10

On engineering support for business process modelling and redesign

H.M. Franken, M.K. de Weger, D.A.C. Quartel, L. Ferreira Pires*

** Telematics Research Centre*
P.O. Box 589
7500 AN Enschede
The Netherlands
H.franken@trc.nl

Centre for Telematics and Information Technology
University of Twente
P.O. Box 217
7500 AE Enschede, The Netherlands
{deweger, quartel, pires}@cs.utwente.nl

Abstract

Currently, there is an enormous (research) interest in business process redesign (BPR). Several management-oriented approaches have been proposed showing how to make BPR work. However, detailed descriptions of empirical experience are few. Consistent engineering methodologies to aid and guide a BPR-practitioner are currently emerging. Often, these methodologies are claimed to be developed for business process modelling, but stem directly from information system design cultures. We consider an engineering methodology for BPR to consist of modelling concepts, their representation, computerized tools and methods, and pragmatic skills and guidelines for off-line modelling, communicating, analyzing, (re)designing business processes. The modelling concepts form the architectural basis of such an engineering methodology. Therefore, the choice, understanding and precise definition of these concepts determine the productivity and effectiveness of modelling tasks within a BPR project. The current paper contributes to engineering support for BPR. We work out general issues that play a role in the development of engineering support for BPR. Furthermore, we introduce an architectural framework for business process modelling and redesign. This framework consists of a coherent set of modelling concepts and techniques on how to use them. The framework enables the modelling of both the structural and dynamic characteristics of business processes. We illustrate its applicability by modelling a case from service industry. Moreover, the architectural framework supports abstraction and refinement techniques. The use of these techniques for a BPR trajectory are discussed.

Keywords

Business Process Redesign, modelling, architecture, framework, concepts, methods

1 INTRODUCTION

Since the early papers of Hammer (1990) and Davenport and Short (1990) there has been a growing (research) interest in business process redesign (BPR), also referred to as business process innovation (Davenport, 1993). The BPR-paradigm, defined as the radical redesign of business processes enabled by information technology (IT) in order to achieve dramatic improvements in their performance (Hammer, 1990), has been described abundantly in managerial journals (e.g. Davenport and Short, 1990; Earl, 1994; Hall et al., 1993; Hammer, 1990; Kaplan and Murdock, 1991). Several

G. Doumeingts et al. (eds.), *Modelling Techniques for Business Process Re-engineering and Benchmarking*

management concepts, principles, guidelines, checklists and step approaches have been proposed showing how to make BPR work. Still, the number of detailed descriptions of empirical experience and evidence in literature is small. Furthermore, consistent engineering methodologies to aid and guide a BPR-practitioner are still lacking (Jacobson, 1995). We consider an engineering methodology for BPR to consist of modelling concepts, their representation, computerized tools and methods, and pragmatic skills and guidelines for off-line modelling, communicating, analyzing, (re)designing business processes. Such an engineering methodology should be an integral part of the field of business process (re)design. The modelling concepts form the architectural basis of such an engineering methodology. Therefore, the choice, understanding and precise definition of these concepts determine the productivity and effectiveness of modelling tasks within a BPR project. The current paper contributes to engineering support for BPR by the introduction of a coherent set of modelling concepts and techniques on how to use them. We refer to the latter as an architectural framework for business process modelling and redesign.

Innovation within manufacturing and service industries is a continuous prerequisite to compete with international competition. Information Technology (IT) has been identified as an important innovation enabler. Within office environments, IT was mainly used to automate existing routine business functions and tasks during the 1970's. Innovation projects in the 1980's were characterized by quality and efficiency operations, automating entire departments through the integration of IT functionality. In the beginning of the 1990's IT initiatives had a broader perspective, integrating processes at corporate level, mainly through the use of large database applications (Hollingsworth, 1993). Current IT-based innovation attemps are enabled by the dramatically increased possibilities offered by the high degree of integration of telecommunications and informatics (also referred to as telematics), enabling transparant (multimedia) information processing (access, manipulation, transfer, distribution and archiving) distributed in time and place. Moreover, the costs of information processing has decreased significantly over the years as well, making investments in the latter technology feasible.

Innovation has led to the redesign of business functions in all of the above phases. However, radically redesigning both business processes and its IT-support simultaneously has received attention only recently. A process in this context is being defined as a structured set of logically related tasks/activities performed to achieve a defined outcome (Davenport, 1993). Current economic climate, characterized by shortened product life-cycles, increased competition, high demands of customers regarding response time and diversification in product specification, demand flexible business organisations. BPR initiatives, integrating beside IT, human resource task description, data/information/knowledge and organisational structure points of view can realize redesigned business architectures that are process-oriented, customer focussed, and optimized in terms of costs, quality, effectiveness and flexibility. As stated above, IT has a strong enabling role in BPR projects. Through the above mentioned transparant information processing possibilities, the scope of a process to be redesigned can be broad. Commonly accepted guidelines on this scope are that considered processes should cross intra-organisational boundaries and that they must be customer-driven, i.e. include customer interfaces (e.g. Hammer, 1990). The latter guidelines are essentially different

from the process scope in Total Quality Management initiatives, which generally focus on seperate functions or departments (Kaplan and Murdock, 1991).

