

How to compose an Object-Oriented Business Process Model?

P. Kueng¹, P. Bichler², P. Kawalek¹ and M. Schrefl²

¹ *IPG, Computer Science Department, University of Manchester, Oxford Road, Manchester M13 9PL, UK. Tel: +44 161 275 6183 Fax: +44 161 275 6236 Email: {kueng, kawalek}@cs.man.ac.uk*

² *DKE, Department of Information Systems, University of Linz, Austria. Tel: +43 732 2468 9479 Fax +43 732 2468 9471 Email: {bichler, schrefl}@uni-linz.ac.at*

Abstract

Faced with the intensive business process reengineering activities in many companies, it is not surprising that the issue of process modelling has become a central concern. This paper shows that object-oriented system development methods can be applied to the field of business process modelling, but that certain steps are needed in advance. For example, it is necessary to compose a goal-means hierarchy, to establish necessary activities and roles, and to determine the input and output for each activity. In this paper, we examine step by step how business processes can be modelled, which data are needed for each step and which result would be produced during each step.

Keywords

Business process modelling, goal-means hierarchy, object-orientation

1 INTRODUCTION

Today, many organizations undertake fundamental change programmes with the aim of improving their market competitiveness. Typically, the main challenges they confront are the reduction of cycle time, decreasing overall costs, and the improvement of customer satisfaction. In pursuit of such benefits the organization may seek to adapt or design processes with the aim of simplification, better control or the ready availability of information relating to the state of extant business cases. These changes are often accompanied by increased dependency on complex and heterogeneous software systems. Against this background it is not surprising

that more and more enterprises establish new workflow systems for which they aim to prove coherent support of their business processes. Whilst the demand concerning business process related software and development methodologies has already reached a notable level, commercially available workflow management systems are only now evolving beyond a rather overly simple, Taylorist, production-line metaphor. Furthermore empirically proven methodologies for modelling and implementing business processes do not exist, cf. [Swenson/Irwin 95].

The rest of the paper is concerned with the presentation of a modelling approach. It focuses upon the notion of a goal. The hypothesis is that the modelling of behaviour (e.g. a business process) is best understood as purposeful, and can be described through goals. This work has been developed at the University of Linz and shall be progressed in collaboration with the University of Manchester. The paper is organized as follows: Section 2 gives an overview to today's business process modelling approaches. Section 3 gives a short introduction to a business process, showing how a system development life cycle could look like and how our goal-based modelling process is embedded. Section 4 shows how both enterprise-wide and business process-related goals can be modelled. In section 5, we use a case study to transfer goals into activities and explain how logical dependencies between activities can be visualised. Section 6 presents and applies the concept of roles. In section 7, our example will be transformed into an object-oriented model. Section 8 concludes with a summary and an outlook on additional issues to be addressed in our research. Overall, we believe the contribution to be the innovation of a methodological framework rather than the creation of new notations.

The case study considers an insurance company with offices throughout Europe. The investigators spoke to commercial underwriters and administrators in the London headquarters and local office. Compared to other insurance sectors (e.g. motor policies, home insurance), the commercial sector is low volume and highly labour intensive. Underwriters receive submissions from brokers which describe major risk proposals (e.g. all the factories of a multinational manufacturer). To process a single submission can be time-consuming. It is likely to involve many interactions with the broker and within the insurance company itself (e.g. between underwriters and administrators). The case study took place with the company in a phase of expansion.

2 THE STATE OF THE ART

To date several methodologies have been proposed. They can be grouped into four broad categories:

Activity-oriented approaches: As the name implies, activity-oriented approaches focus primarily upon activities (sometimes referred to as tasks). The flow of information, the involved organizational units, and data are either not considered or are understood in the context of the description of activities. Activity-oriented approaches are well suited to high level process description. At a lower level they are used for simulation, e.g. estimating of cycle time. There are many activity-oriented approaches, e.g. Information Control Nets [Ellis/Nutt 80], Trigger Modelling [Joosten 94], Event-driven Process Chains [Scheer 94]. Taking into account that the mentioned methods differ from each other, criticism can be only fragmentary:

- Activity-oriented approaches generally offer good support to the process of refinement. However this may encourage too much attention to be paid to the detailed process structure and too little to the main structure of the business process.
- Activity-oriented approaches tend to define a business process as a specific ordering of activities. This mechanistic view may fail to represent the true complexity of work, and may lead to the failure of the implementation of a new business process.

Object-oriented approaches: The principles which we associate with object orientation, for example encapsulation and specialization, may in various ways be a part of other approaches (e.g. activity-oriented approaches, role-oriented approaches). The well known object-oriented methods (e.g. [Booch 94], [Embley et al. 92]) are widely used for designing and implementing software systems. It seems obvious that the principles of object orientation be applied to business process modelling. Are the techniques, in their current form, adequate for business process modelling? Not fully, because:

- If the focus is only upon objects – describing structures and methods – the objectives of the business process may not be considered. Starting with our hypothesis about the value of understanding business through purpose, it can be implied that whilst object orientation offers well understood benefits, it does not follow that a business needs to be considered as a set of objects at all levels of abstraction. An alternative is to recognise other semantic concepts such as purpose and to map them to an object structure at an appropriate level. In this way the approach differs from some others which also give a high level view of the business, e.g. [Graham 95].
- Business processes are not designed by information systems specialists but primarily by process owners or their team members. Empirical evidence of case studies suggests that if you ask these people how a certain business process operates, they will give a description of activities, e.g. [Kawalek 95]. In other words: process owners and team members describe their work through activities rather than objects.
- Most object-oriented methodologies apply object interaction diagrams. In these diagrams we can identify the concept of roles. The weakness of this approach is that the assignment of roles is done normally as a minor matter. If roles are important to us then we need to give more attention to their assignment.

