Quick Introduction to Business Process Re-engineering

by Robin Beaumont   Date: 04/02/2000

e-mail: mailto:robin@robinbt2.free-online.co.uk

Contents

Important notice:

This handout is a much simplified version of three handouts:

11.2 Object modelling

11.3 Dynamic modelling and Business Process Re-engineering (BPR)

11.4 Theories Underlying Approaches to Systems Modelling

You can find the original documents at:

http://www.robinbt2.free-online.co.uk/virtualclassroom/contents.htm

1. Learning outcomes check list for the session

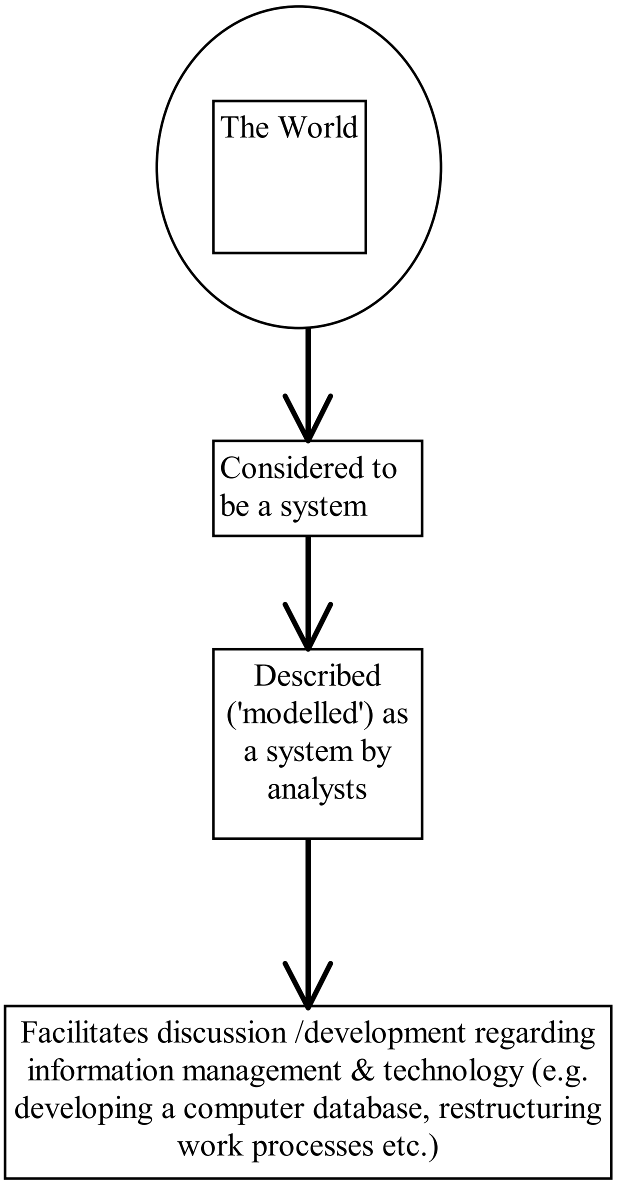
This session aims to provide you with the following skills (the 'be able to's' below) and information (the 'know what's' below). After you have completed the session you should come back to these points ticking off those you feel happy with.

Learning outcome	Tick box
Know the purpose of a model	❑
Know the role 'systems analysts' play in modelling	❑
Understand the concept of the object as used in object oriented (OO) modelling	❑
Be able to develop Sequence diagrams	❑
Understand what Business process re-engineering (BPR) is	❑
Be able to draw Sequence diagrams to carry out BPR	❑

Back to contents

2. Introduction

This section introduces the idea of modelling, more correctly referred to as systems modelling. The purpose of modelling is to provide a description of a system of some type. A system can be a human body, an organisation such as a hospital or frequently a database. A model of a system is a description of it with a particular **purpose** in mind. Developing such models is the principle role of analysts. The 'description' is technical in the sense that it provides the necessary information in the relevant format to allow the system to be analysed/mimicked or possibly supported in some way (i.e. possibly by a computer).


The World
Considered to be a system
Described ('modelled') as a system by analysts
Facilitates discussion /development regarding information management & technology (e.g. developing a computer database, restructuring work processes etc.)

It is very important to note that the modelling process is not attempting to develop a copy of reality. It is much more concerned with describing the important features of the system in terms of processes and information considered important by the analysts and important people in the system ('stakeholders'). Importantly modelling can be used as a method of social engineering.

2.1 Systems modelling as a change agent

A model may attempt to define and develop some utopian type of system which may be subsequently forced upon the original system. This is what has happened to a large extent in the community where systems have been designed to mimic the ideal (from the managers perspective) rather than the actual processes. The dire consequences of such an approach are obvious with user's blame the new system rather than the actual 'change' per se.

It is important to realise that change in any organisation must be handled extremely carefully reducing the degree of anxiety as far as possible. At other points in the notes methods of achieving this are discussed.

See chapter 12:
http://www.robinbt2.free-online.co.uk/virtualclassroom/contents.htm.