Even though great interest is expressed by organisations in BPR, they remain sceptical about it, not in the least based upon a high percentage of BPR projects that have failed (Hammer and Champy, 1993; Hall et al., 1993). Unclear strategic alignment (Venkatraman, 1994), high investments, poor quantitative measures, business continuity hazards, and legacy systems, are some of the other reasons for this position. Organisations realize that they need a clear alignment strategy and better insight in the levering effects of IT. IT must improve the effectiveness of organisations. It is too expensive to simple support existing business processes with IT, since this can only yield marginal efficiency improvements. To make IT investments profitable, organisations should persue collective instead of individual advantage, i.e. be process-minded. The latter statement holds for both within and between organisations. Organisations consider BPR a precarious venture. It is for this reason that organisations seek for methods to off-line analyze the effects of BPR (see e.g. Baum, 1995). Such methods can help to analyze current and envisaged business processes and determine a safe migration path. Especially the redesign of business processes which are in the critical path of an organisation should be carefully examined, preferrably using off-line techniques. Modelling is considered an appropriate off-line method to analyze the effects of BPR (Jacobson et al., 1995).

The demand for the above methods for off-line analysis of BPR effects motivated our reseach on engineering support for business process modelling and redesign. We currently investigate an architectural framework that enables the conceptual integration of IT, human resource task description, data/information/knowledge and organisational structure points of view, thus achieving an architectural blueprint of the business process. Besides modelling the structural characteristics of the business process, such as business entities and their interconnection, we also model its behavioural characteristics, such as the temporal relation between business actions (performed by the business entitities) and interactions (performed by co-operating business entitities), the functionality of these actions (e.g. products realized) and related attributes representing properties of the business process. When based on a suitable architectural basis, such conceptual models facilitate analysis of how business resources are being utilized. Moreover, modelling the behavioural characteristics of the business process gives insight in the way business is done and furthermore enables optimization of quantitative metrics such as throughput, processing time, and work load.

The remaining of the current paper is structured as follows. Section 2 discusses general issues that play a role in the development of an engineering support for BPR. Section 3 describes a case of a typical business process in the service industry. This case will be used to illustrate the architectural framework, which is presented in section 4. This framework also supports abstraction and refinement. The use of the architectural framework for a BPR trajectory is discussed. We present the major conclusions.

2 ISSUES IN ENGINEERING SUPPORT FOR BPR

In this section we discuss general issues that play a role in the development of an engineering support for BPR.

Worlds of reasoning

An engineer imagines, conceives and creates a design of a business process (or a model of an existing business process) in his mind. At some point in time the engineer will document this design such that it can be saved and remembered. The documented design allows team-based communication, analysis and improvements. Individuals can also learn how business is performed from a documented business process model. Documentation neccesitates some language to describe/represent the designs. We refer to this language as a *specification language*. Thus, a design is an abstraction of an imagined business process, whereas the specification is the representation of this abstraction. A realization of a specified business process is a real-world implementation of the imagined business process. Above, three worlds have been identified, the real world which contains actual business process, the conceptual world which contains the mental models of a designer, and the symbolic world which contains the representation of the imagined design.

Abstract concepts and their representation

A design is generally conceived as a composition of components. Each component can in turn be conceived as a composition of more elementary components. When continuing the reasoning iteratively one ends at a level of the most elementary building bricks. The latter building bricks and their allowable composition facilities form the *basic architectural concepts*. The success of a modelling technique is largely determined by the choice, correct understanding and precise definition of its basic architectural concepts. These concepts mirror elementary and common characteristics (of both the structure and dynamics) of business processes, abstracting from irrelevant detail. A poor choice of modelling concepts inhibits (creative) thinking. When a team is working on a business process model, precisely defined modelling concepts are needed to enable the unambiguous exchange of ideas. One should avoid adopting concepts from other domains, e.g. information systems engineering, without carefully analyzing (and changing) them, in order to avoid viewing a business process as a computer with a database and a program (Jacobson, 1995) which is not the way a business process is generally build and viewed on by stakeholders involved. Thus, the concepts must match the domain of interest.

The latter concepts should posses the right abstraction level and granularity. This means that they should not be too elementary so that simple business components and activitities can only be expressed by cumbersome combinations of these basic architectural concepts. On the other hand, too complex concepts do not facilitate easy distinction between different components. Moreover, the construction of composite concepts must be possible to model frequently occurring activity patterns. The basic architectural concepts should form a complete set. This set should allow to model all relevant aspects of a business process. Moreover, the concepts should be broad spectrum, i.e. be applicable at all levels of detail. This reduces the number of necessary concepts for modelling.

For the representation of the designs, a similar reasoning is followed. The design should be expressed in the way it is conceived. The ideal representation can be achieved when each basic architectural concept and necessary composite concepts are uniquely representated by one element (e.g. an icon) in the specification language. We name the unique mapping between the basic architectural concepts and its representation, which enables unambiguous interpretation of a given specification, the architectural semantics.

Mathematical basis

Additional advantage is achieved when the modelling concepts are uniquely described by a mathematical model. The mathematical foundation then uniquely captures the semantics of the modelling concepts. This foundation is an assurance for the designers that a business model can be expressed unambiguously (in mathematical terms). The mathematical model facilitates exact comparison, analysis and manipulation of specified business processes. Besides (computer) simulating the specified models, exact judgement of characteristics can be given in terms of e.g. deadlocks, feasibility and necessity of occurrences of certain business actions, performance of processes and verification of specified functionality.