Role-oriented approaches: Probably the best known role-oriented technique is the Role Activity Diagram (often called just RADs) [Ould 95]. The origin of the technique lies with the modelling of coordination by [Holt et al. 83]. The concept of ‘role’ is obviously central and yet is rather loosely defined. Ould suggests that a role “... involves a set of activities which, taken together, carry out a particular responsibility or responsibilities” [Ould 95, p. 29]. To Halé a role is “The position played in a process by an individual, team or unit” [Halé 95, p. 237]. Given these broad definitions we can describe many things as roles, whether they be whole job descriptions (e.g. administrator), parts of work activity (e.g. make expense claim) or sub-parts of that activity (e.g. calculate expenses). It follows that roles are conceptually similar to modules. They allow a grouping of primitive activities which can then be assigned to a particular person or agent. [Kawalek 95] argues that the strengths of RADs lie with their ability to express this modularity of work through roles and the synchronisation between these roles. Essentially this means that through a role-oriented approach we are able to describe process behaviour at different levels. We can describe the co-ordination between roles and demarcate this from our concern for the co-ordination within roles. RADs are increasingly popular, especially within the UK. They seem to have many strengths but also some weaknesses, for example:

- RADs are not very suitable if it is important to express an intricate sequencing logic. For example, it can be difficult to express behaviours where two activities can be carried out alternatively. It is still more difficult a behaviour where the sequence of two or more activities is undefined except for the fact that they cannot be carried out concurrently.

Speech-act oriented approaches: Speech act theory was mainly created by Austin and his student Searle, cf. [Winograd/Flores 86, p. 58]. Further development, under the label “Language/Action Perspective” were made by Winograd, Flores and Medina-Mora, cf. [Medina-Mora et al. 92]. The underlying concept behind Language/Action Perspective is the so called “ActionWorkflow Loop”: In each communication process (workflow) we can distinguish between a customer and a performer. The communication process itself consists of a four-phased loop: proposal, agreement, performance, and satisfaction. The speech-act approach is novel, interesting and potentially very significant. From various sources (e.g. [Agostini et al. 94]) empirical examples of its use are being assembled. What are the current limitations of speech-act oriented approaches?

- Carrying out business cases is always seen as a communication between a customer and a performer. The model doesn’t take into account several parties. Furthermore it is not always obvious which part is customer and which is performer. In different business cases they can have a different behaviour.
- It is not clear whether speech-act oriented approaches are primarily dedicated for analysing existing processes or for creating new processes. In the former case, a speech-act oriented approach could help to analyse communication flow between process participants. In the case of creating new processes, this approach doesn’t provide much help: neither does it help to find adequate roles nor does it help to identify activities for supporting given goals.

As a broad conclusion concerning the aforementioned approaches, it seems that today’s business process modelling approaches are still immature. There could well be a viable synthesis of approaches in the future. Such a synthesis would have to address three things: First, it would have to look broadly at business processes and appreciate the relationship between the operational behaviour and managerial co-ordination, control, development and policy [Beer 79]. Secondly, it would need to describe the process of modelling. Thirdly, developing the previous point, they need to describe what kind of information has to be input to a methodology and its output from its application. In this way we would be able to select methods in an contingent way according to the circumstances of our modelling project.

3 KEY ELEMENTS CONCERNING BUSINESS PROCESS MODELLING

Generally speaking, a model highlights certain aspects of the real world and omits others. What does this mean with regard to the subject of business processes? According to [Davenport 93] and [Hammer/Champy 93] a *business process* can be characterised by five elements: (1) a business process has customers; (2) a business process consists of activities; (3) these activities create value for the customer; (4) activities within a business process are carried out by humans or machines; (5) business processes often involve several organizational units; that means more than one organizational units are responsible for a whole business process.

How are these aspects interrelated? As a core element of a business process model we have *business cases*, which are instances of a business process. Business cases have to be carried out. That means fulfilling *business goals* and satisfying *stakeholders*. Business cases are composed of *activities* (sometimes referred to as functions or working steps or tasks) which can be further decomposed into subactivities. Activities within an office environment need information as *input*. That input has to be provided by an *information producer*. Activities also produce an *output* which will be delivered to the *customer*, probably the most important player within a business process. Activities have to be carried out by *roles*.

Creating and implementing new business processes is a highly complex task. There are still few empirically established examples. Furthermore it is difficult to appreciate the most important requirements at the beginning of a project. It is effectively impossible to estimate if the proposed process model would lead to the desired state.