Back to contents

2.2 Object oriented modelling

The object oriented approach to systems modelling is just one of many. We will look at some aspects of one particular method of object oriented modelling that was developed by James Rumbaugh, and Micheal Blaha et al called **OMT**. It has been documented in a book by them called Object-oriented modeling and design, Prentice Hall 1991. I have made heavy use of this excellent book for these notes. Back to contents

2.3 Unified Modelling Language (UML)

In 1998 Rumbaugh joined forces with Grady Booch and Ivar Jacobson, who also have their own modelling languages, to create a unified modelling language (UML). The latest version of UML is 1.3 March 1999 (http://www.omg.org/library)

Back to contents

3. Objects

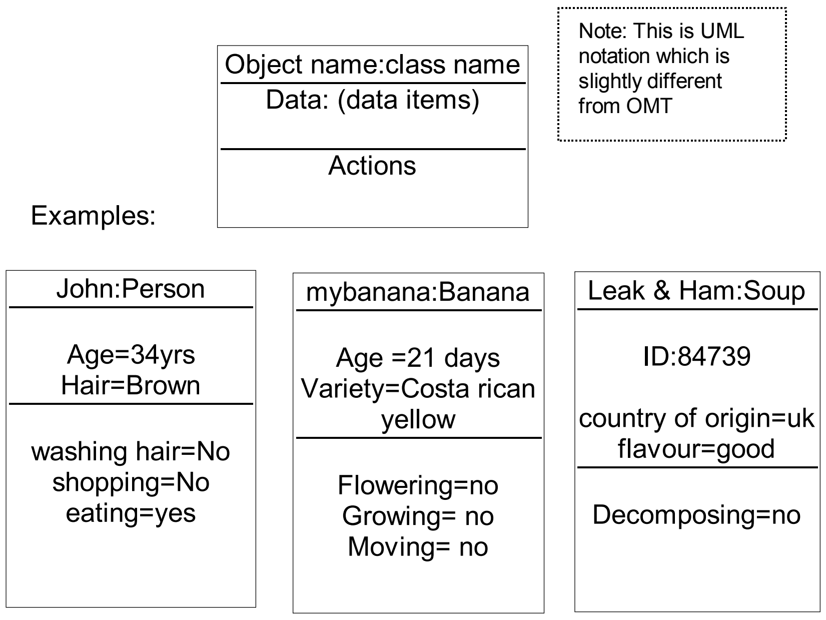
What then is loosely an 'object' in the context of 'object oriented modelling'? An **object** is best thought of as a definable thing in the world that has a number of facets that result in us perceiving it as something that is different from something else. In biology people often talk about something having form and function. That is, what describes the object and what it does. Taking this a little further one can think of the 'descriptive' side of things as being **data** e.g. a person having a name, age, hair colour and address etc. and the 'function' side of things as being **activities** e.g. washing hair, giving birth, going shopping etc. Moreover, the data and activities are inextricably linked in our conceptualisation of the object. If the object should change either its activities or characteristics it exhibits we feel distinctly uneasy. That is when we conceptualise an 'object' in the world, we do not naturally divide it out into these two aspects. For example, when a person sees something they likke they tend to think of actions as well as the objects characteristics (data). Taking this further, when one thinks of a banana or a bowl of soup not only do both have pleasant characteristics such as colour and smell but also have pleasant actions associated with them such as the soup being made and the banana growing.

In the context of 'object modelling' an object is therefore something in the investigators' mind that usually possesses a recognisable number of both characteristics (data consisting of data items) and actions. As in all areas of reserach one word is never enough and both OMT and UML uses the word '**attribute**' instead of data or characteristic to mean the same thing and uses the words '**operations'** (UML) or '**methods'** to mean activities/actions. I will use the words **data**, **data item** and **activity** for this handout.

The diagram below shows how Objects are drawn in UML. Each object is divided up into three sections:

1. Top Gives the name of the object along with its class name (more about that below)
2. Middle section lists each data item along with the value it has for the particular object
3. Bottom section lists each activity along with the value it has for the particular object

Objects (UML)



Back to contents

4. Classes

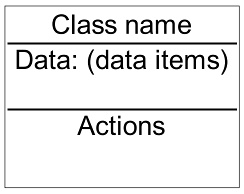
The best way to think of a class is to think of a 'thingy' - something you know intuitively exists but are unable to explain: it is often called a concept although philosophers argue about the differences. A class can be thought of as an abstraction from the object. Notice that in the above examples a single banana was described. The object 'mybanana' is said to be of type class banana. Similarly John is said to be of class men and Leak and Ham of class Soup. Notice that while the class men could exist without any objects of that type the reverse is not true. We say that an object is an '**instance**' of, or an '**instantiation**' of a class. This may appear rather pedantic but it comes in very useful in modelling as it provides a method of classifying groups of things often in very useful ways. Basically:

