

MODELLING IN THE RE-ENGINEERING PROCESS⁺⁺

Aphrodite Tsalgatidou, Stefan Junginger⁽¹⁾

Dept. of Informatics, University of Athens,
Panepistimiopolis, TYPA Buildings, Athens, 157 71, Greece
email: afrodite@di.uoa.gr

⁽¹⁾Inst. of Applied Computer Science and Information Systems , Dept. of Knowledge Engineering,
University of Vienna, Brünner Strasse 72, A-1210 Vienna, Austria
email: sjung@dke.univie.ac.at

1. Introduction

Terms like Business Process Re-engineering and Business Re-engineering have their origins in the publications of Hammer (Hammer, 1990) and Davenport (Davenport & Short, 1990). As it is stated in (Soles, 1994), one of the key ideas of Re-engineering is to ‘focus analysis on business processes across the firm with customer satisfaction in view’. We define a business process as ‘*a set of logically interrelated activities within an organisation the execution of which contributes in the achievement of the business objectives*’.

In Re-engineering the analysis of business processes is combined with advanced information systems technology (IT). In this way, IT acts as an enabler for realizing new products which satisfy the users’ needs along the lines of business process analysis results. Workflow Management Systems are an example of systems that can be used as tools for implementing and executing business processes (White & Fischer, 1994).

Re-engineering can start in an enterprise only if the enterprise’s strategic management has realised that there is a need for change and for reformulation of the enterprise’s vision. Usually the enterprise’s strategic management selects the business process(es) that should be re-engineered. Re-engineering corresponds to the second

⁺⁺ *Appeared in ACM SIGOIS Bulletin, Special Issue: Business Process Reengineering, August 1995, Vol. 16, No. 1, pp. 17-24.*

process within the Business Process Management Systems (BPMS) - Framework (Karagiannis, 1994, Sep.) which considers all the activities, that have be executed while dealing with business processes, as processes themselves.

The Re-engineering Process may be seen as consisting of a number of subprocesses (Karagiannis, 1994, May; Soles, 1994). Here, the Re-engineering Process is viewed as consisting of four main subprocesses (see Fig. 1):

- *Goal Definition*, where the re-engineering goals are defined.
- *Information Acquisition*, where information necessary for business process modelling, (like activities, people, roles, control etc.) as well as information needed for the Re-engineering Process (like cost, time, specific laws etc.) is collected.
- *Modelling* which deals with the modelling of the (new) business process.
- *Evaluation* which is concerned with the evaluation of the business process model against the re-engineering goals.

Depending on the evaluation results, the subprocesses of *Information Acquisition*, *Modelling* and *Evaluation* may be iteratively executed until all the re-engineering goals (defined during the *Goal Definition* subprocess), have been reached.

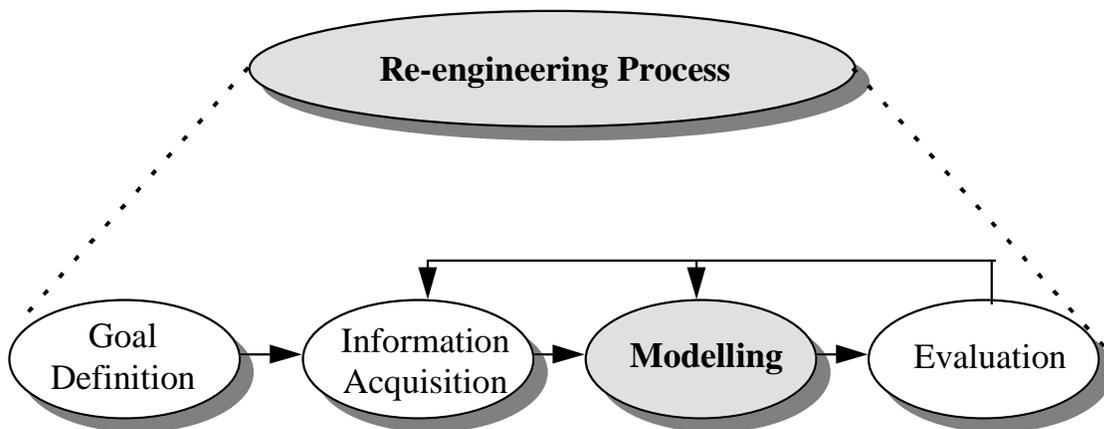


Fig. 1. The Re-engineering Process and its subprocesses.