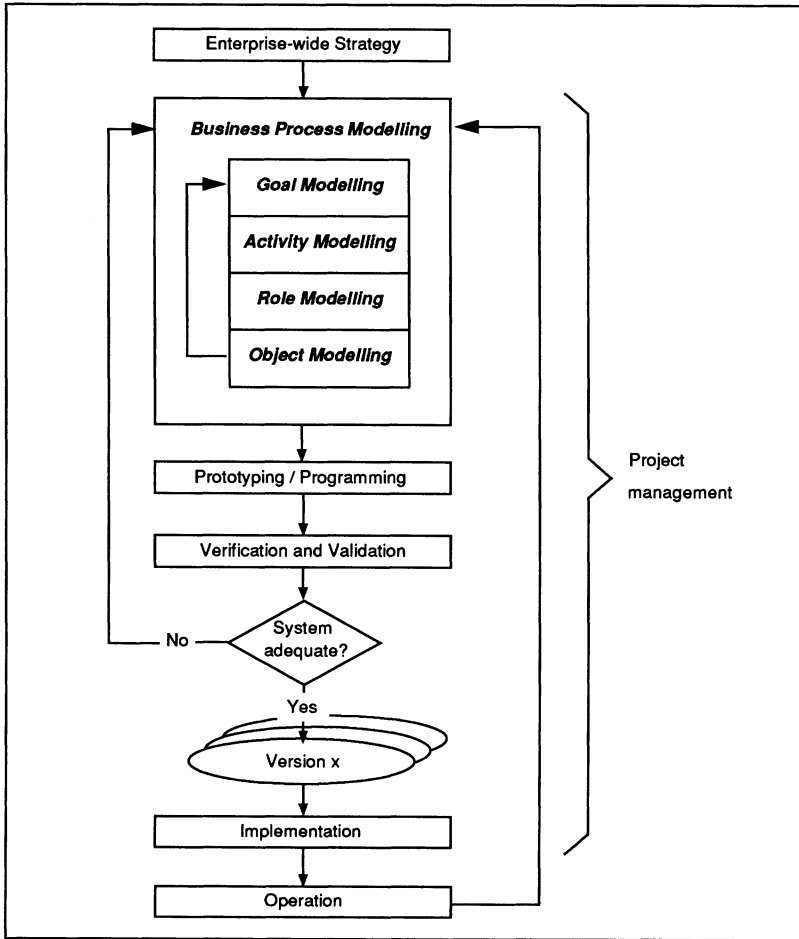


Figure 1 The system development life cycle.

In order to reduce these problems we propose to apply a cyclic stage model. Figure 1, which includes ideas from [Floyd et al. 89] and [Hammer/Champy 93], shows that the development and implementation of a business process is made up of several activities: First of all, an enterprise-wide strategy – which describes the enterprise-wide as well as the future product and

service portfolio – has to be developed. After that, the business process(es) have to be modelled; an activity we describe in detail later. Subsequently the modelled business process has to be verified and validated. Whereas verification is usually done by formal methods, validation may efficiently be carried out by prototypes. Prototyping allows potential users to judge if the system is adequate or not. After several iterations the business process model has to be transformed into an executable system. This might be done using either a workflow management system or in a traditional way, e.g. coding a C++ application. If an executable program has been created it has to be implemented in a designated environment. Generally speaking, implementation means developing a plan that addresses the organization of change. On the organizational side this may include transformations concerning hierarchy of management, incentives, performance measurement, job description, job changes, skills, and training. After successful implementation the system goes into the operation. After an unknown duration in operation, process goals are likely to change and this eventually leads to a new development cycle. Furthermore customer needs, overall policies or profitability of certain activities may also change – and as a result, business processes or part of them may be outsourced.

Figure 1 shows that the mentioned steps of a system development life cycle are surrounded by project management. It is an ongoing activity and defines some key elements which affect the success of a project much more than technical aspects. Project management includes, but is not limited to the following aspects: organizational structure of the project team, staff (quantity and qualification), maximum project duration, maximum project cost, methods and standards concerning each development step, monitoring and measurement of progress and risks, facilities and technical equipment.

Before proceeding, three preliminary remarks have to be made: (1) During the modelling process (analysis and design), we do not look at a possible implementation strategy. That means, the subsequently modelled business processes would be technology-independent. (2) Business processes discussed in this paper concern the area of office systems, e.g. informational processes. In other words, at present, manufacturing processes which have the objective to produce physical materials (tables, printers, cars, ...) are beyond our discussion. (3) We do not take into account existing IT components. In contrast to the fountain life cycle model (cf. [Graham 95, p. 350]), aspects of reuse are beyond the scope of this paper.

4 MODELLING OF GOALS

In traditional system development life cycles, the first step is typically described as “requirements analysis”. Experience of several projects has shown that the process of eliciting requirements cannot be seen as an isolated step, cf. [Floyd et al. 89]. Furthermore, traditional requirements analysis may produce lists with a vast number of items. For this reason, we propose to model not traditional requirements, but the more abstract notion of goals which have to be fulfilled. Goal modelling can be applied on an enterprise-wide level as well as on business process level. After starting the first business process modelling project, we have to create a model on the enterprise-wide level. That means, the goals on the top-level have to be captured and subsequently broken down into subgoals until they can be assigned to business processes.

Our work leads us to believe, that the following approach would be fruitful. To achieve a clear structure of our goal model, we divide goals into the following three categories:

- **Business process-related goals:** If business processes have to be modified – and this should happen in every process modelling project – this kind of goals are of primary importance. In the case study that follows, business process-related goals include: “making profit by selling insurances” and “increasing turnover”, cf. figure 2. How should process-related goals be modelled? The objective is to reduce or decompose process-related goals until they can be transformed into activities which have to be carried out within a business process. If a certain activity is a component of several business processes, this activity has to be placed – for reduction of redundancy – as part of a *support process*. It should be noted, that goals may not only be defined in the positive sense of “ensure that something happens” but also in the negative sense of “ensure that something does not happen”.
- **Information system-related goals:** This kind of goals are sometimes referred to as *requirements concerning the product*. The product would be, in our area of application, an information system, such as a workflow system. IS-related goals are usually not very process-specific and for this reason it would not be efficient to define them for each project. For the purpose of reuse, we can distinguish between enterprise-internal and enterprise-external IS-related goals; furthermore we can divide the two categories into enterprise-specific and process-specific goals. This means that IS-related goals can be defined for the whole company and then just have to be specialized (by refinement or extension) for a certain business process. Furthermore, IS-related goals cannot often be translated into business process activities during the first stages of process modelling. Nevertheless, it is important to make clear at the beginning of a project, which IS-related goals have to be considered whilst a new business process is being created and probably implemented in several information system components. Examples of IS-related goals are availability, conformity with user expectations, confidentiality, consistency, controlability, error tolerance, modifiability, reliability, response time, self-descriptiveness, and suitability for learning.
- **Project management-related goals:** These goals deal with the process of “information system development”. As we can infer from figure 1, these goals cannot be addressed to a certain development step. In other words, it is not possible to generally define on which modelling step what type of project management-related goal has to be used. What do we mean by project management-related goals? Some examples may make it clearer: standards and guidelines concerning modelling methods and implementation aspects, organizational structure of the project team, project duration, costs, monitoring and measurements of progress and risks, quantity and qualification of project staff, facilities and technical equipment.