Class = Template

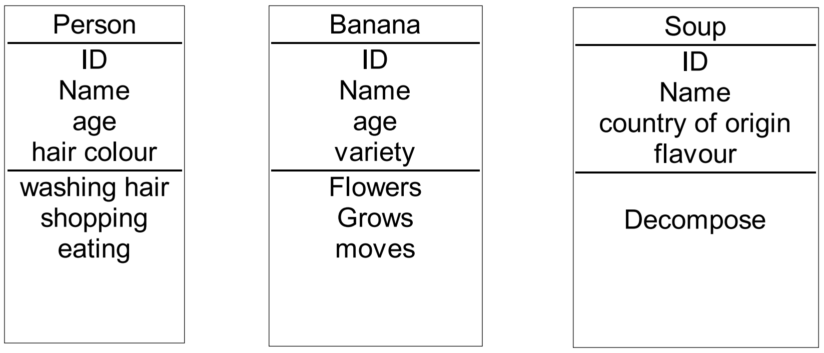
Object = Instance

Computer scientists whenever possible use diagrams or symbols rather than words. This is because words are considered to be the most ambiguous form of communication, or more probably because most computer scientists are notoriously poor at English. The diagram below shows how UML and OMT would possibly represent several of the classes described above:

Back to contents



Examples:



In the diagram above each box represents a class. The box is divided into three sections. The top section of the box provides the name of the class, usually a noun. The next section down provides details of the data (attributes - data items) in the class while the last and final section provides names of the activities (methods) that make up the class. While the first two examples, of person and banana, are relatively easy to define in this way the third is more difficult. This is probably because Soup has a large number of attributes but has very few activities. The only one I could think of was the process of decomposing when it's left too long. Not all classes have actions but all must have a name and something that uniquely distinguishes them from another class. Because objects are just instances of a particular class these rules also apply to them as well.

Back to contents

4.1 Class diagrams and amount of detail shown

In a 'class diagram' a varying degree of detail of a class can be shown; sometimes all that is shown is the class name; alternatively the class name and the data are shown, or alternatively as above, details of the data and actions are shown.

Exercise 1

Consider your job to be a class. List two of your clinical activities along with the information about them which would be useful to collect. If it helps consider it in the context of a learning log book of some sort.

Exercise 2

Draw a diagram that represents one or more of the following Classes; Community nurse, GP, Clinical manager in a hospital, Day care co-ordinator, Director of finance in an acute trust, patient, Yourself from the perspective of your boss.

Exercise 3

Draw an object diagram that represents an instance of some of the classes you described in the above exercises.

Back to contents

4.2 Views

At the beginning of this section it was stated that the identification and description of objects was partly a matter of individual interpretation. This is clear when one considers the examples given above. Now considering the NHS a 'patient' as perceived by a hospital chaplain is a very different object from that perceived by a director of finance. If you asked each to draw an object diagram like those given above, each would come up with a different set of data and activities.

A model is always context specific.

It is never purely a reflection of reality - whatever that is!

It is interesting to note that a large amount of effort in the modelling process is directed towards getting these differing views consolidated in some way. The potential problem is tackled in different ways depending upon the particular software development lifecycle chosen. Frequently a 'stakeholder' or cultural / political analysis is carried out to see how important each of the views are thus allowing a way to prioritise one view over another. Another method is used during the actual software development phase of the lifecycle where different 'views', that is user interfaces, to the data can be developed for different user types. These are just two of a whole host of techniques available to tackle this problem.

The importance of being aware of potential problems with differing views in the health care sector is discussed in

Willcocks P L. Mark A L. (1989) IT systems implementation: Research findings from the [health care] public sector. J I T [June] 92 - 103