These four subprocesses may be differently ‘instantiated’, depending on the chosen methodology. For example, Hammer says that re-engineering teams should try

to think as companies are starting anew. This means that, the execution of the *Information Acquisition* subprocess depends on whether it is executed within a ‘radical re-engineering approach’ (Hammer & Champy, 1993) or within an ‘incremental re-engineering approach’ (Davenport, 1993). Furthermore, a number of different techniques may be used for each of these subprocesses. For example, animation, simulation or statistical analysis are some of the techniques that may be used for *Evaluation*. The use of a specific technique is also a characteristic of the chosen methodology.

The focus of this paper is on the third subprocess of the Re-engineering Process, i.e. on *Modelling*. This subprocess is the core of the Re-engineering Process as the goals of a re-engineering project can only be reached if the business process is analysed properly and afterwards implemented according to the constructed business process model. A general description of *Modelling* is given in section 2. As it can be seen in Fig. 2, *Modelling* may be viewed as consisting of three tasks, namely: *Choose Modelling Philosophy*, *Choose a Modelling Formalism* and *Apply Formalism to the selected Business Process*.

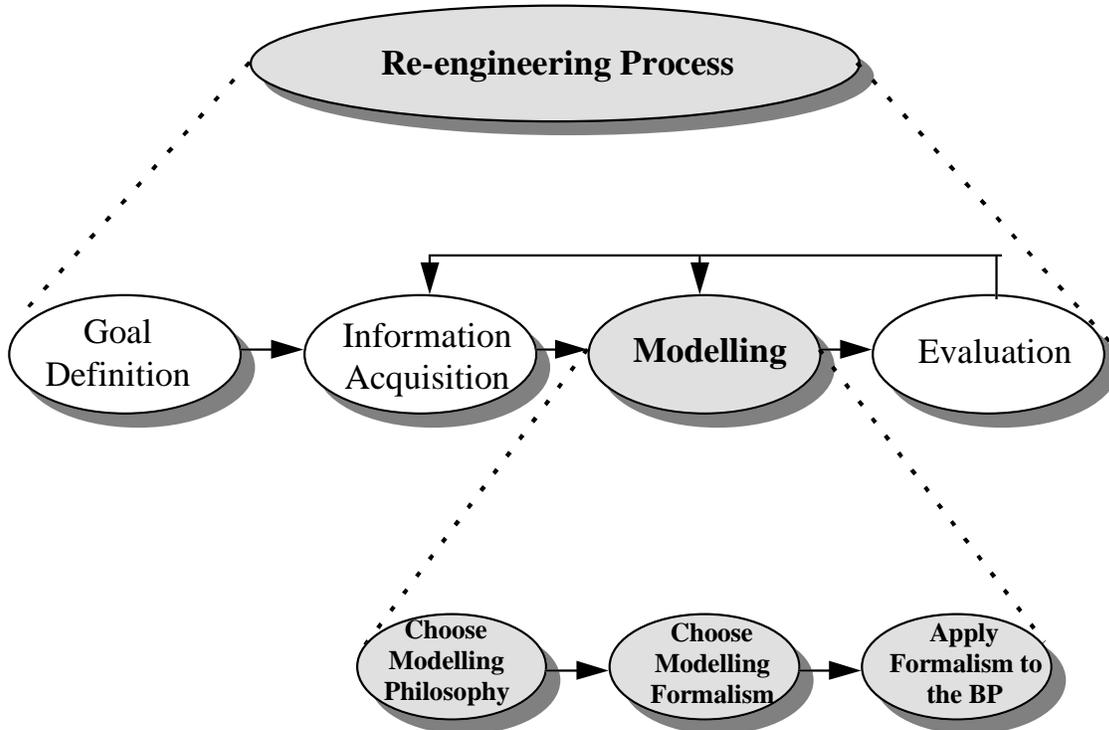


Fig. 2. The tasks of Modelling.

