

Analysis and Design for Process Support Systems using Goal-oriented Business Process Modelling

By Denise Downs, Dr Ken Lunn

School of Computing & Mathematics, University of Huddersfield, England

Summary

This paper focuses on the ideas that it is important to start with a business process model which fully articulates the vision and goals for the new process (at whichever level the process is within the organisation) when producing software. Furthermore, these goals need to be auditable through the process to ensure the developed system assists attainment of the process and its goals.

A small team at Huddersfield University are developing a methodology which commences with vision and goals of the business process, requires capture of the new process but facilitates development of process support systems as well as standard transactional systems and ensures the requirements of the process and its goals are met and traceable. While at the early stages the methodology intends to build on current methods and techniques and at this stage has mainly highlighted issues with current approaches.

Transactional systems have dominated the literature in terms of design and software development. This is a view reinforced by the use case approach to software requirements [Rosenburg 1999], where the system is seen as a set of largely distinct functional groups. This approach has led to a style of analysis and design that leaves a lot of business issues outside of the system, and implicit in the analysis.

Process support systems are being perceived as increasingly important to a number of companies. This reflects a growing maturity in organisations, which are keen to automate more of their activities. Application areas, such as call centres, can achieve significant productivity and quality improvements by utilising process support and workflow systems to improve information flow, speed up response, and reduce problems in dealing with customers. Customer relationship management is requiring greater understanding of the process goals and values, and a consolidation from transactional systems to provide for the business process of managing customers with the capture of the relevant process data. In essence these types of activity are best captured by business process models.

Workflow approaches have been reported in the literature, but they seem orthogonal to much of the object-oriented analysis and design approaches that are documented. The authors' experience in analysing and designing process support systems hit problems when advice was sought from experts in other methods. Structured methods experts wanted to construct data flow diagrams that did not adequately reflect process flows in the business. Object-oriented experts wanted to look for use cases and object representations, without too close a consideration of the process flow; indeed, modelling process flow in objects began to look over-complicated. None of them start with the process goals, vision and values and provide a traceable audit of ensuring these are met through the design.

When the software design reflects business processes the resulting software seems to be more intuitive to users and thus easier to use. Further, we contend that full acknowledgement of goal-driven business processes leads to a more comprehensive design that better meets user

requirements. However the transition from business process models to software design is not a trivial one. The only method we have come across that approaches this adequately is the Select Perspective [Allen 1998]; this approach looks at use case identification from fully analysed processes using a notation similar to IDEF; alas this method has not been fully documented in the literature.

In this paper, we describe our approach to analysing and designing a process support system. We begin with a statement of the vision and goals of the business, and an analysis and assessment of the current business processes. We then produce a revised business process description, and then seek an object decomposition and set of user interfaces to support the business processes. We have in fact missed out an explicit use case step, and we do not see this as a problem in this instance, as discussed later.

1 Background

The School of Computing and Mathematics' Placement Unit undertook a project to encapsulate the activities carried out at the School of Computing and Mathematics' Placement Unit. The project, MaPPiT (Mapping the Placement Process with Information Technology), has resulted in the development of a complete Electronic Process Support system, designed and developed using Lotus Notes, with subsequent Web-enabled support developed in Lotus Domino. The intention was as much to encapsulate and communicate best business practice as well as to develop an IT system. The system is now operational and supports approximately 200 placement students a year, and both processes and system are now adopted by a number of other universities. The system supports the tracking of students and work placements. It provides support to students, potential employers, academic staff and support staff through the whole process of applying for, achieving and undertaking a work placement. This includes management of online CV's through to acceptance of work-based assignments that are used as part of the assessment of students on placement.

2 Analysis and Design Approach

We split the analysis and design into two phases. The first phase is to comprehensively map out the desired business processes, defining goals and vision, and identifying issues. This is comprehensively refined through the following stages:

- Articulated Goal, Vision, Current Issues/Problems with As-Is Process
- Identified Core/Key Processes, Management Processes & Support Processes
- Capture tasks in current process on cards
- Working groups put tasks in cohesive groups
- Process Names for groups decided
- Process Designed by work groups
- Check with original tasks for completeness
- Document Business Process Designs with roles, triggers

The Development Phase is broken down into the following stages:

- Identify objects
- Identify Process Tracking Objects
- Identify Attributes – data and status
- Identify Business Pattern Objects like To Do
- Produce object diagram for relationships
- Produce User Interaction Matrix for each object using Roles from BPD & PFD
- Organise User Screens (referred to as user agenda's) to reflect Process