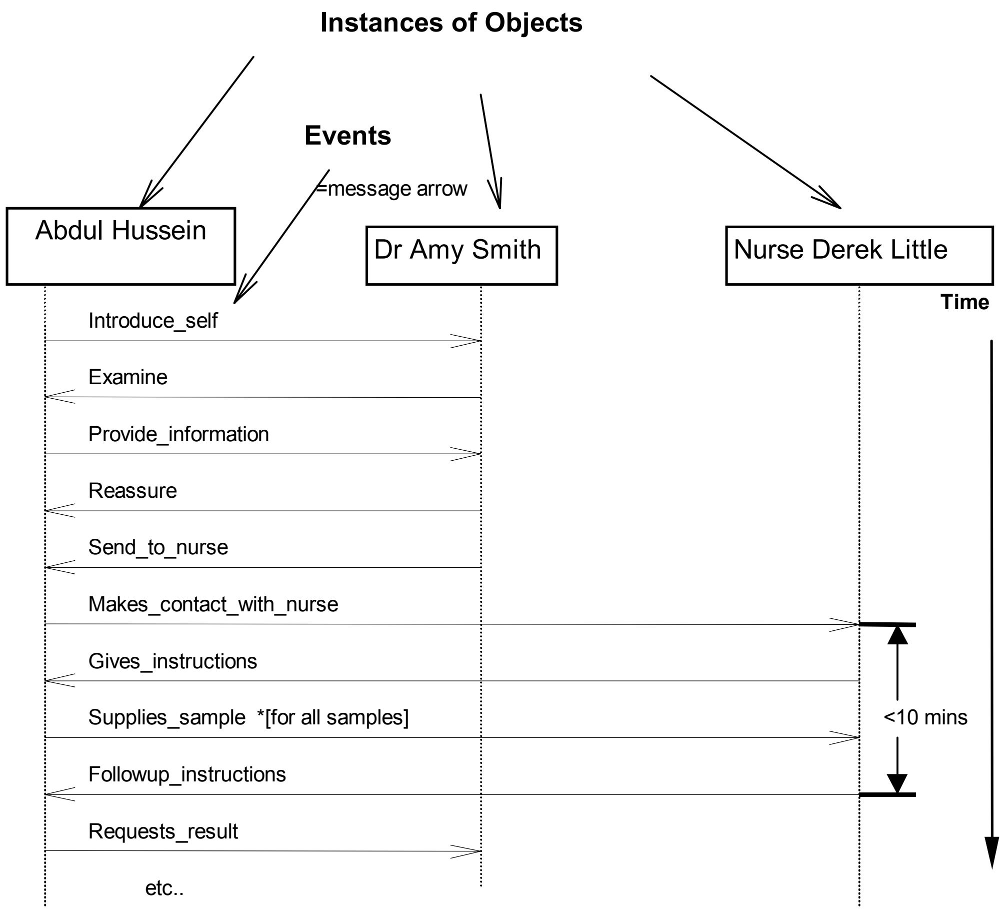
Exercise

Describe what might be the main differences between some of the following: Community nurse, GP, Clinical manager in a hospital, anaesthetist, Day care co-ordinator, paediatrician, psychiatrist, microbiologist, Director of finance in an acute trust, patient.

Back to contents

5. Sequence Diagrams (Event traces in OMT)

Sequence diagrams provide a graphical account of the order of events that might occur between the objects being modelled. It can be thought of as a **scenario**. Consider the example below of a sequence diagram describing the interaction that might take place between a patient, GP and a nurse:



The three vertical lines (called **lifelines**) represent the three instances of objects in the system under study, Abdul Hussein is a instance of the class patient, Dr Amy Smith is an instance of the class GP and Nurse Derek Little is an instance of the class nurse. Instances are used instead of the more abstract classes because we are concerned with a particular scenario not general behaviour at this point in the modelling.

There is no significance to the horizontal ordering of the objects.

The events (shown by **message arrows**) have names (this is not obligatory) and pass from one object (the sender) to another (the target). Although it is not shown in the event trace diagram above it is possible for an event to have more than one target (i.e. a double or multiple headed arrows). Time moves down the event trace diagram but the exact vertical distance between arrows is not important. However you can show a time period between events as demonstrated in the above also various labels (such as timing constraints, descriptions of actions during an activation, and so on) can be shown either in the margin or near the transitions or activations that they label.

You can show the data that passes with the message arrow by including it along the arrow.

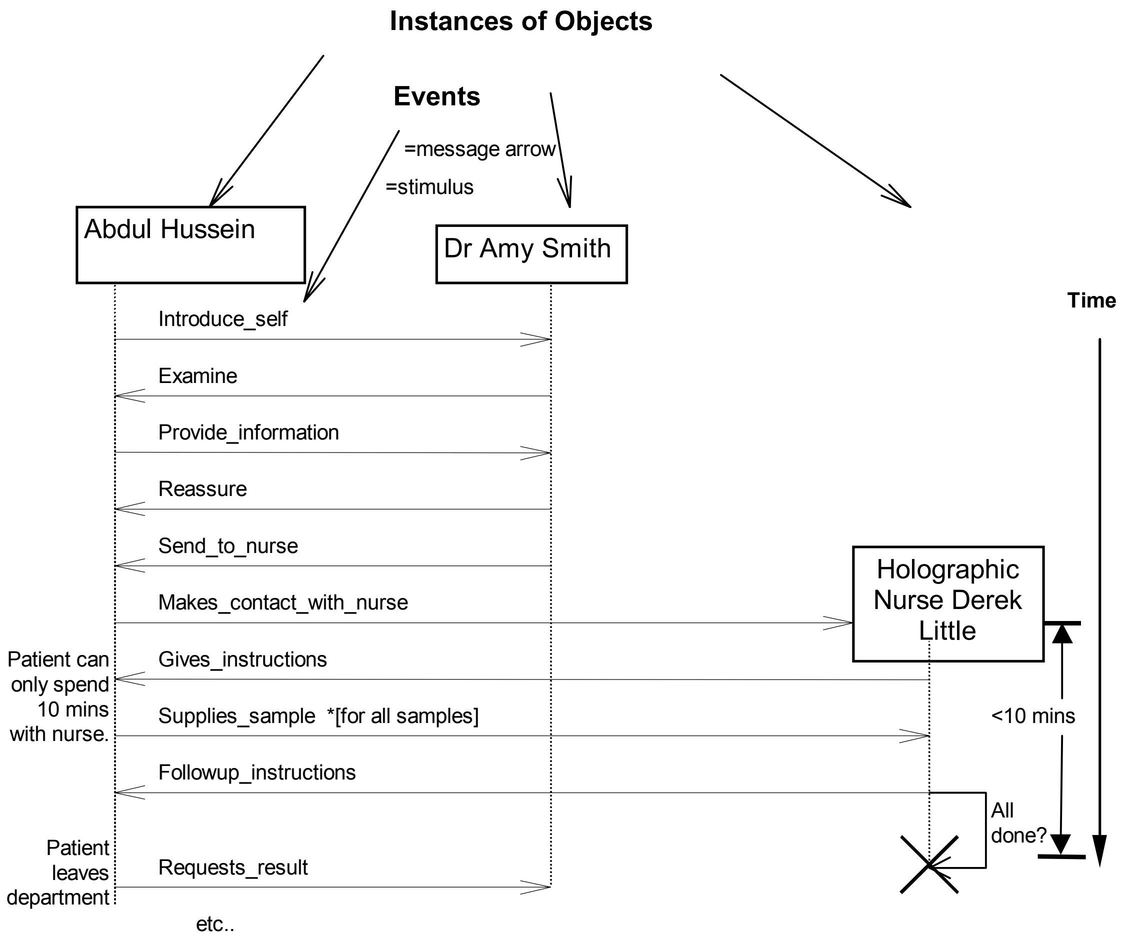
You can also show repetition (technically called **iteration**) with the use of an asterisk (*****) along with a description in square brackets beside it. The above example uses it for the supply samples event.