In section 2 a brief description of the business process objects to be modelled is given and a number of requirements to be satisfied by business process modelling formalisms is set. The first modelling task is analysed in section 3. Business process modelling approaches are classified under two modelling paradigms, namely task-oriented paradigm and business policy-oriented paradigm. The other two modelling tasks are analysed in section 4 by describing and comparing two representative approaches of the two modelling paradigms which are applied to the same example. Finally, section 5 draws some conclusions from this work and focusses on issues which still remain open in this area and gives some suggestions for further work.

2. Modelling Business Processes

The task of modelling in general, aims to provide an abstract description of one slice of reality by omitting details and thus reducing complexity which is usually inherent in real world situations. In the Re-engineering Process, modelling is the task of producing an abstract description of an actual or a proposed business process. A characteristic of a business process is that most of its elements are enacted by human actors.

Developments on business process modelling have been influenced a lot from other areas like for example CSCW/Groupware (Ellis & Wainer, 1994), Office Automation (Zisman, 1978; Ellis & Bernal, 1982), Software Process Modelling (Curtis et al., 1992), Requirements Modelling (Fickas & Finkelstein, 1993), Conceptual Modelling (Brodie et al., 1984) and Transaction Management (Wächter & Reuter, 1992).

Two very important issues in business process modelling are the information to be modelled and the modelling formalism. The following two subsections discuss about these issues in relation to business process modelling.

2.1. Business Process Objects

A business process model should be able to depict all process-related information. This information can be seen at two levels: at the re-engineering level and at the implementation level. This paper deals with process modelling at the re-engineering level. The produced model is later transformed to another model at the implementation level, in order to be used by application development programs or to be directly executed in an existing working environment.

At both modelling levels, there are certain kinds of information which have to be modelled; the intersection is called *core business process information*. In this paper, it is viewed in terms of objects which are called *core business process objects*. Fig. 3 shows the relation between *the core business process objects* and the level specific objects.

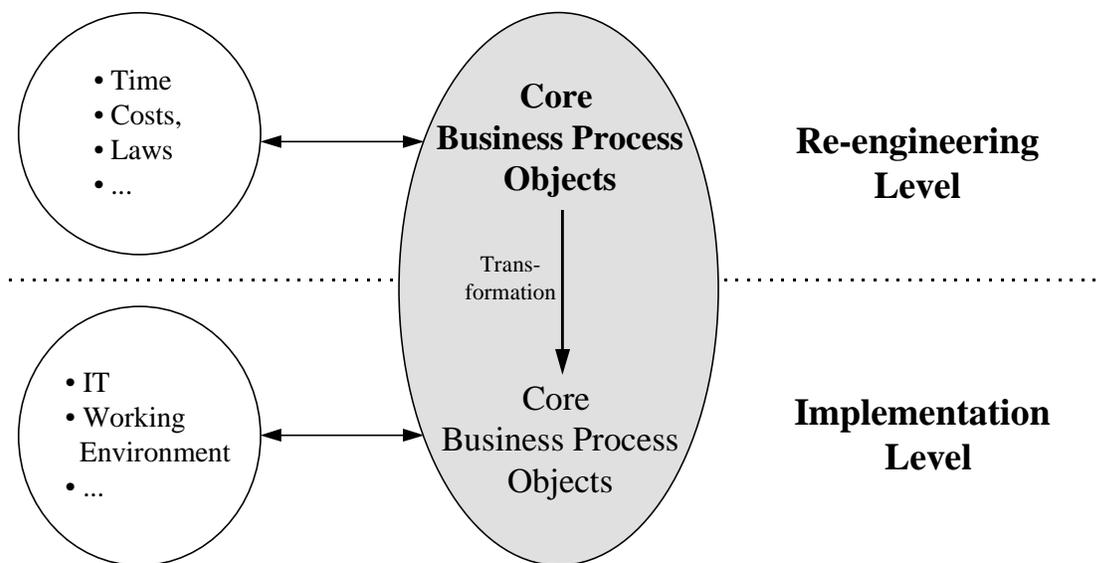


Fig. 3. Core business process objects and level specific objects.

The main *core business process objects* may contain information about:

- **activities** which are the basic elements on which a business process is built up. Usually they are defined as a unit of work that can not be subdivided.

- the **control** of a business process which describes *when* and *which* activity is executed.
- **resources** which are assigned to activities. These are objects that are necessary for the execution of activities, for example documents, data etc. The relations ‘has input’ and ‘has output’ (see Fig. 4) represent the resource flow which shows how resources are exchanged between activities.
- **organisational structure** which can consist of organisational units, people, roles, competence, etc. The relationship ‘has-actor’ represents the assignment of an object of the organisational structure to activities.