Types of models

The word model is often used in very different ways. We identify at least three different types of models. The above discussed architectural concepts, consisting of building bricks and their allowable composition facilities, can be considered an architectural design model or framework. Such a design model is thought to be constructed by an architect. A design of a business process, using the latter architectural concepts, can be considered an instantiation of the latter architectural design model . These instantiations are referred to as business process model types which are constructed by business process designers. Finally, we refer to the mathematical basis of a considered architectural design model. Within the chosen mathematical domain every possible instantiation using the architectural concepts can be analysed. The latter mathematical models allow a.o. the derivation of property analysis theories. Such models and theories are constructed by mathematicians. All types of models are necessary for business process modelling. A business process designer should however only be concerned with a considered instantiation. This designer should be optimally equipped with an expressive set of architectural concepts. The mathematical basis should be facilitate computerized support so that the designer can easily analyze specified characteristics.

Complexity: abstraction and composition

Abstraction is an essential characteristic of an architectural framework to effectively model an redesign a business process. To abstract means to ignore details which one considers irrelevant. Abstraction facilitates a distinction between what (is (to be) done) and how (it is accomplished). We introduced abstraction already above when presenting architectural concepts. Related abstraction levels enable effective means to communicate designs with stakeholders at appropriate levels of detail. Higher abstraction levels are refined to lower abstraction levels by giving more details on how an abstract specification is to be realised, i.e. adding more design decisions. Given an existing business process, lower abstraction levels contain more detail of the existing business

process. Refinement and abstraction should be accompanied by a set of rules which ensure consistency with (modelling) decisions between the abstraction levels.
Decomposition techniques facilitate another means of seperation of concerns. Decomposition enables one to refine one component of a design in more detail, i.e. filling in more detail, leaving other components to be refined later or concurrently.

Tools and methodology

Modelling and (re)designing of complex business processes can only be effectively performed when supported by an integrated computer tool environment. We mention only a few tool functionalities such as (dynamic) graphical editing and syntax and semantics checking. Tools should also give insight into the characteristics of the specified business model, for example through simulation and property analysis mechanisms. The considered architectural framework for business process modelling and redesign is an integral part of an engineering methodology. We consider an engineering methodology for BPR to consist of modelling concepts, their representation, computerized tools and methods, and pragmatic skills and guidelines for modelling, communicating, analyzing, (re)designing business processes. This methodology is accompanied by a design strategy such that a design trajectory as a whole is covered. An engineering methodology thus enables designers to systematically deal with all concerns, requirements and constraints involved a in complex business process redesign task. The tool environment consistently supports this engineering methodology.

Context of BPR

Finally, when an architectural framework is to be used for BPR, it should have some additional facilities. One must be able to analyze constructed models on resource allocation and related contention problems, performance characteristics, task and responsibility assignments and so on. Also, the framework must allow comparison of models on the basis of criteria and/or metrics concerning structural and dynamic business characteristics. This requires a set of expressive attributes. Moreover, a business process can be considered as a *concurrent system* (defined as any collection of activities which can either be performed independently or interact with each other). Thus it is required that the adopted framework can express and mathematically support concurrency.

3 CASE DESCRIPTION

In this section we give a textual description of a typical business process in the service industry, the selling of an insurance policy at a front-office outlet. We will use this case to illustrate the applicability of our architectural framework.

A considered insurance firm has several geographically distributed front-offices to serve customers and one back-office to deal with administration, checking, batch processing of applications, etc. A textual description of a typical interaction of a client with a front office follows below. We name this process the *insurance-selling process*.
The client has decided to buy a certain insurance and visits a local front office of the insurance company of his choice. The counter-employee of the latter front office collects

the client's wishes and discusses with him the possibilities. Accordingly an application form is filled out. The counter-employee evaluates the form and can reach, based his authority and the available information, three distinct conclusions.
The first possible conclusion is that insurance-policy can be issued directly. The counter-employee supplies a covering note to the client and informs the client that the back-office will process the policy-request and send the definitive policy by mail. The counter employee sends a copy of the approved policy, with status approved, to the back-office via internal mail and files the original copy.
The second possibility is that the insurance can not be issued. The counter employee explains why and can together with the client search for alternatives. The discussed instantiation of the insurance-selling process ends here. If an alternative is found the insurance-selling process starts over again.
The third possibility is that the counter-employee is not sure whether the insurance can be issued. There can be several reasons for this. For example, the counter employee is not authorized to issue the insurance on the basis of the available information or cannot retrieve necessary additional information at the front-office site. In this case, the client is informed about the uncertainty and told that he will be further serviced by the back office via postal mail. The counter-employee sends a proviso form to the back-office and files the original form. The internal-mail service transports documents between the front offices and the back office.
The back-office processes the two types of documents from the front-office concerning the insurance-selling process. The approved forms are registered in the back-office information system. The definitive form is send by postal mail to the client and the front-office is notified. The forms with status uncertain are evaluated again resulting in two possible conclusions, acceptance or rejection. When accepted, the client is registered in the information system and receives a policy by postal mail. When rejected, the client is informed about this by postal mail. The reasons why and suggestions for alternatives are given included. In both cases the front-office is notified.