Back to contents

You can add further complexity to a sequence diagram by using additional symbols.

"If the Object is created or destroyed during the period of time shown on the diagram, then its lifeline starts or stops at the appropriate point; otherwise, it goes from the top to the bottom of the diagram. An object symbol is drawn at the head of the lifeline. If the Object is created during the diagram, then the arrow, which maps onto the stimulus that creates the object, is drawn with its arrowhead on the object symbol. If the object is destroyed during the diagram, then its destruction is marked by a large "**X**," either at the arrow mapping to the Stimulus that causes the destruction or (in the case of self-destruction) at the final return arrow from the destroyed Object. An Object that exists when the transaction starts is shown at the top of the diagram (above the first arrow), while an Object that exists when the transaction finishes has its lifeline continue beyond the final arrow." (UML 1.5 p311)

For example Imagine that you're in Start Trek working on the Enterprise and Nurse Derek Little is a Hologram which you can create and destroy at will. You would represent this by moving the object name down to where he is created and use the 'X' symbol to show his destruction:



The above sequence diagram also includes a **self-delegation**, a message that an object sends to itself, by sending the message arrow back to the same lifeline.

Back to contents

5.1 Further adornments

You can also add other embellishments (called technically **adornments**) to sequence diagrams, it is basically up to you how much you want to add. For completeness a list of some of them is provided below:

Sequence number - "The arrow may also be labelled with a sequence number to show the sequence of the Stimulus in the overall interaction. Sequence numbers are often omitted in sequence diagrams, in which the physical location of the arrow shows the relative sequences (UML ver 1.5 p313)

Guard condition - A Stimulus may also be labelled with a guard condition." (UML ver 1.5 p313) such as 'if patient weepy' then carry out reassurance.

Activations - An activation (focus of control) shows the period during which an Object is performing an Action either directly or through a subordinate procedure. It represents both the duration of the performance of the Action in time and the control relationship between the activation and its callers (stack frame). An activation is shown as a tall thin rectangle whose top is aligned with its initiation time and whose bottom is aligned with its completion time. The Action being performed may be labelled in text next to the activation symbol or in the left margin, depending on style. (UML ver 1.5 p312)

The basic idea of a sequence diagram has been around for some time and as the UML reference manual (ver 1.3 p306) states. "Much of this notation is drawn directly from the Object Message Sequence Chart (POSA) notation of Buschmann, Meunier, Rohnert, Sommerlad, and Stal (1996), which is itself derived with modifications from the Message Sequence Chart notation." Jacobson 1995 calls them Interaction diagrams (p132).

Back to contents

5.2 Incremental development

In the process of modelling the modeller gradually develops a number of scenarios considering first how the system would behave in normal conditions and then when something untoward might happen e.g. the patient passing out or the nurse not being available etc. Finally the modeller consolidates them by comparing and contrasting each scenario. UML version 1.3 suggests you amalgamate then if possible! The modeller may also consider how to improve the system in some way which is the subject of the next section.

Exercise 1

Device a sequence diagram for two situations (scenarios) involving yourself at work.

Exercise 2

Device a sequence diagram for one of the following situations:

items of service claims forms for GPs,

Clinical audit within a Hospital department,

Multidisciplinary assessment in the community,

Commissioning negotiation.