These *core business process objects* and a number of relationships are shown in a meta model depicted in Fig.4.

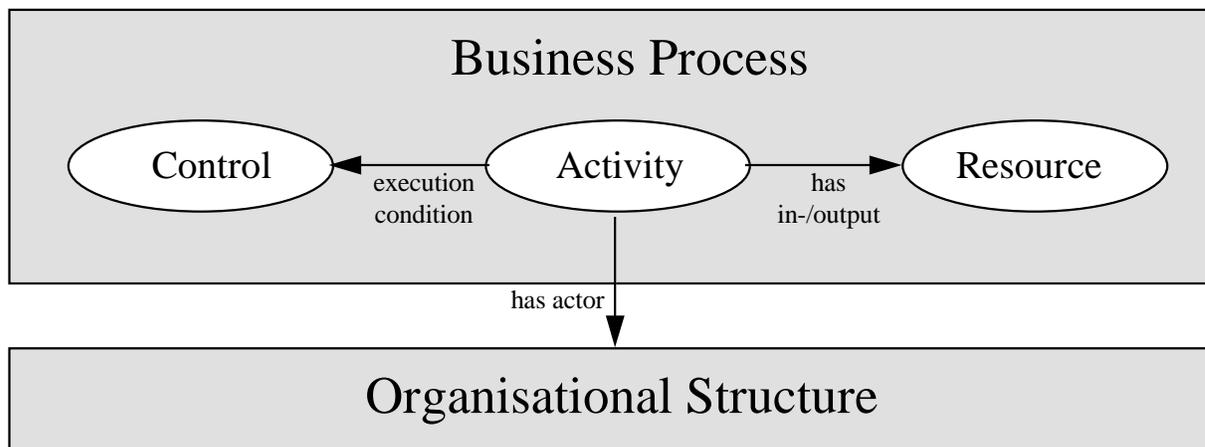


Fig. 4. A meta model of the core business objects and their relationships.

Apart from the core information, additional re-engineering specific information, like for example the execution time and cost of each activity, the activity waiting time, the cost of people, laws governing the organisation and so on, should be modelled.

Therefore, a formalism selected for business process modelling should be able to depict all the aforementioned business process specific information. This formalism should also exhibit some other characteristics which are discussed in the next section which focusses mainly on the modelling of core business process objects. The

modelling of re-engineering specific objects is heavily dependent on the chosen modelling approach for the core business process objects and is not discussed here.

2.2. Requirements for Business Process Modelling Formalisms

What is essential in every kind of modelling is the used formalism. A formalism used for business process modelling should be able to *model* all process-related information and additionally *facilitate evaluation* of the modelled process (against some criteria already predefined during the first subprocess of the Re-engineering Process, called *Goal Definition*, see Fig. 1). The formalism should also *enable the transformation* of the business process model to another model on the implementation level. One further point to be taken into account during business process modelling is that the people who will be involved in the modelling process (either as information providers or as evaluators of the modelled process) are not necessarily IT experts. Following these observations we came up with a list of requirements to be satisfied by a formalism used for business process modelling for effectively applying re-engineering. These requirements dictate that such a formalism should:

- be graphical and easy to understand by all parties involved in the Re-engineering Process
- enable and encourage the reusability of the modelled components
- be supported by algorithms and enable the employment of certain techniques (e.g. animation, simulation, statistical analysis) appropriate for the analysis and evaluation of the model
- have clearly defined semantics in order to enable the transformation of the business process model from the re-engineering level to the implementation level (see Fig. 2).

The way these requirements are satisfied by a modelling formalism is greatly affected by the modelling philosophy behind the formalism and this is the subject matter of the following section.

3. Business Process Modelling Paradigms

A number of classification schemes for business process modelling approaches may be found in the literature (Curtis et al., 1992; Starke, 1994). These classifications schemes are mainly influenced by the implementation techniques used for the realisation of the different modelling approaches. This paper attempts to classify business process modelling approaches at a higher level of abstraction, along the lines of the classification referred by Karagiannis (Karagiannis, 1995).