2.1 Goal of the process(es)

We begin our analysis by defining the vision and goals of the overall process. The collection of goals is a complex activity. Inevitably there are conflicting views on what are the appropriate goals of a process, depending on the views of stakeholders. However, we view it as essential that there is agreement on the overall goal of the business process(es) before commencing with analysis. Our agreed goal for the MaPPiT system is: *“To secure, develop and monitor rich learning experiences (on placement) that build on students' current skills and knowledge, in line with their career aspirations; enhance their employability through experience in the workplace; and increase their skills and knowledge, subsequently enabling higher levels of achievement.”*

2.2 Business Process Modelling

The business process model describes the activities of placing students in terms of a series of inter-linked business processes. It is a static model users can tailor to their own requirements. It tries to satisfy a number of aims, including:

- To document the placement processes, following a review of their efficiency, to ensure that necessary processes are included and nothing is omitted.
- To address issues which had emerged from the analysis and other projects, eg. some placement units operate in an unprofessional way: missing deadlines, not meeting requests..
- To facilitate a move from a placement activity run predominantly by academic staff to one involving more administrative staff; at Huddersfield there are two full-time administrative placement staff in the unit; this allows us to be more professional and responsive but requires greater co-ordination and attention to communication.

These aims give an understanding of what the model is about, and guide the definition of the processes. The focus is very much an operational one: the project team investigated what is done, why, how it can be improved, what the students think, what companies think, etc., and looked to define good practice in the model. As part of the operational focus, the team examined what materials are used to prepare students for placements, the training given and the skills developed.

Issues this raised were how do we audit our goals to ensure they have been addressed in the resulting system ?

2.2.1 The Processes and Symbols of the Model

The processes of the model are grouped into three types: Core, Management and Support. The underlying rationale behind these types is:

Core Processes	These are key to the business unit and represent the core activities. They must be undertaken within the Placement Unit
Management Processes	In general these can be managed by individuals not involved on a day to day basis with the placement team and represent review and monitoring/directing processes. They must involve representatives from the placement team
Support Processes	These can generally be outsourced to other supporting groups. Small placement teams, who do not have the resources for the full model, may 'pass on' responsibility to other personnel within the school/department (e.g. careers, personal tutors, pathway leaders, commercial

2.2.2 Symbols Used in The Model

The models we developed used a notation similar to process flow diagrams (PFD) in IDEF3, shown as a flow of activities (see Figure 1). However, because we knew the next stage was to develop a software application to support the running of these process flows, we paid particular attention to the triggers (see Figure 2) which either started the process or were required to keep it running – each wait state had a trigger identified for the next stage. We also had a time trigger for those tasks we wished to keep on time for quality customer service, but were dependant on external triggers (eg. Receiving a CV from a student). In effect, if the process flow did not progress on it’s own naturally (e.g. an employer did not contact us to organise interviews after receiving CV’s), the time trigger would ensure we built into the software the ability to manage these threads.

On each activity within a process a suggestion is made, through a label, of the role felt to be appropriate to carry out that activity. Activities are marked as essential, desirable or optional. The team believes that all those which are marked as essential should be carried out by any placement unit; desirable activities are those which are felt to bring benefits, and optional activities are for larger placement units to consider. This perspective also provides a priority system when considering which processes to adopt/review.

