will the mobile staff require, what information will we need to store?" Notice the "what" question. It may be difficult to divorce the practical elements from our thinking but initially we need to focus on the *concepts*. For example the idea of playing demonstration product videos on our hand-held PC or smartphone may look like a must-have marketing tool but what if we end up with insufficient memory to store customer orders?!

Our primary objective at this stage is to identify what information is needed - we need to ignore the latest hand-held gadget as the temptation to start tailoring the system to suit the hardware is huge.

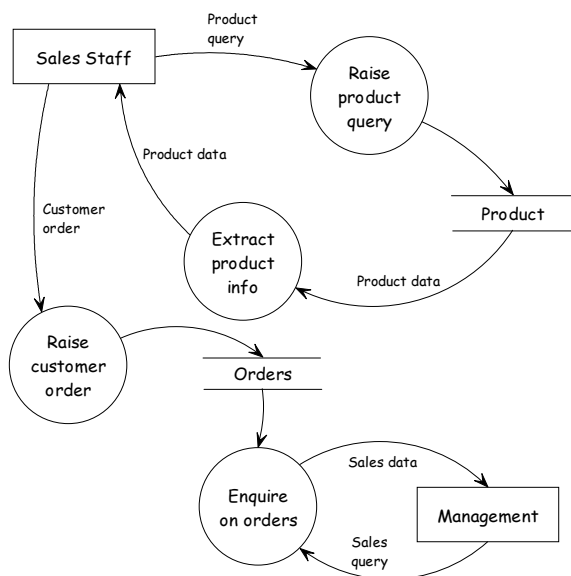


Figure 1

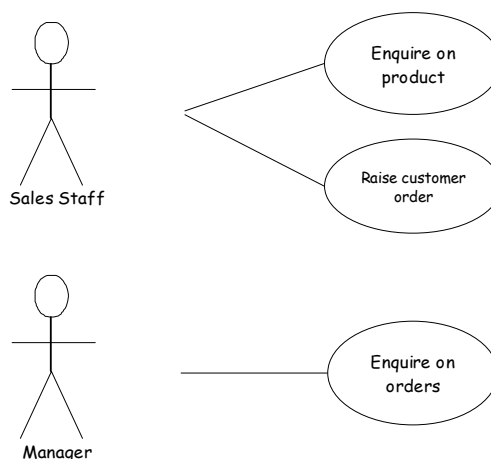


Figure 2

We may produce Figure 1, using a DFD (or Figure 2, a use case) showing our sales staff requesting/obtaining product information and recording a customer order. This order data is accessed by management. We have entirely ignored the real-world issues of mobile data capture, storage, methods of access, response times and so on. If we can work on the concept - and agree it with management, then we can turn it into reality. The latter process is one that most IT people will relish, and that may be why essential modelling is sometimes by-passed!

Analysing existing systems

The previous example is one where we start with a clean sheet. But suppose we are working with a current system? Very often we will have, or may produce, a model of the current system in its real-world state, the current physical system. This will include all of the methods and mechanisms that we've designed in the past to make the system work as effectively and efficiently as we could.

The system may also contain many features that were added subsequently - to get around problems or to support changes to requirements. Some of these may be sound and some may be irrelevant in today's world. In fact, the system may be so complex that it's difficult to see what is going on at all! The dangers lurking here for the business analyst are obvious. It is not uncommon for an organisation to be using a system that is twenty years old, having had hundreds, even thousands of changes made to it including several changes of hardware/software - but underneath it lurks the original system.