The first task of the *Modelling* subprocess of the Re-engineering Process is the capture of information related to the operations of an enterprise. This information usually consists of a set of guidelines and rules describing, in a rather general way, requirements for and consequences after the execution of an activity, ways of activities execution, qualifications necessary for an actor to be responsible for an activity, etc. Actual business processes can be described by applying these general guidelines to a specific situation.

In this paper, current approaches to modelling business processes are classified in two different paradigms: *task-oriented* paradigm and *business policy-oriented* paradigm, depending on the philosophy they use to capture and model the required information. Task-oriented and business policy-oriented approaches correspond respectively to the ‘analysis’ and ‘synthesis’ approaches, as they are referred in (Karagiannis, 1995).

	<i>Task-Oriented Approaches</i>	<i>Business Policy-Oriented Approaches</i>
<i>Information Acquisition</i>	<i>High Complexity</i>	<i>Low Complexity</i>
<i>Modelling</i>	<i>Low Complexity</i>	<i>Low Complexity</i>

Fig. 5. Handling of complexity in the Re-engineering Process
(see also Fig. 1. in (Karagiannis, 1995)).

Complexity is an inherent aspect of human thinking. Approaches following the *task-oriented* paradigm execute the *Information Acquisition* subprocess of the Re-engineering Process adopting a philosophy which does not simplify this complexity. All potential situations and exceptions which may occur during the operation of an enterprise are forecasted and described, by applying the general description of the operations of an enterprise to various specific situations. The use of such a philosophy affects *Information Acquisition*, increases the complexity of the information to be modelled and, *Modelling* results into complex representation models. The complexity of these models is handled by using appropriate techniques such as decomposition, which are usually incorporated in the modelling approach, e.g. multi-level Petri nets (Tsalgatidou et al., 1994).

Business policy-oriented approaches follow a different philosophy which aims to reduce the complexity of human thinking during the *Information Acquisition* and *Modelling* subprocesses of the Re-engineering Process. The information which is captured and modelled is the general guidelines and rules which can be found in organisational handbooks. The use of appropriate representation schemes results into simple and compact representation models. The analysis and interpretation of the modelled business rules and guidelines is left to be handled later by the system which will be implemented (Croft & Lefkowitz, 1988; Hinkelmann & Karagiannis, 1992).

Some of the advantages of *task-oriented* approaches are that the modelling objects are usually simple and there are a number of algorithms for analysing the modelled business processes, while the large size and the complexity of these models are amongst their great disadvantages. *Business policy-oriented* approaches, on the other hand, have the advantage of producing compact, small and simple models, thus enhancing understandability of the modelled information but there don't seem to exist many algorithms for model analysis. These advantages and disadvantages are demonstrated by two representative examples described in the next section.

It is worth to mention that independently of the paradigm used for gathering and modelling the business process related information, the information contained in the 'running' business process at the implementation level, will be the same.

4. Examples of Modelling Approaches and Comparison

This section presents two typical exemplars for the task-oriented and business policy-oriented approaches, presented respectively in subsections 4.1 and 4.2. Both exemplars refer to business processes of an example trading company which buys products from different suppliers and sells them to various customers.

4.1. A Task-Oriented Approach

An example of a task-oriented approach is the RBN (Rule-Based Net) (Tsalgatidou et al., 1994) which is defined as a tuple

$$RBN = \langle P, T, F, O, N, sig, class, bu, if, then \rangle,$$

where P is a set of places which are inscribed with signals (i.e. triggers which can activate certain actions), T is a set of transitions inscribed with dynamic rules, F is a set of arcs connecting places to transitions and transitions to places, N is a multiplicity function of arcs, O is the underlying structure of RBN (i.e. an object-oriented rule-based schema, where the behaviour of each object class is described in terms of dynamic rules grouped in behaviour units and triggered by signals of a specific type), sig is a function that maps places (i.e. elements of P) to signal types, $class$ is a function which maps transitions to object classes, bu is a function mapping transitions to behaviour units, if is a function mapping transitions to preconditions and $then$ is a function which maps transitions to actions.

In an RBN, every transition t corresponds to a dynamic rule of the underlying structure and is inscribed with the corresponding preconditions, actions, behaviour unit and class, defined respectively as $if(t)$, $then(t)$, $bu(t)$ and $class(t)$. Each place p of P is inscribed with a corresponding signal type $sig(p)$ and the parameters of $sig(p)$.