5 MODELLING OF ACTIVITIES

As mentioned in section 2, many modelling approaches are activity-based. That means the first step would normally be “defining activities which have to be carried out whilst a business case progresses”. These approaches raise the question “How can we identify the appropriate activities?”. This question is very important but until now relatively little attention has been paid on it. How do we identify activities of a business process? There are four sources:

- **Goals and subgoals:** As [Hammer/Champy 93] mention, business processes should only include such activities which create value for the customer. In other words, activities have to make a contribution to the business process goals. Referring to our example, figure 2 shows that the subgoal “minimum of losses” leads to the activity “refuse risky submissions”.

- **Measurement of goal achievement:** As we have seen above, activities should make a contribution to the fulfilment of goals. This raises the question “How can we measure the extent to which a business goal is fulfilled?”. The degree of fulfilment of goals has to be measured by activities. In other words, for each business process goal we not only have to define the criteria by which the achievement can be measured, we additionally have to define activities for measuring the degree of fulfilment. This is shown in the example. Figure 2 shows for example that the goal “making profit” leads to the activity “record cost and return for each business case”.
- **Restrictions:** The definition of business processes is not a fully boundless task. It is to be expected that we have to take into consideration certain restrictions. They may come from the enterprise itself or from third parties. Restrictions can be divided into three categories: legal restrictions, technical restrictions, and social restrictions. For example, in the case study we only consider commercial insurance, because it is perceived to be technically different from other kinds of insurance (e.g. life insurance, motor insurance).
- **Input delivery:** Business processes may subsume activities, which do not produce value for the customer, which are not used for measuring goal fulfilment, and which are not derived from restrictions. What are they good for? Activities need – if they have to be carried out – input. This input has to be delivered by an activity which is either located within or outside of the business process. As we will see in table 1, concerning our example, we have several activities whose “goal” is to deliver input for already identified (possibly value generating) activities. For example, the activity “record customer request” delivers the input for the subsequent activities.

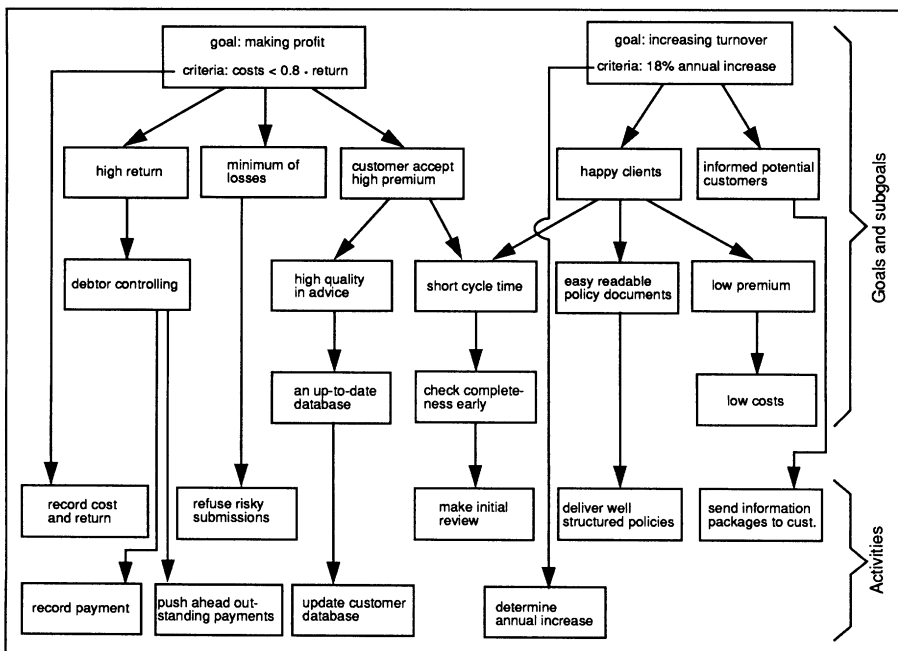


Figure 2 Goal/Activity Model for the business process “Insure customer objects”.

As we have seen above, in order to identify necessary activities of a business process, we have to define not only business goals, business restrictions, and general requirements, but also the input and output for each activity. It should be mentioned that we use input and output not only for identifying activities but also for describing the activities more precisely. In other words, we have to define what data must be available for activities to be carried out. Similarly we have to describe which data (output) has to be delivered to the process customer, to a third party, or to another business activity.

With reference to the definition of Input/Output Models (cf. table 1), it is useful to note that activities are regarded as isolated modules. For this reason any input, which is needed for carrying out an activity has to be listed. In other words, an activity cannot make any reference to a data pool, even certain data has been defined as input for another activity. The advantage of this concept is threefold: (1) activities may be placed freely, because we do not take into account the dataflow between activities; (2) if we analyse the inputs concerning their frequency and volume, we get some indications of which data are candidates for an efficient management, e.g. by database management systems; (3) the concept of isolated modules does not pre-determine any technical realisation.

What level of abstraction would be appropriate to define Input/Output Models? We take a very pragmatic position on this issue and raise the question "What kind of information can we get from people who are participating in the modelling project"? If they are thinking in categories of attributes, we would use them. If they prefer to speak about types of documents, we would record them. Much more important than the level of abstraction is the consistency. We have to guarantee that each data item (attribute or document) which is used as an input of an activity, is produced by another activity. In other words, Input/Output Models may "generate" new activities. If we compare figure 2 with table 1, we may discern that the activity "record customer request" in table 1 does not take feature in figure 2. This is because this activity has not been derived from the goal hierarchy, but for carrying out the activity "make initial review" we would use the customer request as input. Another example shows that the input of certain activity must not be produced in the discussed business process: the activity "assess risk" needs a "risk/usage matrix" as input, but as we can recognize, this "risk/usage matrix" would not be produced within our business process.

