ReTIS'95 - Re-Technologies for Information Systems (eds. J. Györkös, M. Krisper, H.C. Mayr)
OCG Schriftenreihe (Lecture notes) Vol 80 (1995) 162-17x, R. Oldenbourg Vienna Munich

COOPERATIONS - AN ABSTRACTION CONCEPT SUITABLE FOR BUSINESS PROCESS RE-ENGINEERING

Roland Kaschek, Claudia Kohl, Heinrich C. Mayr[1]

Abstract: *This paper discusses an object oriented approach to business process modeling. The basic idea of this approach is to view business processes as being objects at their own, that incorporate the rules and conditions for a particular business of a given enterprise or organisation. A specific abstraction concept called cooperation is introduced allowing for a natural and comprehensive design of concurrent operations or interdependent interactions that are performed by different agents in the course of a specific business process. Two examples will demonstrate the approach.*

1. Introduction

The notion of business process is one of the actual buzzwords in business administration. Business processes are intended to define the structure of a companies tasks. They refer to the necessary agents and specify the execution sequence of subtasks as well as their logical interdependencies. It is assumed that analyzing the actual way an enterprise runs its business and reorganizing it on the basis of 'well formed' and well defined business processes allows for an organizational change to a leaner and more efficient enterprise and, as a consequence, to a better market position. Business process reengineering (BPR) thus becomes an key issue in strategic planning.

Management science oriented approaches mainly focus on organizational and social aspects of BPR by providing recipies for the realization of organizational changes via BPR, sometimes even called 'survival guides' [1,6,7,9].

[1] Universität Klagenfurt, Institut für Informatik, Austria

Organization theory offers various methods for the description of business processes [12]. Most of these methods distinguish between modeling organizational units with their structure at one hand and the operations they perform [3] on the other hand. Structural aspects mostly are described using ER-model oriented languages. Flow charts, data-flow diagrams or petri-nets are used to describe the dynamic aspects of business processes. Mostly, however, these submodels are not completely integrated. An example for an advanced integration is ARIS [14], but the modeling and representation concepts that are proposed by this approach do not seem to be satisfactory with respect to flexibility and reusability demands that are needed for a creative business process design. There are, e.g., no modeling concepts for handlimg versions or variants of business processes.

This paper aims at a comprehensive business process (meta) model. For that purpose we take an object oriented approach with the basic idea to view business processes as being objects at their own. Thus, a business process object incorporates the rules and conditions for a particular business of a given enterprise or organisation.

A first paper [2] introduced this idea by applying the Object Modeling Technique (OMT) [13] to an example of rather informal business processes (bill appraisal within the Austrian chambers of commerce). It was shown there that this approach leads to a concise and complete description of the structural and dynamic aspects of the processes in question. However, the dynamic model suffered to some extent from the complexity that is caused by the OMT specific use of the concept of finite state machines: Situations where several agents cooperate to perform a particular task lead to a large number of highly connected states, i.e., complex state diagrams.

To reduce this complexity, we introduce, in the present paper, an additional abstraction concept called *cooperation* and we discuss it within the context of an object oriented modeling environment. Cooperations are used as *concentrators* of business rules [10] and as building blocks (primitives) for the definition of business processes.

Section 2 introduces our meta model of business processes and cooperations, called *cooperation approach*. Within section 3 we show how to use it in detail. Section 4 gives an example showing the suitability for business process reengineering. Some conclusions are presented in section 5.

2. A meta model for business processes and cooperations

Following [8] a *process* is defined as a partially ordered set of socalled *steps* which together are intended to reach a particular system goal. A process description is composed of *cooperations* that describe how certain *contributors* (agents) have to cooperate in particular process situations (states). Several aspects of processes are discussed in [5].

A business process is a system of business rules and conditions which represent the control information for the coordination of enterprise units when operating to perform a particular service of the enterprise they belong to. It is thus quite natural to model, within an object oriented environment, business processes as objects at their own. Otherwise, the before-mentioned control information would be spread over the models of the involved enterprise units as is done in traditional methods for object oriented analysis (see [2,9,13]).