Referring to the meta model of Fig. 4, places of the RBN correspond to the control objects of the meta model and groups of RBN transitions correspond to the activity objects of the meta model. The RBN underlying structure O corresponds to

the resource object of the meta model and contains also some information about the organisational structure.

Fig. 6 depicts an RBN that models the business process which is executed when new stock for a product arrives at the example trading company. In this case, the warehouse clerk updates the stock and then a sales clerk has to examine all customer orders for that products which were put on hold as they could not be satisfied due to lack of stock. All the possible cases have to be explicitly described; for each customer order there are two possibilities: either, there is enough stock and the order is satisfied, or the stock is not enough and the order is not satisfied. If there were other cases, they should be also explicitly described and all the possibilities should be examined.

It can be seen that, for the description of such a simple procedure, everything has to be explicitly stated. The result of this is an RBN the size and the complexity of which is very big compared to the complexity of the real world situation. This is the problem with all net-based approaches. Of course, there are a number of advantages for this approach. The main advantage is that the formality of the underlying Petri net enables the development of a number of algorithms for checking desirable behavioural (marking-dependent) as well as structural (marking-independent) net properties. These properties are: *reachability*, *liveness*, *persistence* and *structural liveness* (Tsalgatiidou et al., 1994). Furthermore, since the RBN model is graphical and executable, animation and simulation techniques could be also used for the validation of the model.