4 ARCHITECTURAL FRAMEWORK

In this section we present the current status of our architectural framework for business process modelling and redesign. This presentation is carried out at a conceptual level to appeal to (a designer's) intuition. We have derived the architectural framework o.a. by closely studying business process characteristics. We motivate and introduce the use of abstraction and refinement. We identify two domains of interest to concentrate on both the structure and the dynamic functioning of business processes. In both domains we introduce modelling concepts and their constructions rules. We explain how these modelling concepts mirror essential characteristics of business processes. Finally, we introduce structuring techniques for compositioning constructed models.

4.1 Abstraction and refinement

To abstract means to ignore details which are not relevant from the current point of view. *Abstraction levels* provide effective means to conceptualize and communicate modelling redesign issues with stakeholders at appropriate levels of detail. Abstraction

levels enable a phased separation between *what* (service a business process provides) and *how* (does this service is provided). We consider the highest level of abstraction, referred to as the *service-level*, the service of the considered process to its environment. Thus process-customer issues, i.e. how the considered process behaves in its environment, reported to be one of the most important drives in BPR projects (Davenport, 1993), are dealt with first. Abstraction thus allows designers to focus on some aspects of interest, while ignoring other aspects. Well-defined levels of abstraction guide the designer through a sequence of aspects to be considered. Each level of abstraction relates to a subsequent phase in a top-down design trajectory and its corresponding design goals. A high level of abstraction can be refined to a lower level of abstraction by adding more detail. These levels can also be traversed iteratively. In the current paper we do not elaborate on the exact definition of a complete set of abstraction levels and related abstraction/refinement rules, which is beyond the scope of this paper (see Ferreira Pires, 1994). Instead, we iteratively use two viewpoints to describe a business process.

A business process can be looked upon from two principally different viewpoints: external and internal. The *external viewpoint* considers the business process as a whole with externally observable behaviour. The internal structure and behaviour are not observable from the external viewpoint. The external viewpoint thus considers the business process as a black box which only shows which services are provided to its environment. This viewpoint thus belongs to customers or clients of the considered business process. The customers are only interested in *what* the business process can do for them. They generally do not care and are not interested in the internal structure and behaviour as long as the service provided to them complies with their quality demands. The description of the external viewpoint consequently provides the most abstract view on a business process.

The *internal viewpoint* considers the internal composition of coherent sub-entities and their assigned behaviours which together realize the externally observable behaviour of the business process. The internal viewpoint thus considers *how* the business process provides its services to its environment. The latter viewpoint is that of a business process engineer. The business process engineer aims at specifying and (re)designing the internal structure and behaviour of the considered business process such that its services can be provided. The internal structure and behaviour are not unique, given a unique service description. There are numerous possible internal compositions thinkable that can provide a particular business service to its environment. This is also the basic BPR paradigm.

The external and internal viewpoints can be used iteratively, resulting in a gradual shift from what to how. Given an external viewpoint of a considered business process, one can identify for example a number of key organisation units and their (interacting) behaviours which together are responsible for the business process. Each identified unit can again be considered from an external viewpoint with an internal structure and behaviour to be identified.

4.2 Domains of interest

From a designer's point of view it is traditional to identify *aspects* to discerns domains of interest. This is also considered a means of abstraction. Sowa and Zachman (1992) e.g. introduced the structure, process and data aspects to identify the domains in IS design. A similar decomposition into domains was reported a.o. by Aue and Brue (1994). We propose to identify two highly-related domains of interest and elaborate on these below. We refer to (Ferreira Pires, 1994) where we have illustrated our approach in detail:

1. the *entity domain*, in which structure of the business process is defined.
2. the *behaviour domain*, in which the functioning of the business process is defined.

The entity and behaviour domain are related to each other by *assignment*.

4.2.1 Entity domain

We illustrate part of the entity domain using the example of section 3. Additional concepts that are not used in the example are introduced with motivation for its necessity. At the chosen highest level of abstraction one can discern two distict *entities*, as illustrated in figure 1. One can identify the entities customer and insurance company. We define an entity as a logical or physical part (or actor) of the business process. Another concept introduced in figure 1 is *interaction point*. This interaction point can be interpreted as the service access point for customers to make use of the service provided by the insurance company, in this case the counter at the front office. We define and interaction point as a logical or physical location at which an interaction between two or more entities occurs.

We work out the internal structure of the insurance company considering one client and one front-office only. Figure 2 depicts internal structure of figure 1. The entities in the textual description of section 3 are visualized: a front-office, internal-mail system, information system, and back-office. The interaction between the customer and the insurance company is refined into interaction with the front office (via a counter) and interaction with the back office (via postal-mail). Each identified entity within the insurance company can be refined. Instantiations of the concepts (e.g. of entity and interaction point) must be labelled uniquely to ensure consistent assignment of behaviour.