Table 1 Input/Output Model for the business process "Insure customer objects".

| Activity | Input | Output |
|----------------------------------|---|--|
| record customer request | customer request (i.e. customer name, uninsured objects and their usage) | recorded customer request (i.e. a new business case) |
| make initial review | a) recorded customer request; b) different information about customer and his objects | submission |
| assess risk | a) submission; b) risk/usage matrix | risk score for the submission |
| accept submission | a) submission; b) risk score for the submission | acceptance letter at the customer |
| refuse submission | a) submission; b) risk score for the submission | refusal letter at the customer |
| update database | a) submission; b) risk score for the submission | updated database |
| compose policy document | a) submission; b) risk score for the submission; c) terms&condition list | policy document |
| send policy document to customer | policy document | policy document at the customer |

Furthermore, looking at figure 2, it can be seen that there are some activities which are not taken into account; for example “record cost and return”, or “send information packages to customer”. Why is this the case? In order to achieve a compact business process model, activities which are either components of several business processes, or activities which are of a support character should be part of support processes. Furthermore, during the defining of the essential activities it is sometimes useful to break up an activity into two or more activities. An example is the activity “deliver well structured policy documents” which has been broken down into “compose policy document” and “send policy document to customer”. To summarize, the Input/Output Models helps to find the essential activities.

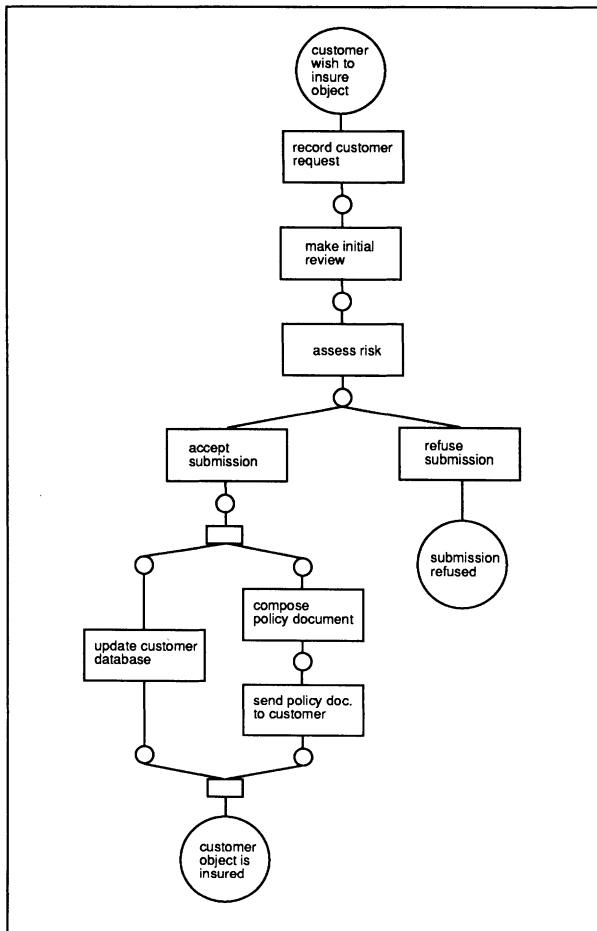


Figure 3 Dynamic Activity Model for the business process “Insure customer objects”.

Although the essential activities are defined (cf. figure 2 and table 1), we have not yet defined the logical dependencies between activities. In other words it is not yet clear which activities can be carried out sequentially, alternatively, or concurrently. To consider this aspect we apply Petri-nets; and within this category we use condition/event nets, cf. [Jensen 92]. Condition/event nets possess three strengths: (1) they facilitate a compact and precise description of the dynamic process aspects; (2) they support the concept of specialization, cf. [Kueng/Schrefl 95]; (3) they support process simulation as well.

Figure 3 shows the Dynamic Activity Model for the business process "Insure customer objects". It should be recognized that the usage of symbols in condition/event nets may vary. In these diagrams, states are shown as circles, transitions as rectangles. To make the net more compact, states are shown as very small circles and a textual description is omitted. Furthermore, transitions which have only support character (e.g. AND forks) are drawn as small rectangles. We can see that the activities "record customer request" and "make initial review" proceed sequentially whereas the activities "accept submission" and "refuse submission" are carried out alternatively. The activities "update customer database" on one side, and "compose policy document" and "send policy document to customer" on the other hand are carried out concurrently.

6 MODELLING OF ROLES

What do we mean by the term role? The Workflow Management Coalition [WfMC 94] distinguish between two kind of roles: Process Roles and Organizational Roles. The first term refers to a collection of activities. The second one refers to the functional requirements of an organizational or technical unit. Following Ellis and Wainer we use the term role in the following way: "A role is named a designator for an actor, or grouping of actors which conveniently acts as the basis for the partitioning of work skills, access control, execution control, and authority/responsibility. (...) A role may be associated with a group of actors rather than a single actor. (...) An actor is a person, program, or entity than can fulfil roles to execute, to be responsible for, or to be associated in some way with activities and procedures" [Ellis/Wainer 94, pp. 78].

The strengths of the role-concept is twofold: (1) during the modelling stage we do not have to discuss skills, functionalities, competences, and responsibilities for each activity within our business process; (2) during the operational stage people and machines (programs) with the same role are potentially interchangeable.

In some traditional approaches (e.g. Porter's value chain), activities have sometimes been misinterpreted as functions – in an organizational sense. "Experience from development and analyzes in the Norwegian TOPP study in mechanical and electrotechnical industries shows that activities are easily interpreted as functions. Misinterpretation of the term activities as functions will bring you back to the outdated organizational structure model" [Rolstadas 95, p. 153]. Without careful role assignment we may unintentionally develop hierarchical departmentalized structures and have a negative impact on the motivation of the staff. This in turn may lengthen cycle time and may decrease customer satisfaction. In other words, the assignment of activities to roles is important and has to be done very carefully – and separately from activity modelling.