With respect to the way agents cooperate various types may be distinguished. From [4] we derive a classification into four basic types: *closed, random, open* and synchronous cooperation. We consider these to be fundamental for business processes. They differ with respect to their kinds of system regulation, priorities and the strategies of decision making which are followed by their contributors.

The *closed cooperation* assumes a hierarchy where one agent controls the others. Each position in the hierarchy has ist own well defined roles. This kind of cooperation is best applied in routine projects.

The *random cooperation* considers the freedom of the individual to be more important than group interests and is best applied for creative invention.

The *open cooperation* is based on an adaptive collaboration, integrating innovation with stability and individual interests with collective ones through negotiation. It is best suited for adaptive, collaborative projects.

The *synchronous cooperation* is based on an effortless coordination through the alignment of contributors with a common vision that reflects the common goal.

These four cooperation types differ with respect to various aspects of communication and structure. From the definition of business processes it is clear that they are the natural representants of the closed portions of the actions within an enterprise.

Our model is guided by viewing business processes as is exemplified in figure 1: A certain enterprise, here an university institute, among others offers a service (an operation, a method) *run exam* to its environment. The institute comprises the agents professor, secretary and assistant. These cooperate following the rules of a business process *exam organization* (they are involved in or they drive that process) which carries the closed portion of realization and control information for performing the institute's service *run exam*. Note that in general each service of an object may be mapped to an object internal business process on the next deaggregation level of this object (if there is one). Following that paradigm a business process instance is invoked when the corresponding service is requested (i.e. the respective message is received).

Once running, a business process instance shows a certain behaviour. This is described by means of socalled phases, e.g., *preparation*, *execution* and *termination.*Their causal and temporal dependencies are specified using a finite state machine. Modeling these phases as states as is done in figure 1, our approach so far is completely compatible with OMT.

Let us now have a closer look on the phase preparation that is specified by the cooperation *prepare_exam*. Several contributors, namely assistant, secretary and professor are involved in that phase/cooperation. This is expressed by *contributions* to the cooperation that, as parts of the cooperation, relate to services of the contributors. Firstly, the contributions *design* and *prepare* are started. They respectively are contributed to by assistant and secretary. This means that, when a preparation instance is active (running), they execute the resp. operations, i.e., the secretary reserves an examination room and announces the examination date while the assistant invents the problems to be posed to the students. After the design is completed a *check* contribution is activated and because of this the professor checks the design. Possibly there are iterations of check and *redesign* of the exam until the design is checked OK and the phase preparation of exam organization is finished.