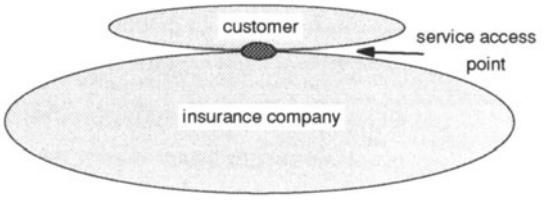

Figure 1 External viewpoint of the entity structure of the *insurance-selling process*

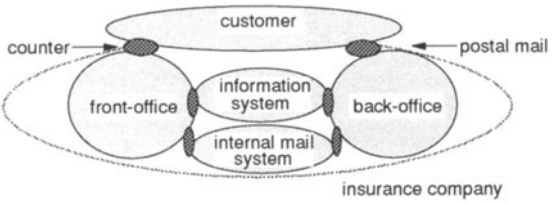

Figure 2 Internal viewpoint of the entity structure of the insurance company providing the *insurance-selling service*. We only show one front-office outlet and one customer. More outlets and customers can be identified and modelled.

Furthermore, we introduce the *action point* concept. We define an *action point as the* logical or physical location at which an action occurs. An action point lies within an entity. For example, the location in the back-office where the proviso forms are judged can be seen as an action point. An action point can be refined into an interaction point at a lower level of abstraction, whenever the corresponding action is divided among two or more co-operating sub-entities.

4.2.2 Behaviour domain

In this section we give a description of the modelling concepts in the behaviour domain, their representation and the motivation for introducing them. Furthermore, we illustrate their use in making simple business process models. A description of the formal semantics for a specification language based on these concepts is beyond the scope of the current paper.

Actions, interactions and attributes

An *action* is an abstraction of some activity in the real world. We use an action to model a unit of activity performed by a business entity. We symbolize it by a circle. An action is the most abstract model of activity at the abstraction level at which the action is defined. *What* the whole activity does can be referred to as one abstract action. A more detailed model of *how* the activity is performed can be specified in refinement steps, in terms of sub-activitities and their relationships. The relevant characteristics of these sub-activities can again be modelled by distinct actions. The action concept is independent of the abstraction level or granularity ar which specific actions are modelled, i.e. is broad spectrum. An example of an action at service-level is insurance-selling: the selling of an insurance at a front office outlet. How this abstract action is performed can be specified at a lower level of abstraction by a model consisting of multiple actions, such as apply for insurance, judge application form and accept application, and their relationships. Each instance of activity is unique and we refer to its abstraction action uniquely by an action name. Four essential characteristics of an activity are modelled as *attributes* of an action:

- *data*: the result (e.g. information and/or product) established in the action;
- *location*: defines the physical or logical location at which the action occurs;
- *time*: defines the time at which the data is established and becomes available;
- *probability*: defines the probability that an action occurs according to its definition, once enabled.

We think that the above attributes fully model the relevant characteristics an action at a certain abstraction level. For example, a succesful execution of the action insurance-selling can be characterized by the data attribute: insurance-policy, the location attribute: counter at front-office site, and the time attribute: issuing date. The probability attribute plays an important role when analyzing the performance of the latter business with multiple active instantiations of the abstract action insurance-selling. A precise description of the probability attribute is beyond the scope of the current paper. Finally, the location attribute corresponds directly to the action point identifier in the entity domain (at the same level of abstraction).

Another important concept is *interaction*, which is introduced to model that several sub-entities must co-operate to achieve a certain outcome. We symbolize an interaction by a pie-cut circle with the number of pie-parts corresponding to the number of involved sub-entities. The contribution and responsibility of each sub-entity can be distinguished. The interaction attribute values have the same type-classification as the above actions and result from a conjunction of all individual constraints. The resulting attribute values are available to all entities involved. For example, the above insurance-selling action is in fact an interaction between the customer and the insurance company. Its attributes are determined a.o. by a conjunction of wishes of the client and competences of the insurance company. The final result is available to both parties involved. Whenever one party fails the interaction will not take place. Interaction is therefore a refinement of the action concept. When abstracting from the fact that more entities are involved one ends up with a specified action. Whenever an action is refined into an interaction, the corresponding action point in the entity domain is refined into an interaction point. We motivate the introduction of interaction concept as an expressive means to model shared activity, to structure an overal business process behaviour and to allocate reponsibilities and conditions.

Causality relations and behaviour patterns

A business process was defined as a structured set of logically related tasks/activities performed to achieve a defined outcome. In other words the activities in the business are related. We model the relations by means of causality relations. A causality relation states the conditions under which an action becomes *enabled*, its enabling conditions. For example judge application form can only occur when the application is filled out properly. When the causality conditions are satisfied, the occurrence of the action is enabled. Thus, only enabled actions may take place. These actions can, in their occurrence, refer to the occurrence and resulting attributes of actions with which they have an enabling relation. We name this property *reference*.

The basic conditions are shown in figure 3. Figure 3a illustrates the *enabling relation* (a solid arrow); the occurrence of action a enables the occurrence of action b. Action b can make use of the attributes released by action a. Figure 3b illustrates the *disabling relation* (a dashed arrow); the occurrence of action a disables the occurrence of action b. Action b can only occur if action a has not occured yet and can thus not make use of the attributes released by action a.

Figure 3 a: The enabling relation; **b:** The disabling relation.

The above action attributes can be used in the definition of the relationships between actions in order to describe a business process behaviour. These attributes can also be used to define *constraints*. Constraints give restrictions or conditions on the actual contents of an attribute. For example, an insurance-policy can only be issued to a client when his credit is good. Constraints can be used as a *precondition* or logical guard for an action to become enabled or a *postcondition* for an action to happen and thus release its attributes. We do not elaborate on the use of constraints.