Before activities can be allocated to roles it is necessary to decide which activities should be carried out by humans and which by machines. According to [Bailey 89, pp. 189] we can distinguish between five allocation strategies:

- *comparison allocation*: each activity has to be analysed and then compared with established human and machine performance criteria;
- *leftover allocation*: as many activities as possible are allocated to a machine and the activities left over are done by humans. Bailey remarks, that this strategy would probably be the most popular;
- *economic strategy*: the decision, man versus machine, based completely on financial assessment;
- *humanized task approach*: the main goal of this approach is to design meaningful human jobs/ human roles;
- *flexible allocation*: humans allocate activities in the system based on their values, needs, and interests.

Due to space limitation we resist a further discussion of these allocation strategies. Nevertheless, it seems reasonable to propose that the humanized task approach and the flexible allocation would be the two most appropriate strategies.

How should activities be assigned to roles? Although this question has been considered by many researchers, it has not been answered properly. Here we make just a general remark: instead of many particular specialized workplaces we should create “self-contained units”. We achieve this by adherence to the principles of decoupling and cohesion which govern good modular design. This helps to reduce the need for coordination and scheduling work. Furthermore, it would possibly improve lateral relations crossing the divisions’ borders, cf. [Rolstadas 95, p. 156].

After this short theoretical background we turn to the roles in our case study. We show how we graphically assign them to activities. The roles we are dealing with are the following: “administrator”, “assistant”, “customer” (policy holder), and “underwriter”, cf. figure 4. In our small example, every activity is assigned to humans. Of course, this does not always happen, but during modelling on a relatively high level this is the normal case. In other words: although we do not have any assignments to machines this doesn’t imply humans wouldn’t be supported by machines; it indicates that activities have to be split up into subactivities for their further assignment to machines.

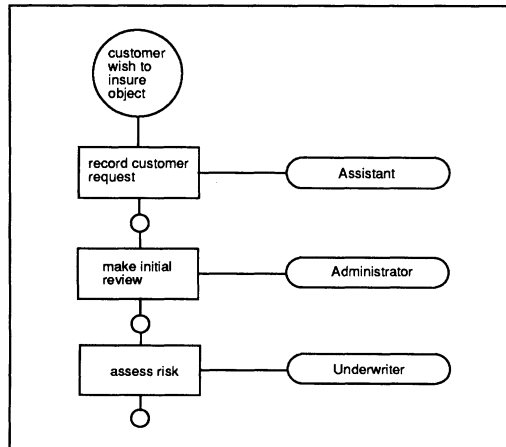


Figure 4 Role Activity Model (partly) for the business process “Insure customer objects”.

7 OBJECT MODELLING

In the previous part of the paper we showed how to model business processes on a conceptual level. This section explains how a conceptual model can be transformed into a pre-implementation, object-oriented model. Object-oriented models are constructed out of – as the name makes clear – objects. The two core elements of objects are: (1) they have a structure, and (2) they have a behaviour. While the structure is normally described by attributes and relationships to other objects, the behaviour of objects is defined by the methods objects can carry out. In other words, objects can be described by Object Relationship Models and Object Behaviour Models. Furthermore, in our universe of discourse, we have to take into account that certain objects can interact (in a certain state) with other objects. To depict this information, we need a third type of model: an Object Interaction Model.

In the phase Object Modelling (cf. figure 1) we have to answer the question “Which object classes should our model subsume and how do these objects interact?”. To answer this we distinguish three object classes:

- **Business case classes:** Objects of these class describe and control the sequence of events. Their attributes describe the actual states of the running business cases, and they define the relationships between a certain business case and the associated input-output classes. In other words, business case classes define the characteristics of business processes.¹ How do we identify business case classes? It is simple: each business process has one business case class. The name of this business case class would be identical to the name of the business process itself. In our case study, the business class would be called “Insuring objects”, cf. figure 5. As instances we have business cases, e.g. “business case 29”.
- **Input/output classes:** Objects of these classes are passive, i.e. they can not initiate an action or a communication to other objects. Objects of input/output classes are identified by looking at the Input/Output Model, cf. table 1. It gives relevant information concerning data (objects) which has to be available for carrying out activities within the business process. Furthermore, the Input/Output Model shows which data has to be produced – for the process customer or a subsequent activity. In our case study, we can identify as input/output classes e.g. “Customer”, “Customer Request”, “Customer Objects”, and “Submission”, cf. figure 5. An instance of the class “Submission” could be “submission 84371”.
- **Role classes:** Objects of these classes are roles (cf. figure 4) – which carry out activities. These objects can send messages to every other object (to passive objects as well as to active objects). Therefore objects of role classes are referred to as active objects.² In our example, we have four role classes: “Customer”, “Administrator”, “Assistant”, and “Underwriter”. For having a compact example of an Object Relationship Model (figure 5) we consider only the first two mentioned roles. An instance of the role-class “Administrator” is e.g. “Mrs. Smithfield”.

To show the interaction between the captured objects (figure 5) we use Object Interaction Models, cf. figure 6. Reading this diagram, in natural language, we would say: if a customer places a request for insuring his objects at the insurance company, the request has to be

1. It has to be remarked, that the main characteristics of a business process could also be defined – as attributes and methods – in the “normal” classes. The advantage of creating a separate object class is twofold: (1) the structure of a business process would be easier to understand, and (2) the concept of inheritance would be applicable in a more extensive way, cf. [Müller-Luschnat et al. 93].

2. The concept of roles – in the context of Office Systems – has been introduced by [Lyngbaek/McLeod 84].

recorded. As an object-oriented expression, we would say: the object Customer sends a message to the object Customer Request.

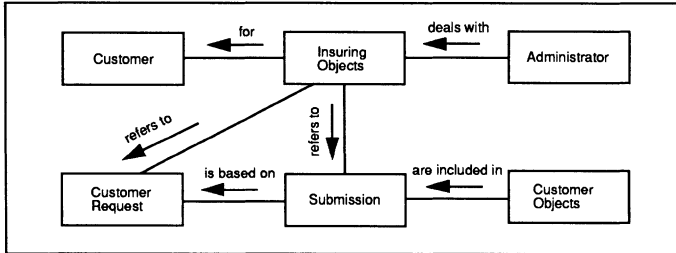


Figure 5 Object Relationship Model (partly); notation according to [Embley et al. 92].