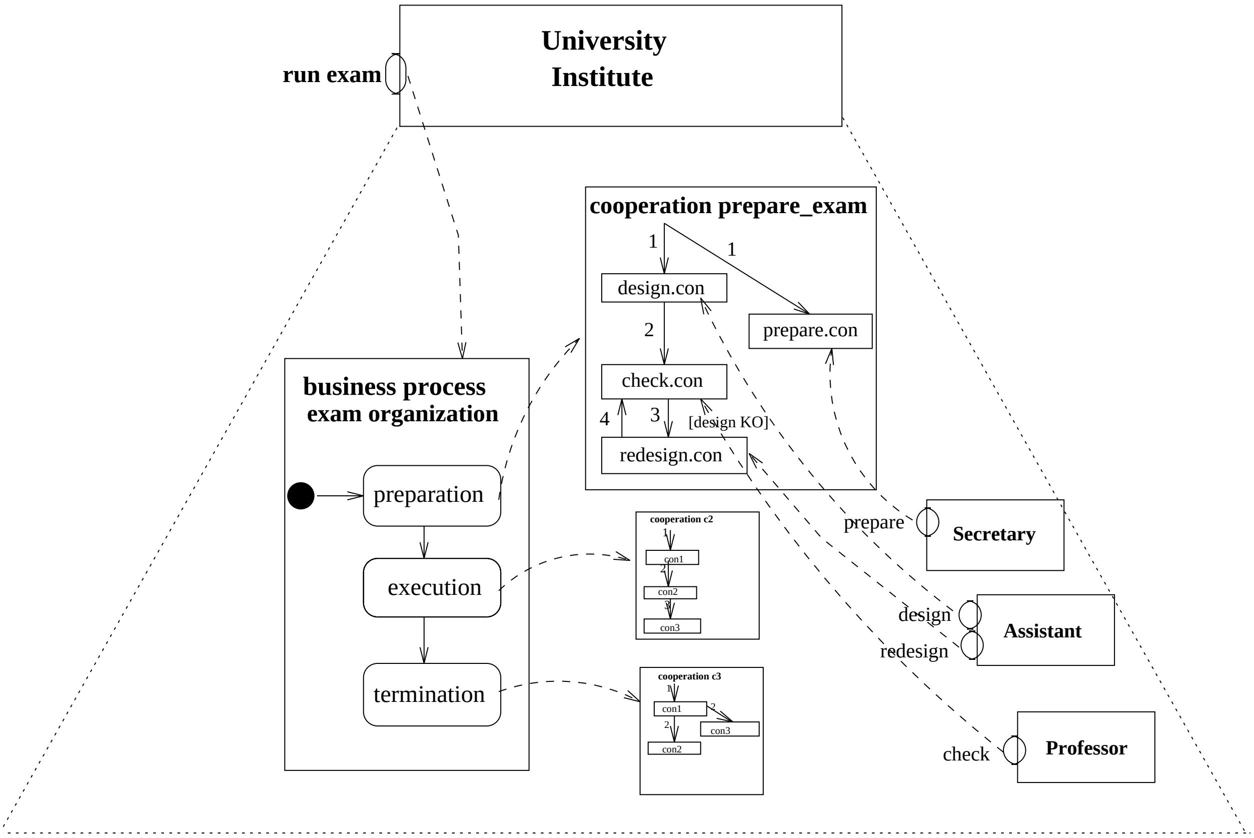


Figure 1: The business process *exam organization*

The central idea of this approach is to encapsulate those business process aspects we could not clearly describe with OMT in the middle between the interface of an object offering a business process and the objects serving as actors really doing the work. Cooperations are objectified associations wherein certain objects play the role to contribute certain of their operations. Clearly this role can be played several times within one cooperation. A meta model formalizing this ideas is shown in figure 2.

There are two main reasons for not using finite state machines (FSM) to describe the dynamics within a phase: Firstly, various objects contribute. With OMT FSMs we would have to define specific aggregations of contributor operation executions the dynamic of which includes the dynamic of the cooperation. But these schemas would tend to become too complex. Secondly, the contributors often contribute concurrently. FSMs are not really convenient for that purpose.

Essential to our paper is the left half of figure 2 while the right half shows the most essential modeling concepts of OMT. Business processes consist of *phases* and may be generalized or specialized. Specialization means addition of new phases or redefinition of existing ones. Generalization means simplification through omitting unnecessary phases. Business processes describe the implementation of operations (services) which are provided for by objects (see also [2]).

Phases are states and may be used for modeling the behaviour of business process instances. The change of states is desribed by *transitions,* which are a connections between states. Transitions are defined as consisting of *events* (‘evoking’ the transition), *actions* (that take place during the transition) and *conditions* (guarding the modification of the current state). Phases in turn consist of *cooperations*. These describe which operations objects contribute and in which order this is done.