Composite enabling conditions can be modelled using the operators $\wedge$ (and) and $\vee$ (or) and basic enabling and disabling relation. Figure 4a illustrates that action c is enabled once action a and b have both occured. Action c can refer to the attributes of actions a and b. Figure 4b illustrates that action c is enabled by the occurrence of action a or action b. Action c can only refer to the attributes of its enabling action, either a or b, which is thus esthablished dynamically. Figure 4c illustrates that action c can only occur if action a has not occured yet and action b has occured. Action c can only refer to the attributes of action b. Figure 4d illustrates that action c can only occur if action a has not occured or action b has occured. Action c can refer to the attributes of action b in case action b enables action c. There can be no reference relation between action a and action c. Figures 4a-d model many characteristic real-life situations.

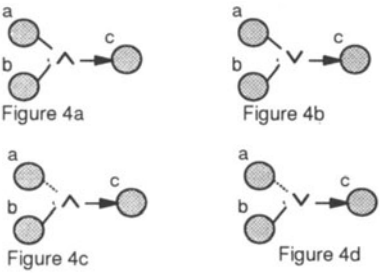

Figure 4 Graphical representation of some causality relations.

More complex behaviours can also be modelled. Figure 5 illustrates some typical behaviours patterns. They can be interpreted as explained above. For example, figure 5c illustrates e.g. that action c can only be enabled when action a has occured but action b has not occured yet. Whenever action b occurs before the occurrence of action c, action c will be disabled. Action c can thus refer to the attributes of action a only. Figure 5d introduces interleaving. This behaviour pattern is used e.g. to model the use of a shared resource. Actions b and c can thus both occur but not at the same time. We also introduce the idea of shorthand notations for often used constructions such as choice (see figure 5e and 5f).

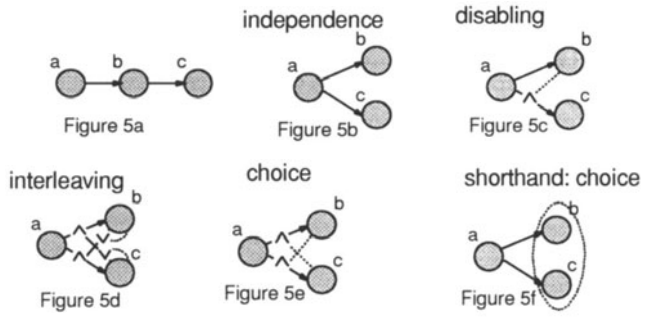

Figure 5 Graphical representations of some typical behaviours.

Both the functional and dynamic aspects of the business process can be modelled using our causality-based modelling framework. Functional aspects can be modelled with the creation or transformation of attribute values. A detailed description is beyond the scope of the current paper. Dynamic aspects follow from the relations between actions.

Case

Figure 6 illustrates the refinement of the abstract action insurance-selling. Not every sub-activity is worked out. The action *apply for insurance* results a.o. in creation of the data attribute: application form. Every activity to result in a filled-out application form is captured within the latter abstract action. This action enables the abstract action *judge application form*. Again, every activity to reach a conclusion is captured within the latter action. This action enables one of three distinct actions *reject application, accept with proviso* or *accept application*. The dashed elips (shorthand notation) indicates the choice construct, indicating that only one action may happen. Only the actions following the accept with proviso are further worked out in figure 6. Accept with proviso enables the fill in proviso form action, followed by a judge proviso form (by the back office). The latter action can enable two distinct actions reject or accept on basis of proviso form. The action accept on basis of proviso form captures every activity including the notification of customer and front office. Evidently, the description in figure 6 only shows part of the details. Attributes can be added in the specification to show what is released or to enable specification of constraints. More complicated constructions are necessary to specify/describe the insurance selling process in more detail. The above concepts and construction rules allow such a description.

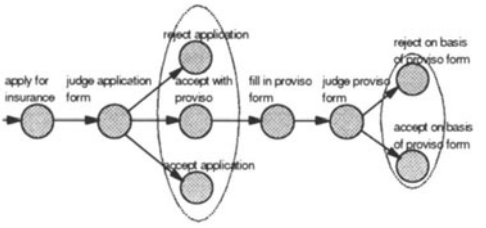

Figure 6. Behavioural aspect of the insurance selling process.

Refinement and structuring techniques

Actions can be refined. This means that one abstract action is refined into multiple logically related (less abstract) actions describing in more detail how the latter abstract action is performed. For example, the action judge application form can be refined into logically related actions such as check if form is complete, find customer file in information system, check customer characteristics, etc. Moreover, the attributes can be refined as well. When refining an action, two basic rules must be obliged: the attributes of the refinement must match the attributes of the abstract action and the context of the refinement, i.e. how it relates to its environment, must match the context of the abstract action. Refinement is accompanied by a set of consistency rules (Ferreira Pires, 1994).