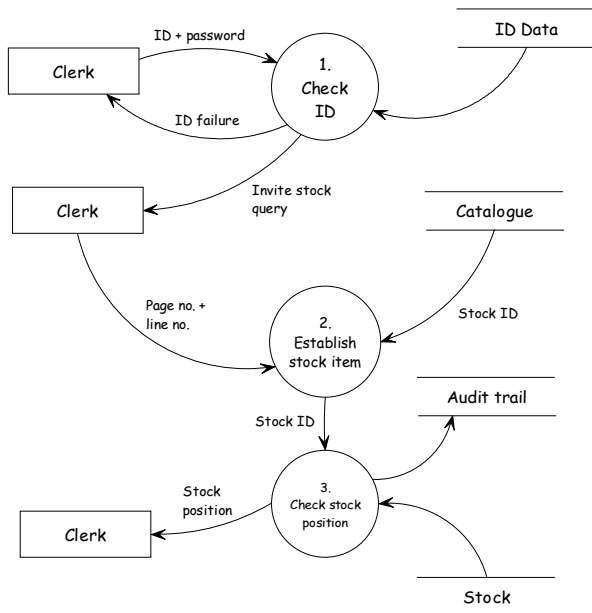


Figure 3

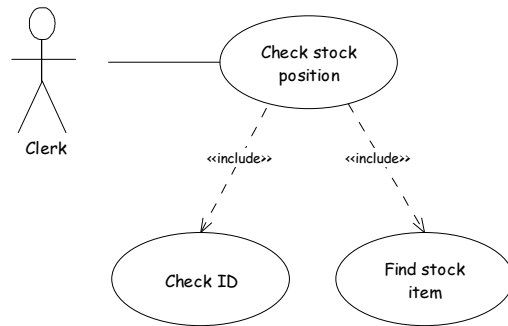


Figure 4

Figure 3 is a current physical DFD (or use case diagram in Figure 4) of an enquiry process, where staff are making stock enquiries. They use a printed stock report and input the page number and line number of the stock item (we may have asked and been told that this is less input than keying in a fourteen digit stock number). Can this current, physical model be reduced to produce an essential model?

Producing the essential model

We do this by looking at each process and asking ourselves "are we doing this for a valid business reason or for some other reason?" Unless we have a positive response, we remove the process. We do something similar for the data as well. We also remove all references, data and processes that are connected with hardware, methods, procedures, geographical constraints, security, audit, back-up and particular software packages.

What we have left should be the basic business processes that we must do in order to achieve the business objectives. This implies of course that we have asked and understood exactly what the business objectives of the system are (what is its purpose, why are doing this?)².

² While "Check ID" (fig. 3 & 4) represents an important security feature, it is a non-functional process which does not contribute to the business objectives of the system. It can therefore be left out of our essential business model along with the related data store and data flows. Note that in figure 3, "Establish stock item" and "Check stock position" are part of the same sequential process which fulfills a single, identifiable business objective. The clerk enters the request for information into the first of these processes and gets the answer back in the second. They are shown as two processes because the old legacy system "always" did in that way (does this sound familiar?). Finally, we have removed the audit data store and data flow as it's a non-functional process.

Figures 5 and 6 on the following page show our current physical system reduced to its' essential business function.

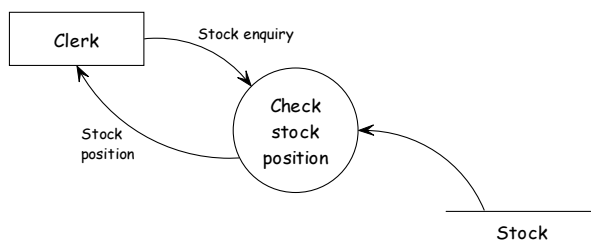


Figure 5

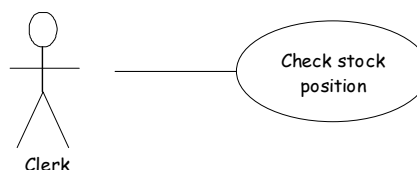


Figure 6

Some business analysts find this quite difficult to work through at first. Some of us are so used to working with the practical, real world that we may be uncomfortable in moving to the abstract. Similarly, business people may find it strange to be removing the audit trail, "we all know it's a 'must have' requirement!" True, but it's a must have, *non-functional* requirement. The remaining *essential system* is now more easily examined and perhaps modified or redrawn.

Using the essential model

Now we can use problem analysis/problem solving techniques to examine the real business issues. We can now ask "What is being achieved here?" "Why are doing this?" We might question whether the "Establish stock item" process is really required or query the use of page number /line number. It is just a means of arriving at the stock item identity. Is there another way of achieving this?

In fact do we need the process at all or is there another way of achieving what we want? In the warehousing/distribution example we've been using here, the traditional approach has been to check stock, then send a shipping order to the warehouse. Why not send customer orders direct to the warehouse - get them to generate back orders for out of stock items - and eliminate a whole process from our system.

By thinking conceptually we may just arrive at a really good solution - and a new essential model. Then we'll convert this to a new design - the new physical model, with security, audit trails etc designed in *from scratch*. This business analysis work can be the most exciting and rewarding work of all - and for the business it might change everything!

© 2008 IRM Training Pty Ltd. All rights reserved.

Send feedback and comments to: training@irm.com.au

You may use this article in your newsletter or internal document free of charge, provided that you do not alter it in any way and include all copyright notices.