A & E Assessment

Care Programme Approach review meeting in Psychiatry

Back to contents

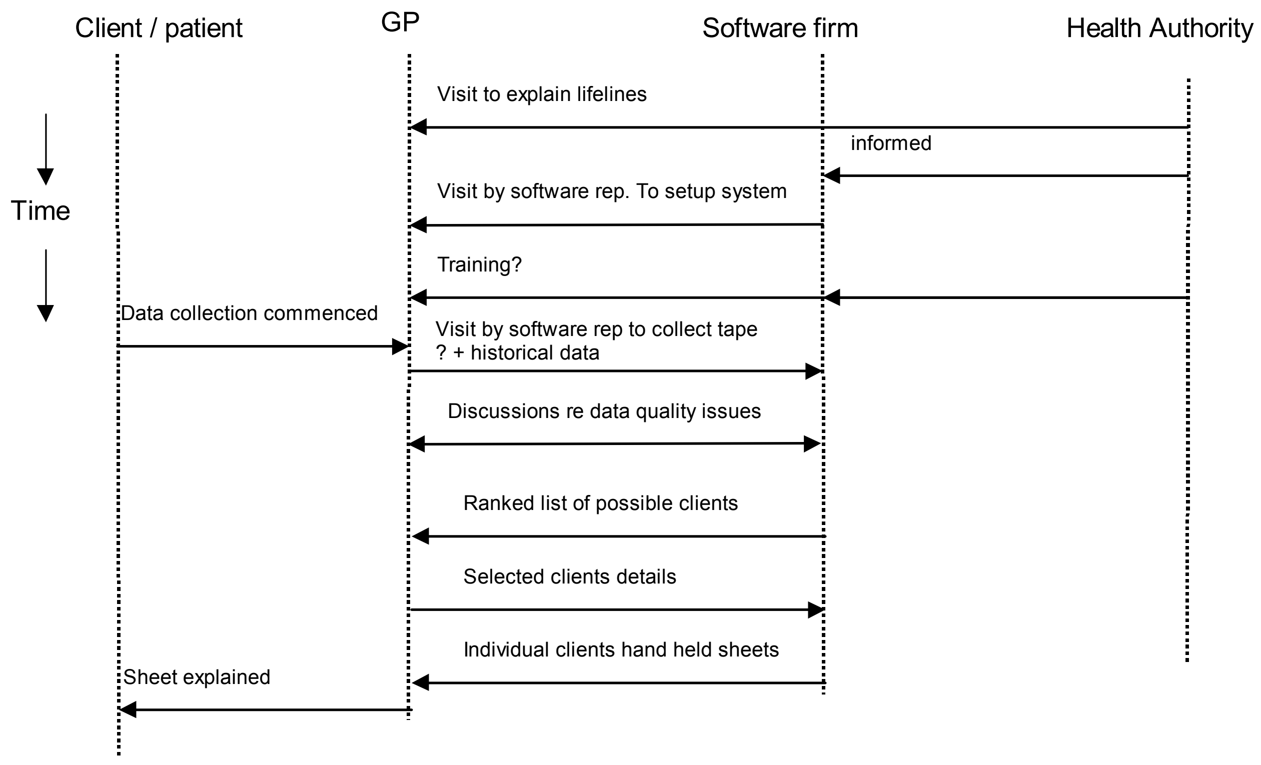
6. Business Process Re-engineering (BPR)

This very impressive term is really nothing more than analysing an organisation using amongst other techniques Sequence diagrams. But as would be expected they are given numerous other names in BPR. BPR uses the sequence diagram concept to consider how the events in an organisation might be re-organised for more efficient and or effective behaviour.

I feel an example coming on:

A particular health authority are running a project called 'heartrisk'. This is a GP community based project which provides data concerning individual patient risk of coronary heart disease based upon measuring various risk factors (smoking, drinking, weight etc.) to patients in the form of a hand held sheet. At the present time a representative visits practices they feel might be interested in participating in the project. If they are interested a representative from the external software company is informed who in turn informs a colleague who visits the GPs and sets up the GP system to collect the relevant data. A representative of the software company or the health authority then visits the site to train them. A further visit by a member of the software firm collects a copy of the GP system backup tapes to obtain the necessary data after a suitable time period. Following initial data analysis by the software firm in consultation with the practice a ranked list of each patients CHD risk score is printed out and sent to the practice. The practice then decides who would most benefit from the leaflets and requests them from the software firm. These are then posted out to the practice which contacts the patients to offer them advice if they haven't died already!

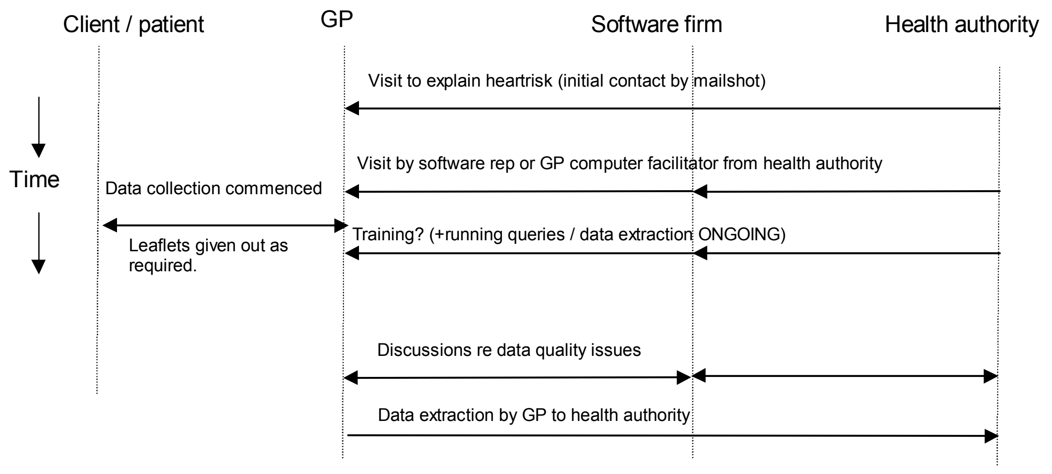
Present sequence diagram for Heartrisk (computerised practice)