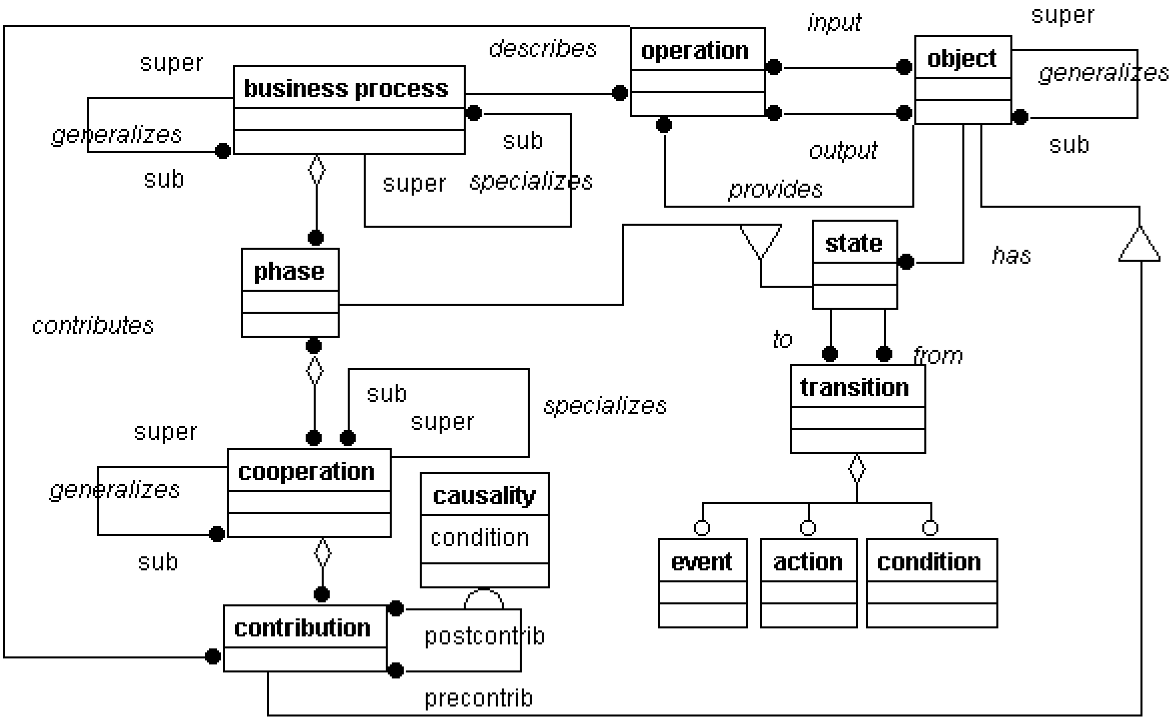


Figure 2: Meta model of business processes

Cooperations are aggregations of contributions. Specialization and Generalization here are understood as with business processes.

A contribution is the execution permission an object submits to a cooperation with respect to one of its operations. Among the contributions within a cooperation there exists a causality order having an attribute condition, which denotes a logical restriction to the causality, i.e., validates or invalidates the specified causality.

Contributions are defined to be objects. We assume that aggregates of objects again are objects. Therefore cooperations, phases and business processes are objects.

Business processes, phases, cooperations and contributions are intended to be instantiated to do the real work. We assume business process instances and cooperation instances to offer certain operations. The semantics of these operations is defined in section 3.

3. Business process modeling using the cooperation approach

For to use the cooperation approach we suppose that a busines process modeler uses an object oriented modeling environment, which, by default, offers the class *object* and the basic classes *business process*, *phase, cooperation* and *contribution*. These are shown in figure 3. We assume these classes to have operations like *visualize* and *animate* to show the structure of business processes and its components. Further we assume that statistic focussed queries may be run against business processes or their components rendering information on their structure.

We further assume the classes to be containers holding the objects compatible to the class. The set of objects in this container is called the extent of the class. Modeling now is intended to be done by creating and instantiating subclasses of the basic classes and by deriving super- or subclasses of these. The subclass relationship is thought to imply the inclusion of the subclass extent in the super class extent. The newly defined classes only may be related to each other in accordance to the meta model described in section 2.