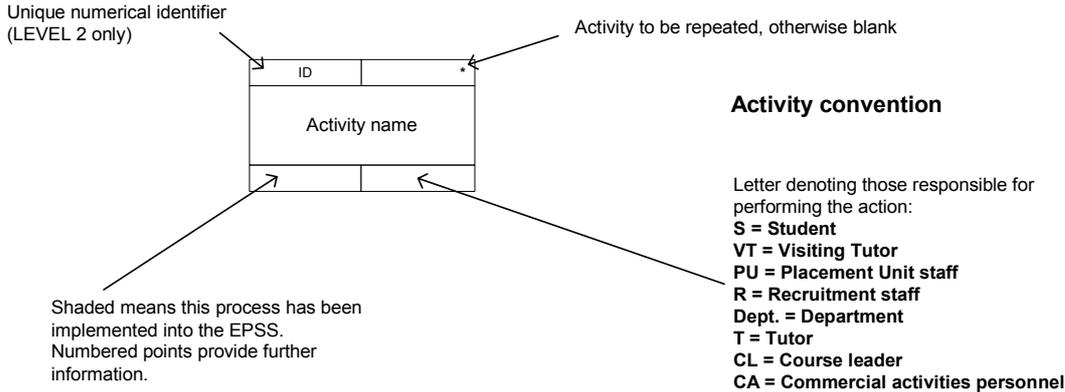


Figure 1 - notation for an activity

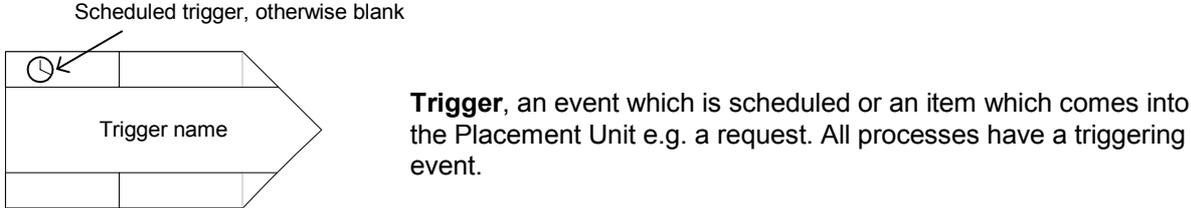


Figure 2 - notation for a trigger

The result of our business process design was approximately 50 process flow diagrams, as illustrated in Figure 3. Many of these were much more complicated with branching and joins. Many of the tasks were pure people tasks, many were people tasks that required the IT support system, very few were IT only because of the type of process we were modelling – the placement of students into industry for their 12 month work experience. While very similar to a recruitment process we also had training and monitoring of the placement. The

fact that there were no substantial IT-only tasks, was a factor which made the transition to information system more difficult.

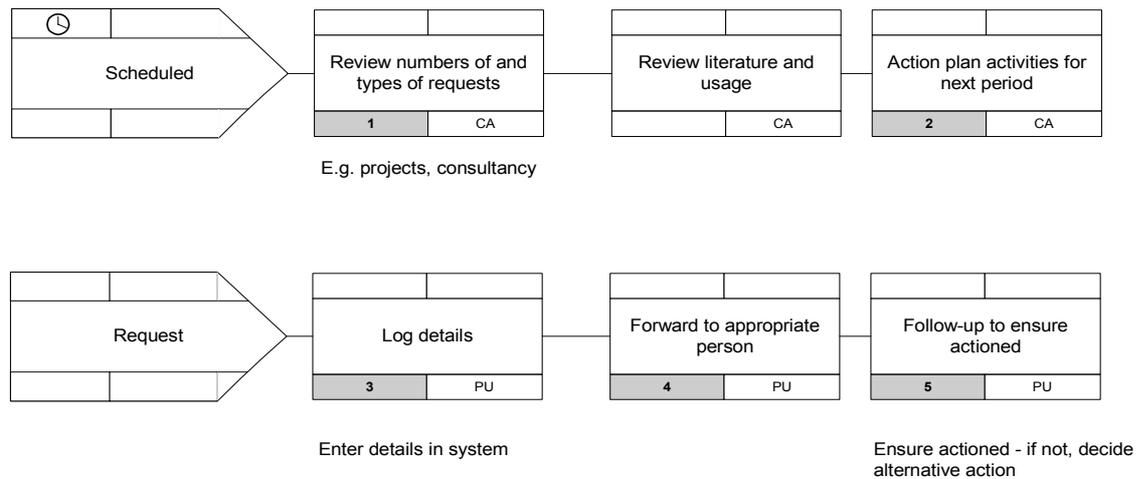


Figure 3 - typical business process flows

2.3 Design and construction of the information system.

The delivery platform we use is Lotus Notes. Lotus Notes has an object database with hierarchical relationships. Collection classes (views) can be created from simple queries on the database. Each object is associated with a form which provides the user interface to the object data as well as the visibility to the designer of the data items in the object.

Our approach was to first trawl through the PFD'S and identify all our objects. This represented a comprehensive domain model. What this did not initially reveal was the objects we required to manage our process threads. This was the next trawl and we integrated these into our object model so we could document the relationship with our real world objects. Experience also told us that we needed some pattern type objects like 'To Do' and 'Activity Reports' to track and audit problems or special cases we had to deal with within the process. These were difficult to document on the object model as generally they were going to be made available to most objects in the system. This availability was documented in a User Interaction Matrix (e.g. Figure 4). The next phase was to identify all the attributes for each of our objects. Some of this was from experience e.g. Company obviously requires name, address, tel No, but some were status type attributes (placed/unplaced; month to contact) detected from the PFD's.