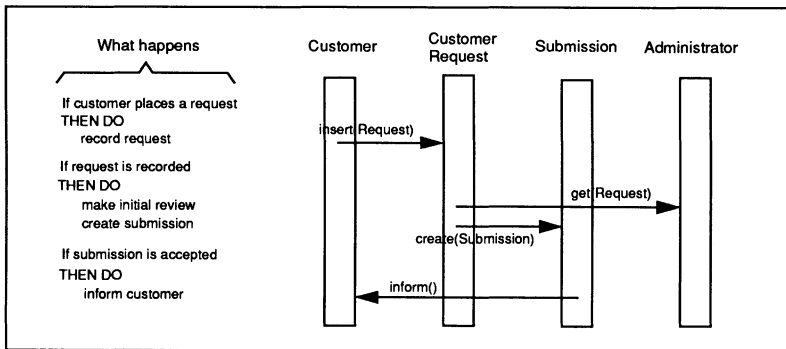


Figure 6 Object Interaction Model (partly); notation following to [Jacobson et al. 94].

It would be possible to develop this model further. For this example we have refrained from doing so as the reader will, in any case, be able to reference the huge volume of literature about object-oriented systems development.

8 SUMMARY AND FUTURE WORK

In an influential paper [Curtis et al. 92] suggested there are four important perspectives to process models. These are the functional, the behavioural, the organizational, and the informational perspective. They provide a useful framework for considering the coverage of the approach presented in this paper. The functional perspective of a business process is given by Object Interaction Models, cf. figure 6. The behavioural perspective of a business process (e.g. activity sequencing) is represented on two levels: at the pre-object-oriented level it is represented by Dynamic Activity Models (Petri-nets, cf. figure 3) whilst at the object-oriented level

it is represented by Object Behaviour Diagrams. The organizational perspective of a business process (which shows by whom are activities carried out) is represented by Role Activity Models, cf. figure 4. The informational perspective of a business process (e.g. the entities produced) is represented by Object Relationship Models, cf. figure 5.

What are the strengths of the goal-based modelling approach which is presented here?

- The methods of the objects (activities) are derived from business goals. In other words, if business goals change, we can easily establish which methods have to be updated.
- Activities are used as modules because all of them have their own input-output interfaces. The advantage of this is twofold. First, we can arrange our activities (at least in the first development cycle) within a business process in a way that is relatively free from restriction. Secondly, for every activity we can decide to perform it in-house or to buy it from a third party.

Where are the limits of our approach?

- The goal-oriented approach does not provide any help for appraising design alternatives.
- In order to realise successful business processes, we not only have to have a mature business process model, we also have to implement it successfully. In other words, following the proposed steps cannot guarantee efficient business processes.

This paper has given an overview of an approach to modelling business processes. We have showed which steps would be needed to create an object-oriented business process model, how these steps can be carried out, and how the main ideas are applied in a case study. Some answers have been given, others need further research investigation.

A prominent issue is the development of the goal model. It is intended to extend the goal model in order to support the description of richer goal structures. It is intended to do this with more enhanced features, such as conjunctive, disjunctive, and conflicting subgoals. These will not only allow to identify the activities needed to achieve them, but will also allow to infer restrictions on their logical order. For example, if a goal is decomposed into two conjunctive subgoals, the activities associated with these subgoals must be either executed in sequence or in parallel, but may not be executed alternatively.

To conclude, it is useful to list other questions which need further research. These include the following: What organizational and technical means do we have to depict individual/social/personal/organizational goals? How could user participation, during business process modelling, be augmented? How can we support the process of finding the appropriate activities? How can we establish the appropriate role for each activity? How would business process modelling be influenced by implementation aspects (WFMS versus conventional programming)? How can we measure the quality of a business process model in each modelling step?

Acknowledgement

The work of Peter Kueng has been supported by the Swiss National Science Foundation.