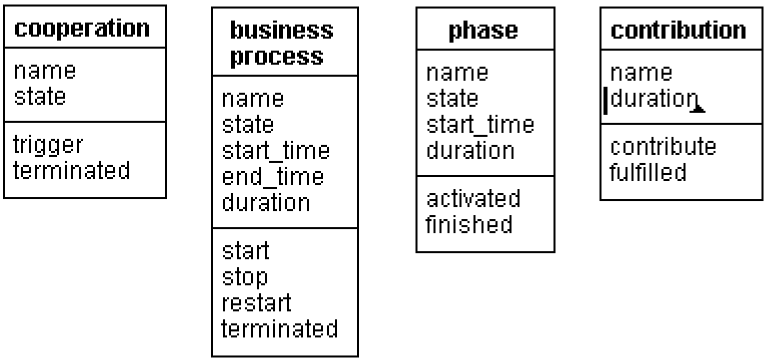
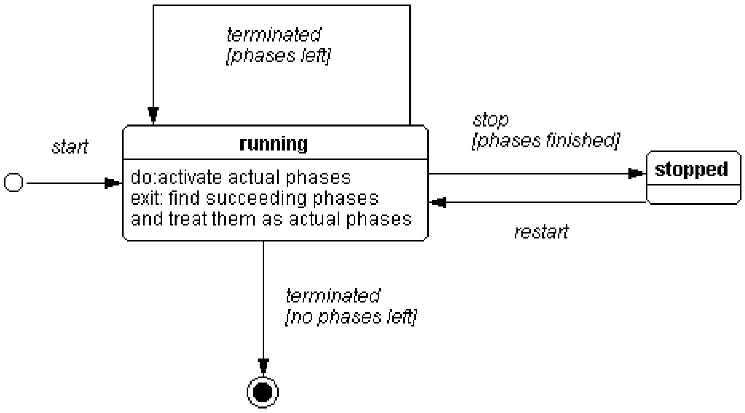


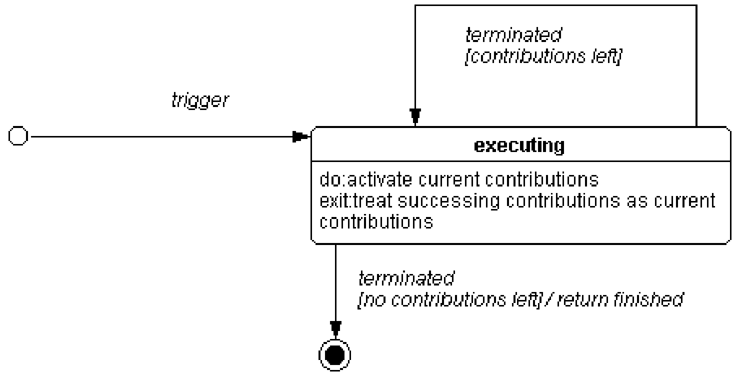
Figure 3: Basic classes of the object oriented business process modeling environmenzt

In what follows we describe the dynamics of instances of the basic classes (see figure 4).

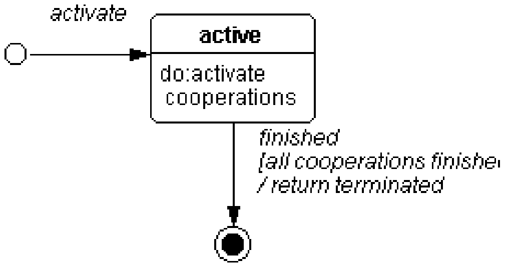
Instances of **subclasses of business process** have state *running* and *stopped* as well as an initial and a final state (see fig. 4.1). Receiving the *start* event in the initial state the instance enters the running state. Then the instance determines and activates the actual phases. The completion of a phase is signalized by the event *phase terminated.* Receiving this event the business process instance determines the set of succeeding phases as an exit activity. In case the set is empty the instance reaches its final state. Otherwhise it reenters the state running activating the succeeding phases. During running the occurrence of the event *stop* drives the instance to enter the state *stopped* after the activated phases have been completed. The occurrence of a *restart* event in the state stopped drives the instance to enter the state running again.