The next issue was that as our flow diagrams were decomposed to the third level to identify the tasks, so a number of different diagrams affected the functionality of a number of objects. Designing what actions should be visible to the user at what stage of the process and what that action should do if selected was problematic. The solution was a User Interaction Matrix (e.g. Figure 4). This allows us to identify what methods (via buttons/actions in Lotus Notes) were available to which user groups (roles) interacting with the object and when.

Actions	Close	Save	Job Description	Interview	Correspondence	Complete
Action Detail:	Close Form	Save details close Form	Create new Job description details	-Update form with date -Add interviewees -create interview details form -Schedule follow-up contact	Prompt for standard letter to send Create letter with contacts details Save letter and print	Prompt for outcome Set status to successful or unsuccessful
Visibility	All	PUAdmin,	PUAdmin,	PStaff	PStaff	PStaff

		PTutors	PTutors			
Stage	✓	✓	✓	×	✓	×
New						
Diary reminder	✓	✓	✓	×	✓	×
Waiting to advertise	✓	✓	×	×	✓	×
Advertised	✓	✓	×	✓	✓	×
Interviews	✓	✓	×	✓	✓	✓
Successful	✓	×	×	×	×	×
Unsuccessful	✓	×	×	×	×	×

Figure 4 - User Interaction Matrix

The Notes Interface has its own particular style. It presents a left Navigation bar to the user with a series of buttons or icons. These change the selection into the database and the resulting objects are displayed in a definable way to the right. Each Key High level Process has its own Navigator bar and the buttons represent the Key wait positions in the process that require managing. We added a couple of extra Navigators for Administration (of the software), Reports and one which mirrors the web screens we make available to students – Opportunities. The buttons on the Navigator are presented in the order the process runs, or sometimes important wait states are highlighted or the default when the navigator is selected (e.g. open student activity reports which represent problems to be managed are first on all student screens). Emails are used for triggers which do not occur very often eg. a student coursework arrives.

3 Discussion

The approach here has used goal-oriented business processes as the domain modelling tool for requirements analysis. This differs from many approaches. The Unified Process [Jacobson 1999] includes a business modelling step, but is not prescriptive on the approach and implicitly assumes a business use case approach [Jacobson 1995] that differs considerably. Our approach is akin to the Select Perspective [Allen 1999]. Where we have differed from contemporary object-oriented approaches is in leaving out the use case step [Rosenberg 1999]. Our use cases are in fact automated (or semi-automated) process steps. Skipping the explicit use case step did not seem to cause us any particular problems, though elements such as roles are embedded in process steps and are equivalent to actors in a use case diagram.

The architecture of the target environment (Lotus Notes) did lend itself to the analysis approach. When we tried a more traditional OO approach to defining the system identifying users and use cases, we got bogged down in a lot of detail and complexity. At this stage, we cannot be too assertive about whether the business modelling approach is superior, or whether it just fit our particular application domain and target system architecture.

The system has been remarkably successful, and meets the goals we set it. Its success lies, we believe, in it being based on a solid understanding of the business processes, and the goals of those business processes. We also were systematic in fitting the functionality of the support system to the business process. Also, along with the system, comes a clear definition of best practice, as defined by the process map.

In the Prism research group at the University of Huddersfield, we are investigating a variety of approaches to requirements gathering and systems analysis. Our expertise comes from a broad background, covering soft systems methods, structured systems methods, object oriented methods and business process methods. We are building a broad picture of modelling approaches to support analysis and design. By looking at problems from different perspectives, we hope to uncover some underlying principles common to all methods.

4 References

- [Allen 1998]: , Allen, P., Frost, S., Component-Based Development for Enterprise Systems: Applying the SELECT Perspective, , ISBN0521649994
- [Bennett 1999]: Object-Oriented Systems Analysis and Design, Bennett, S., McRobb, S., Farmer, S, , 1999, ISBN0 07 709497 2
- [Jacobson 1995]: The Object Advantage, Jacobson, I., , 1995, ISBN0 201 42289 1,
- [Jacobson 1999]: The Unified Software Development Process, Jacobson, I., Booch, G., Rumbaugh, J., , 1999, ISBN0 201 57169 2
- [Rosenburg 1999]: , Rosenburg, D., Scott, K., Use Case Driven Object Modeling with UML, 1999, ISBN0201432897