Client / patient
GP
Software firm
Health Authority
Time
Visit to explain lifelines
informed
Visit by software rep. To setup system
Training?
Data collection commenced
Visit by software rep to collect tape ? + historical data
Discussions re data quality issues
Ranked list of possible clients
Selected clients details
Individual clients hand held sheets
Sheet explained

The options available to improve this scenario are numerous. Lets consider the one simple strategy of allowing the GPs own system to do what the external software firm now do to a limited extent. The new event trace is given below. It immediately looks much less complex. Event traces are extremely useful for analysing situations such as the one above.

Back to contents

Embedding 'heartrisk' functionality in the local GP system - Sequence diagram


Client / patient
GP
Software firm
Health authority
Time
Visit to explain heartrisk (initial contact by mailshot)
Visit by software rep or GP computer facilitator from health authority
Data collection commenced
Leaflets given out as required.
Training? (+running queries / data extraction ONGOING)
Discussions re data quality issues
Data extraction by GP to health authority

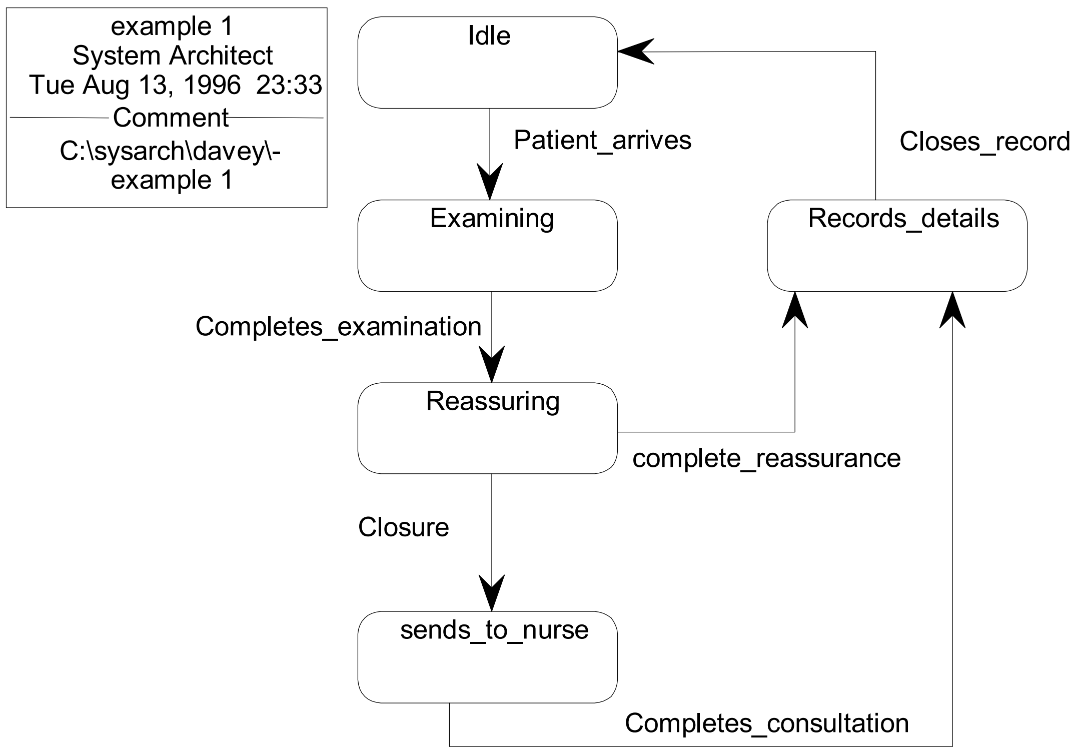
Back to contents

7. Behaviour of a individual object over time: State Diagrams (Harels Higraphs)

Where as the sequence diagram showed the interaction between object a state diagram provides details for a single object.

State diagrams show visually the relationships between events and states FOR A SINGLE OBJECT. An example is given below:

State diagram for GP (see event trace)



The first and possibly most important thing to note is the title. The state diagram relates to one particular class, in this instance that of the GP described in the event trace provided earlier. Comparing the above state diagram with that of the event trace we note that only those events (arrows on the event trace diagram) which either originate or end at the GP class (Dr Amy Smith) are represented in the above state diagram. Thus the event makes_contact_with_nurse is not shown as it does not either originate or terminate at the GP object.

The rounded boxes represent states and the lines represent events (often called transitions).

8. Limitations of BPR

Although BPR methods have been overall a success they have not lived up to their initial expectations.