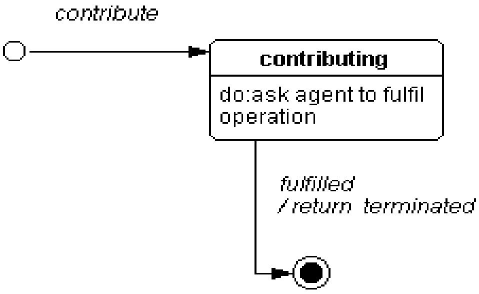


4. 1 Business process instance dynamics


terminated
[contributions left]
trigger
executing
do:activate current contributions
exit:treat successing contributions as current contributions
terminated
[no contributions left] / return finished

4. 2 Cooperation instance dynamics





4. 3 Phase instance dynamics

4. 4 Contribution instance dynamics

Figure 4: Dynamics definitions of business process components

Instances of **subclasses of phase** have the state *active* as well as an initial and a final state (see fig. 4.3). Beeing in the initial state the state active will be entered in case the event *activate* occurs. During this state all cooperation instances within the phase instance are triggered. In case all cooperations have terminated the phase instance reaches the final state.

Instances of **subclasses of cooperation** have the state *executing* as well as an *initial* and a *final* state (see fig 4.2). Receiving the *trigger* event in the initial state the instance enters the executing state. Then the instance determines and activates the current contributions. The completion of a contribution is signalized by the event *terminated.* Receiving this event the cooperation instance determines the set of succeeding contributions as an exit activity. In case the set is empty the instance reaches the final state. Otherwhise it reenters the state executing activating the determined contributions.

Instances of **subclasses of contribution** have the state *contributing* as well as an *initial* and a *final* state (see fig. 4.4). Receiving the *contribute* event in the initial state the instance enters the state contributing. Entering this state the very operation is executed which is contributed to by an object.

4. Applicability of the coooperation approach to BPR

Within section 2 we demonstrated that our approach is suitable for to define business processes from scratch. In this section we show the applicability of the cooperation approach to business process reengineering by continuing the example discussed in section 2. Our goal is to describe a modification of the business process exam organization.

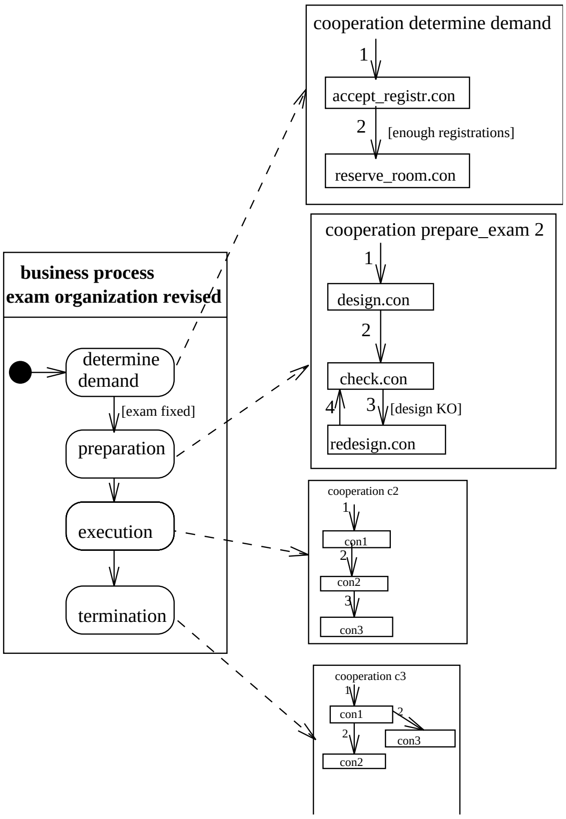


figure 5: the business process exam organization revised

After gathering some experience with the business process *exam organization* the institute notices the following problem: A lot of work is done to run the exams but in several cases no students wish to take part. Therefore one tries to find a better organization of the process reflecting the students registrations. The modified process called *exam organization revised* is shown in figure 5. This new business process is a specialization of exam organization. The structure of the old process is enriched and one of its components is redefined.