9 BIBLIOGRAPHY

- [Agostini et al. 94] Agostini, A.; De Michelis, G.; Grasso, M.; Patriarca, S.: Re-engineering a business process with an innovative workflow management system – a case study. In: Collaborative Computing, Vol. 1, No. 3 (September 1994), pp. 163-190.
- [Bailey 89] Bailey, Robert: Human Performance Engineering – Using Human Factors/Ergonomics to Achieve Computer System Usability. Prentice-Hall, 2nd ed., London 1989.
- [Beer 79] Beer, Stafford: The Heart of Enterprise. John Wiley & Sons, Chichester 1979.
- [Booch 94] Booch, Grady: Object-Oriented Analysis and Design with Applications. Benjamin/Cummings, 2nd ed., Redwood City CA 1994.
- [Curtis et al. 92] Curtis, Bill; Kellner, Marc; Over, Jim: Process Modelling. In: Communication of the ACM, Vol. 35, No. 9 (September 1992), pp. 75-90.
- [Davenport 93] Davenport, Thomas: Process Innovation – Reengineering Work through Information Technology. Harvard Business School Press, Boston 1993.
- [Ellis/Nutt 80] Ellis, Clarence; Nutt, Gary: Office Information Systems and Computer Science. In: ACM Computing Surveys, Vol. 12, No. 1 (March 1980), pp. 27-60.
- [Ellis/Wainer 94] Ellis, Clarence; Wainer, Jacques: Goal-based models of collaboration. In: Collaborative Computing, Vol. 1, No. 1 (March 1994), pp. 61-86.
- [Embley et al. 92] Embley, Davis; Kurtz, Barry; Woodfield, Scott: Object-Oriented Analysis – A Model-Driven Approach. Yourdon Press, Prentice Hall, Englewood Cliffs 1992.
- [Floyd et al. 89] Floyd, Christiane; Reisin, Fanny; Schmidt, Gerhard: STEPS to Software Development with Users. In: Ghezzi, C.; McDermid, J. (Eds.): Proceedings, 2nd European Software Engineering Conference, ESEC '89. LNCS 387, Springer-Verlag, Berlin 1989, pp. 48-64.
- [Graham 95] Graham, Ian: Migrating to Object Technology. Addison-Wesley, Wokingham, England 1995.
- [Halé 95] Halé, Jacques: From Concepts to Capabilities – Understanding and Exploiting Change as a Competitive Advantage. John Wiley & Sons, Chichester 1995.
- [Hammer/Champy 93] Hammer, Michael; Champy, James: Reengineering the Corporation – A Manifesto for Business Revolution. Harper Business, New York 1993.
- [Holt et al. 83] Holt, Anatol; Ramsey, Rudy; Grimes, Jack: Coordinating System Technology as the Basis for a Programming Environment. In: Electrical Communication, Vol. 57, No. 4 (1983), pp. 307-314.
- [Jacobson et al. 94] Jacobson, Ivar; Christerson, Magnus; Constantine, Larry: The OOSE Method – A Use-Case-Driven Approach. In: Carmichael, Andy (Ed.): Object Development Methods. SIGS Books, New York 1994, pp. 247-270.
- [Jensen 92] Jensen, Kurt: Coloured Petri Nets – Basic Concepts, Analysis Methods and Practical Use; Volume 1. Springer-Verlag, Berlin 1992.
- [Joosten 94] Joosten, Stef: Trigger Modelling for Workflow Analysis. In: Chroust, Gerhard; Benczur, Andras (Eds.): Workflow Management – Challenges, Paradigms and Products; Conference Proceedings of CONnectivity '94, Linz, Oct. 19-21. Oldenburg Verlag, München 1994, pp. 236-247.
- [Kawalek 95] Kawalek, Peter: An introduction to a process engineering approach and a case study illustration of its utility. In: Browne, J.; O'Sullivan, D. (Eds.): Re-engineering the Enterprise; Proceedings of the IFIP TC5/WG5.7 Working Conference, Galway, April 1995. Chapman & Hall, London 1995, pp. 248-272.

- [Kueng/Schrefl 95] Kueng, Peter; Schrefl, Michael: Spezialisierung von Geschäftsprozessen am Beispiel der Bearbeitung von Kreditanträgen. In: HMD – Theorie und Praxis der Wirtschaftsinformatik, Jg. 32, Heft 185 (September 1995), S. 78-94.
- [Lyngbaek/McLeod 84] Lyngbaek, P.; McLeod, D.: Object Management in Distributed Office Information Systems. In: ACM Transactions on Office Information Systems, Vol. 2, No. 2 (1984), pp. 96-122.
- [Medina-Mora et al. 92] Medina-Mora, Raul; Winograd, Terry; Flores, Rodrigo; Flores, Fernando: The Action Workflow Approach to Workflow Management Technology. In: Proceedings of the Conference on Computer-Supported Cooperative Work, CSCW '92, Toronto, Oct. 31-Nov. 4, pp. 281-288.
- [Müller-Luschnat et al. 93] Müller-Luschnat, Günther; Hesse, Wolfgang; Heydenreich, Norman: Objektorientierte Analyse und Geschäftsvorfallsmodellierung. In: Mayr, H.; Wagner, R. (Hrsg.): Objektorientierte Methoden für Informationssysteme; Proceedings der der GI-Fachgruppe EMISA, Klagenufurt, 7.-9. Juni 1993. Springer-Verlag, Berlin 1993, S. 78-94.
- [Ould 95] Ould, Martyn: Business Processes – Modelling and Analysis for Re-engineering and Improvement. John Wiley & Sons, Chichester 1995.
- [Rolstadas 95] Rolstadas, Asbjorn (Ed.): Performance Management – A business process benchmarking approach. Chapman & Hall, London 1995.
- [Scheer 94] Scheer, August: Business Process Engineering: Reference Models for Industrial Enterprises. Springer-Verlag, 2nd ed., Berlin 1994.
- [Swenson/Irwin 95] Swenson, Keith; Irwin, Kent: Workflow Technology – Tradeoffs for Business Process Re-engineering. In: Conference on Organizational Computing Systems, COOCS '95, Aug. 13-16, Milpitas, USA. ACM Press, New York 1995, pp. 22-29.
- [WfMC 94] Glossary: A Workflow Management Coalition Specification. Authored by Workflow Management Coalition Members, Brussels 1994. (Updated information may be found on <http://www.aiai.ed.ac.uk/WfMC/>).
- [Winograd/Flores 86] Winograd, Terry; Flores, Fernando: Understandig Computers and Cognition – A New Foundation for Design. Addison-Wesley, Readings 1986.

10 BIOGRAPHY

Peter Bichler received his Dipl.-Ing. degree in computer science from Johannes Kepler University of Linz, Austria, in 1993, where he currently works on his PhD thesis. His research interests are authorization in workflow systems and active object-oriented database systems.

Peter Kawalek is a Research Associate of the University of Manchester. His research considers the use of process models as integrating frameworks. He also works as a consultant for Manchester Informatics Limited and has undertaken many process modelling projects with industrial collaborators.

Peter Kueng received his Doctorate from Fribourg University, Switzerland, in 1994. After finishing his studies in business-oriented computer science he worked at Fribourg University as well as for IBM Berne in the field of database systems. In 1995 he worked as Visiting Researcher within the Data & Knowledge Engineering research group at Linz University. Currently, he is Visiting Researcher at Manchester University.

Michael Schrefl received his Dipl.-Ing. degree and his Doctorate from Vienna University of Technology, Vienna, Austria, in 1983 and 1988 respectively. Presently, he is Professor of Information Systems at Johannes Kepler University, Linz, Austria. His research interests are in the fields of object-oriented systems and workflow management.