Behaviours, consisting of complex graph of related actions, can be structured. Structuring techniques are useful to enable precise assignment of behaviour to (a refined) entity domain and to enable consistent application of consistency constraints from the entity domain during this assignment. We refer to (Ferreira Pires, 1994) for more detail. Figure 7 shows two possible structuring techniques. Figure 7b shows a causality-oriented decomposition of the integrated behaviour in figure 7a. This techniques decomposes the behaviour into sub-behaviours with *exits* and *entries* (graphically using >>). This can be rather helpful helpful to discern certain phases in a behaviour. In figure 7b the first phase is performed by entity E1 and the second phase by E2. Figure 7c shows a constraint-oriented decomposition of figure 7a. This technique enables the specification of *shared activity*, i.e. an *interaction*, to complete a given action. In figure 7c the integrated behaviour is performed by entities E3 and E4. Constraint-oriented decomposition enables expressive and intuitive specification of shared activities which would otherwise have to be specified in (less insightful) terms of distributed actions and causality relations. The insurance-selling process is in fact an interaction between the customer and the insurance company. Both entitities are involved in the interaction and bring along constraints. Complete description of every possible instance of selling an insurance using only actions with causality relations would probably result in a very complex descrition. Given the semantics of an interaction, we can now model this elegantly and maintain insight and control over complexity. The above structuring techniques are also accompanied by consistent rules (Ferreira Pires, 1994).

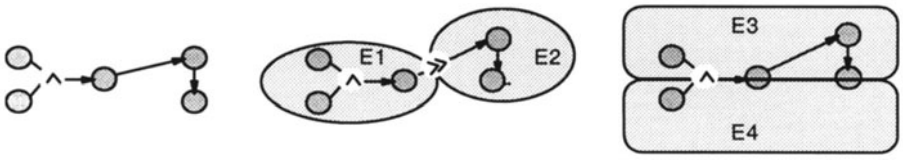

Figure 7. Decomposition of behaviour. a: integrated behaviour; b: causality-oriented behaviour decomposition; c: constraint-oriented behaviour decomposition.

Assigning behaviour to entities

We introduce *assignment* to relate the entity and behaviour. Specified behaviours are assigned to a related instantiation of the entity domain. Actions are assigned to action points and interactions to interaction points. Consistency conditions on the latter

assignment are imposed due to the chosen or existing structure and characteristics within both domains. For example, to complete an insurance-selling process, an interaction between the customer and the insurance company takes place at the specified counter. Restrictions on the type of interactions can be imposed. An example is the specification of a postal credit-loan process. Assume that interaction between customer and the credit-loan company can only take place via postal mail. This imposes serious restrictions on the interactions between the latter entities. The assignment of behaviour to the entity structure of the discussed example is displayed in figure 8. At the bottom of each action we have added the entities involved. When assigning the specified behaviour to the entity domain, all actions appear to be interactions between two or more business entities.

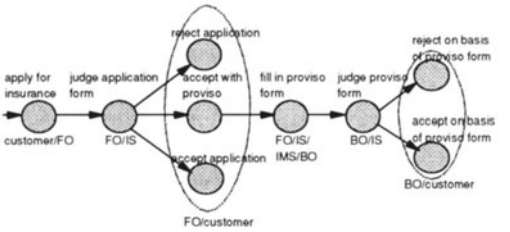

Figure 8 Assigning the behaviour of the insurance-selling process, depicted in figure 6, to the entity structure of the latter process, as depicted in figure 2. The abbreviations mean FO: front-office; IS: information-system; IMS: internal-mail system; BO: back-office.

Unique identification for model analysis

Instantiations within the behaviour and entity domain must be uniquely identified, e.g. to distinguish between multiple behaviour instantiations of a predefined structure. This can be explained as follows. The insurance-selling process is an instantiation of the discussed concepts in the behaviour and entity domain. An instantiation of the latter insurance selling process is a specification of one occurence of the latter process. Since there can be several occurences active at a given moment in time unique identification is essential, for example during simulation. This distinction is very important for performance analysis of business processes. For example, the capacity of entities in the insurance-selling process can be limited. A very simple example is that there may be only two counter-employees to help customers. The latter constraints are of interest when analysing business processes. Many other performance measures may be of interest, e.g. *business service characteristics* as customer response time, or *internal business process characteristics* as: customer-order processing time, customer-order throughput, entity utilization, and cost-ratios. The latter measures can serve as optimization criteria to *scheduling* techniques. These analysis techniques all demand a unique identification scheme for instantations.

5 DISCUSSION ON USE

The following use of the framework is suggested within an approach for BPR. Describe the internal "as is" situation in both the entity and behaviour domain. Derive the external observation for both the entity and behaviour domain. The achieved external viewpoint describes the service of the consider process to its environment. Optimize this service according to e.g. the companies strategic vision and/or the customers comments on the cuurent business process resulting in alternative specifications of the service. These alternative specifications can be compared on the basis of criteria, derived from the business goals. Optimize the internal representation of the process by refining the chosen external description. In the optimization of both the service and internal representation principles regarding the use/availability of information, the employment of technology, the organisation of work, etc. such as suggested by e.g. Hammer (1991) and Davenport and Short (1991) can be used. The latter use is illustrated in figure 9. Moreover, several services of 'to be' processes can be designed and compared. Subsequently, given a selected service, several interal 'to be' processes can be designed, analyzed and compared. This is illustrated in figure 9 by the shaded boxes. We have succesfully applied the framework according to this suggested use for several cases. Our main aim in those case-analyses was to provide insight in the current business situation, the redesign stages and suggest redesigned business architectures.