The original idea of BPR was that it provided the opportunity to radically restructure organisations however this has found to be inappropriate (and possibly dangerous) in complex situations where a **more gradualist, participative approach** is now considered to be appropriate.

The problems lie with both the methods and implementation of change. BPR methods are crude, using sequence diagrams, no matter how complex, and scenarios regardless of how many considered, may not capture all the roles that a particular medical secretary or departmental porter undertakes. Because of this more qualitative (i.e. ethnographic) methods such as video analysis has been attempted, unfortunately this is extremely time consuming and the results are of questionable quality regarding generalisability. As discussed at the beginning of the handout the problem may also be the implementation of change, for example job satisfaction, informal methods of communication, etc. may have been left out of the analysis with disastrous consequences.

9. Summary

Dynamic modelling is gaining in importance as it becomes clear that often the problem with systems is how the components work together and not the individual elements in the system. It is therefore vitally important to develop valid methods which allows the analysis of these aspects.

Back to contents

10. Appendix - Additional information

For the UML view consult the online reference manual at http://www.rational.com/uml/resources/documentation/index.jtmpl This has links to most UML sources including all reference documents including the most current.

UML offers the following diagrams (UML ver 1.3 p230):

- use case diagram
- class diagram
- behaviour diagrams:
    - statechart diagram
    - activity diagram

    Interaction diagrams:
        - sequence diagram
        - collaboration diagram
- Implementation diagrams:
    - component diagram
    - deployment diagram

Elements can be grouped together into what are called **packages** (UML ver 1.3 p231)

Back to contents

11. References

Alavi M Wetherbe J C 1991 Mixed prototyping and data modelling for information system design. IEEE software May 87 - 91 One of very few articles which uses an experimental design in place of the purely non experimental case study, to assess various methodologies.

Bertalanffy Ludwig von 1968 [but still published] General System Theory; Foundations, Development, Applications. George Braziller New York.

Blaha M Premerlani W 1998 Object-oriented Modeling and design for Database applications; Includes UML. Prentice Hall

Buschmann F Meunier R, Rohnert, Sommerlad, Stal M 1996 Pattern-Orientated Software Architecture: A System of Patterns. John Wiley & Sons

Checkland P 1981 Systems thinking: sytems practice. John Wiley An excellent introduction to the history of systems analysis as well as introducing Checklands own 'soft systems' methodology.

Checkland P Scholes J 1990 Soft Systems Methodology in action John Wiley. This book includes a case study from the NHS.

Fowler M Scott K 1998 UML distilled: Applying the standard object language. Addison Wesley

Friedman AL Cornford DS 1989 Computer Systems Development: History, organisation and implementation. John Wiley. A wonderful book, excellent references, clear penetrating insights. etc.

Jacobson I 1995 The Object Advantage. Addison-Wesley.

Klein HK Hirschheim RA 1987 A Comparative Framework of Data Modelling Paradigms and Approaches. The Computer Journal 30 1 9 - 15.

Martin James 1989 - 1990 Information Engineering 3 volumes now published by Prentice Hall. The classic texts concerned with business process analysis. Very easy to read. £30 each volume. Moves from information strategy planning through business area analysis, technology impact analysis, to low level design. Firmly based in the business process paradigm.

Newman M. Rosenberg D. 1985 Systems analysts and the politics of organisational control OMEGA int. j. of mgmt sci., 13 (5) 393 - 406

NHS Executive 1997 Business Process Reengineering in the NHS. NHS executive UK

Rumbaugh J Blaha M Premerlani W et al 1991 Object-Oriented Modeling and design. Prentice Hall.

The Use of Conceptual Graphs for Knowledge Bases, Customisation and Actual Data Organisation, Joubert M, Unpublished project working paper (AIM NUCLEUS)

Wood-Harper AT Fitzgerald G 1982 A Taxonomy of Current Approaches to Systems Analysis The Computer Journal 25 1 12 - 16.

12. Useful Internet links:

http://watson2.cs.binghamton.edu/~dwhitloc/uml/paper/part3.html A short three part tutorial about UML from David Whitlock at Binghamton University

http://www.rational.com/uml/resources/documentation/index.jtmpl Links to most UML sources including all reference documents including the most current.

http://www.rational.com/uml/resources/quick/index.jtmpl Quick reference guide to UML

http://www.omg.org Object Modelling group - International body responsible for setting OO standards

http://ourworld.compuserve.com/homepages/Martin_Fowler/ Martin Fowlers home page. One of the main writers on UML

http://www2.awl.com/cseng/titles/0-201-89542-0/techniques/index.htm Martin Fowlers introductory stuff on Object Oriented analysis and design

http://www.db.informatik.uni-bremen.de/umlbib/ Allows you to search on current UML papers and conference proceedings

http://members.aol.com/acockburn/papers/usecases.htm Alistair Cockburns stuff on use cases

Back to contents

Document Info:

Robin Beaumont Tel:(UK) 0191 2731150 e-mail: robin@robinbt2.free-online.co.uk Source: Laptop; C:\HIcourseweb new\chap11\s5\bprintro.doc 04/02/2000 23:11