For to cope with the new goal the phase preparation is identified as to be modified. A part of this phase is reused and a new phase *determine demand* is introduced. During this phase first the contribution *accept_registration* is activated and the secretary performs her service *accept registration*. In case that enough registrations were collected the contribution *room_reservation* is activated which in turn leads to the secretatry invoking her respective operation.

The new version of the phase preparation also is called preparation but the cooperation taking place within this revised phase is modifyed and called *prepare_exam 2*. This new cooperation is a generalization of prepare exam. It omits the secretary part prepare exam.

5. Conclusions

Our examples have shown that object orientation with its abstraction concepts generalization and aggregation together with the communication metaphor of message passing is well suited to model business processes as cooperations of organisational units. A high degree of modeling flexibility and reusability of design activities could be established bearing a number of significant advantages for business process modeling and reengineering:

- The amount of work necessary to derive a new business process can be reduced by using generalization and specialization.
- The maintainance of business processes is simplyfied because changes made at a high level business process can be inherited automatically by more specialized business processes.
- By arranging the alternative business processes in a generalization-hierarchy the finding, combining, and generating relevant alternatives is greatly simplyfied.
- Versions of business processes may be built by exchanging operations.
- The complexity of modeling an enterprise can be handled through organizing it in a system of cooperating objects, offering only operations relevant at this level of abstraction and hiding not relevant details.

It finally seems that our approach provides for means that might be used for controlling the execution of business processes, e.g., in a workflow system [11]. This has to be elobarated in further research work as well as an implementation of the modeling environment.

6. Literature

[1] ANDREWS,C; STALICK,S.; Business Reengineering - the survival guide; Yourdon Press 1994;

[2] BAUER,M., KOHL; C, MAYR, H.C.; WASSERMANN, J; Enterprise Modeling Using OOA Techniques in: Workflow Management - Challenges, Paradigms and Products. Conference Proceedings of CONnectivity 94', Linz, October 1994, Oldenbourg;

[3] BUSSLER,C.; JABLONSKY,S.; An Approach to Integrate Workflow Modeling and Organization Modeling in an Enterprise; in: 3rd Workshop on Enabling Technologies: infrastructure for Collaborating Enterprises; IEEE 1994;

[4] CONSTANTINE,L.; LOCKWOOD, L.; Orchestrating Project Organization and Management; in: Comm. of the ACM; Oct 1993/Vol.36, No.10;

[5] CURTIS,B; KELLNER,M.; OVER,J.; Process Modeling in: Communications of the ACM, Sept.92,Vol 35, Nr. 9;

[6] DAVENPORT,T.; Process Innovation:Reengineering Work through Infomation Technology. Cambridge: Harvard Business School Press, 1993;

[7] HAMMER,M; CHAMPY,J.; Reengineering the Cooperation - A Manifesto for Business Revolution; Harper Business 1993;

[8] HUMPHREY,W.S.; FEILER,P.H.; Software process development and enactment: concepts and definitions. Techn. Rep. SEI-92-TR-4. Pittsburgh: Software Engineering Institute, Carnegie Mellon University;

[9] JACOBSON,I.: BRICSSON,M; JACOBSON,A; The Object Advantage - Business Process Reengineering with Object Technology ;Addison Wessley 1994;

[10] HERBST,H.; MYRACH,T.; A Repository System for Business Rules; Working Paper of the Institute of Information Systems, University Bern, Jan. 1995;

[11] KOCK,T.; REHAEUSER,J.; KRCMAR,H.; Ein Vergleich ausgewählter Workflow Systeme; in: Information Management 1/95; p 36-43;

[12] LEHNER, F. et al.; Organisationslehre für Wirtschaftsinformatiker; Hanser 1993;

[13] RUMBAUGH,J.; BLAHA,M.; PREMERLANI,W.; EDDY,F.; LORENSEN,W.; Object Oriented Modeling and Design, Prentice Hall 1991;

[14] SCHEER,A.W.; Wirtschaftsinformatik, Springer, Berlin 1994.