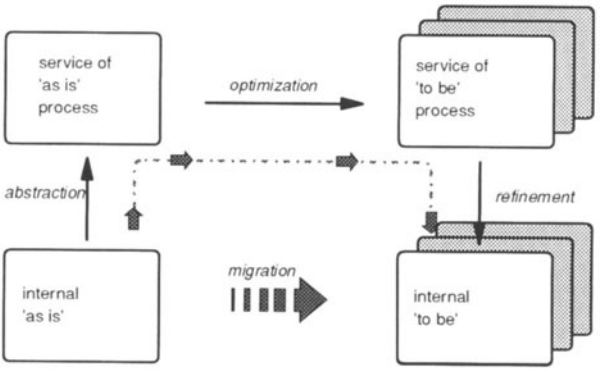

Figure 9 Conceptual use of the presented architectural framework for business process redesign. Migration from 'as is' to 'to be' is performed via the trajectory indicated.

6 CONCLUSIONS

The two described domains facilitate documentation and analysis of essential aspects of business processes. In the behaviour domain a description is given of the functioning of the business process. In the entity domain a description is given of the structural characteristics in terms of entities involved in performing the activities of the business process. The entity and behaviour domain are strongly related through assignment and corresponding consistency conditions. The entity domain facilitates a.o. analysis of the

structural organisation of the business process and the utilisation of basic resources (such as human capital, technology and data/information/knowledge). The behaviour domain facilitates analysis the organisation of work, e.g. of customer interaction, of the value adding or streamlining of actions, etc. The two domains together facilitate analysis of effective use of resources and effectiveness of operation.

The architectural design concepts or building bricks of the engineering environment, defined above as abstractions of essential characteristics of real-world systems, must suit the purpose of its application area (Jacobson et al., 1995). The latter concepts determine how models (of (re)designs) can be composed, understood, communicated, manipulated, etc. Therefore, concepts should be tailored to its application area. Similar use of concepts, referred to as information system concepts (Coad and Yourdon, 1990), ontological concepts (Wand and Weber, 1989) or design concepts (Ferreira Pires, 1994) have been suggested for IS design. In contrast to the latter use, the concepts in the current paper mirror relevant structural and behavioural characteristics of business processes. Moreover, all introduced modelling concepts are broad spectrum. They can be used at all levels of detail.

Acknowledgement: We would like to thank René Bal and Estherella Carstens for their comments on previous versions of this paper.

7 REFERENCES

Aue A. and M. Brue (1994) Distributed information systems: an advanced methodology, *IEEE Transactions on Software Engineering*, **20**, No. 8, 594-605.

Baum D. (1995) The right tools for coding business rules, *Datamation*, March, 36-38.

Coad P. and Yourdon E (1990) *Object-oriented analysis*, Prentice-Hall, Englewood Cliffs.

Davenport T.H. and Short J.E. (1990) The new industrial engineering: information technology and business process redesign, *Sloan Management Review*, **31**, No. 4, 11-27.

Davenport T.H. (1993) *Process Innovation: Reengineering work through information technology*, Harvard Business School Press, Boston.

Earl, M.J. (1994) The new and the old of business process redesign', *Journal of Strategic Information Systems*, **3**, No. 1, 5-22.

Ferreira Pires L. (1994) *Architectural notes: a framework for distributed systems development*, Ph.D. Thesis, University of Twente, The Netherlands.

Hammer M.(1990) Reengineering work: don't automate, obliterate, *Harvard Business Review*, July-August, 104-112.

Hall G.H., Rosenthal J., Wade J.(1993) How to make reengineering really work, Harvard *Business Review*, November-December, 119-131.

Hollingsworth D. (1993) Toward the 4th generation office: a study in office systems evolution, *ICL Technical Journal*, **9**, No. 4.

Jacobson I., Ericsson M. and Jacobson A. (1995) *The object Advantage, Business Process Reengineering with object technology*, ACM Books.

Kaplan R.B. and Murdock L. (1991) Core process redesign, *The Mckinsey Quarterly*, No. 2.

Sowa, J.F. and J.A. Zachmann (1992) 'Extending and formalizing the framework for information systems architecture', *IBM Systems Journal*, **31**, No 3, 590-616.

Venkatraman N. (1994) IT-enabled business transformation: from automation to business scope redefinition, *Sloan Management Review*, **33**, No. 4, 73-87.

Wand Y. and R. Weber (1989) An ontological evaluation of systems analysis and design methods'. in *Proceedings of the IFIP WG 8.1 Working Conference on Information Systems Concepts*, 79-107, IFIP, North Holland, Amsterdam.

8 BIOGRAPHY

Henry M. Franken received an M.Sc. and Ph.D. in Electrical Engineering from the University of Twente. He is currently a member of the scientific staff of the Telematics Research Centre. His current research interest focusses on telematics and business engineering. Mark de Weger and Dick Quartel both received an M.Sc. in Informatics from the University of Twente and are currently persuing a Ph.D. degree in the area of design of distributed systems. Luís Ferreira Pires received an M.Sc. degree in Electrical Engineering from the University of São Paulo (Brasil) and a Ph.D. degree in Informatics from the University of Twente. He is currently associate professor at the Centre for Telematics and Information Technology of the University of Twente, foccusing on design of distributed systems.