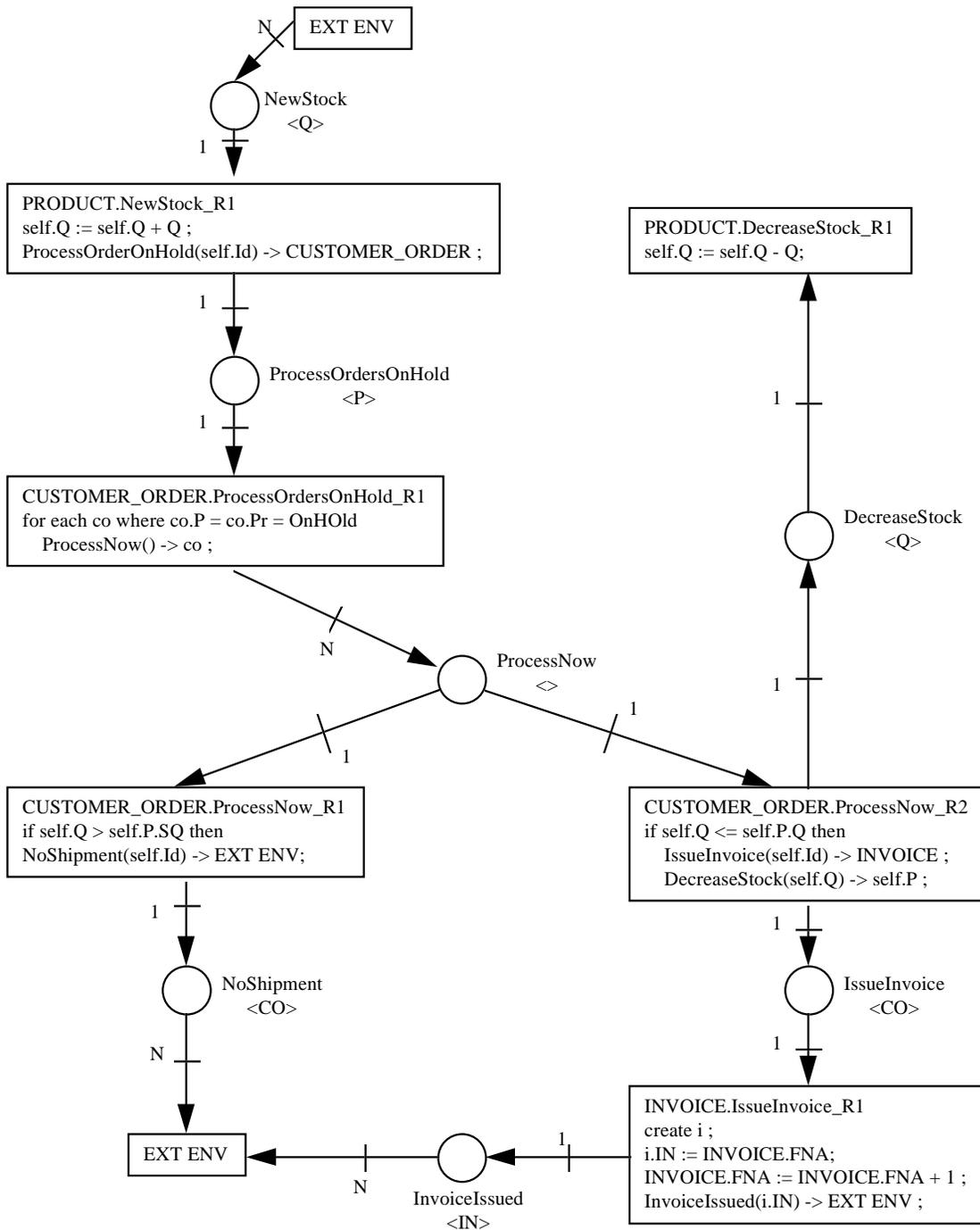


Fig. 6. A RBN depicting the activities which take place when new stock for a product arrives.

4.2. A Business Policy-Oriented Approach

The main idea of this approach is to represent explicitly the business rules of an organisation. Due to space limitations a simplified view is presented. For further details please refer to (Hinkelmann & Karagiannis, 1992; Hinkelmann & Karagiannis, 1990). First the formalism is described by referring to a process which deals with the processing of a customer order that arrives to the example trading company, and then we explain how a modelled business process can be executed.

A modelled business process is defined as a tuple

$$\langle P, A, I, RG, RS, RE \rangle$$

where P is set of business process classes = $\{P_1, P_2, \dots\}$, A a set of activity classes = $\{A_1, A_2, \dots\}$, I a set of information classes = $\{I_1, I_2, \dots\}$, RG a set of generation rules = $\{RG_1, RG_2, \dots\}$, RS a set of sequencing rules = $\{RS_1, RS_2, \dots\}$ and RE a set of execution rules = $\{RE_1, RE_2, \dots\}$.

The rules describe the guidelines of an organisation and are represented by production rules. Rules are usually used in more than one business process. For example, the guideline that a permission has to be given on the application, is applicable to nearly all permission procedures. Rules are collected to rule classes whereby the same rule can belong to several rule classes.

Generation rules determine the activities the business process consists of (by setting specific goals in the attribute Goal of business process instances). The following generation rule describes the business guideline which states that 'at the end of the process the product should either be sent to customer or not'.¹

```
IF          (an Information.Object of ?Current.Process
              is ?Product)
THEN (a Goal of ?Current.Process is
      '(the Send.to.customer.p of ?Product
        is ?yes.or.no)
```

¹Variables are marked by a question mark, e.g. ?Product.

Sequencing rules describe the control of the business process and thus they determine the order of the activity instances during the execution of the business process. For example, the following rule describes the business guideline which states that ‘before the product is sent to the customer his or her creditability has to be checked’.

```

IF      (an Activity of ?Current.Process is
        ?Send.product.to.customer)
        (an Activity of CurrentProcess is
        ?Check.customer.creditability)
        ...
THEN   (a Subsequent.Activity of
        ?Send.product.to.customer
        is ?Check.customer.creditability)

```

Execution rules select the actors of activities and determine which information objects are needed for the execution of activity instances. The following rules describe the business guideline which states that ‘the actor of activity ‘Prepare Invoice’ should be a member of the ‘Accounts.Dep.’ and should have the role ‘clerk’’.

```

IF      (?Prepare.invoice is in class Prepare.Invoice)
        (?Person is in class Person)
        (a Department of ?Person is Accounts.Dept)
        (a Role of ?Person is Clerk)
THEN   (a Potential.Actor of ?Prepare.invoice is ?Person)

```

Activities are structured in an object hierarchy. They can be system activities, which are executed without any user interaction or activities which are being activated and executed by one or multiple persons. For example the activity class ‘Prepare Invoice’ could be modelled as follows:

```

class: Prepare.invoice
      Superclass: Atomic.Activity
      Potential.Actor: unkown
      Input.Information: unkown

```

Output.Information: unkown

Goal: (the invoice of ?Product is prepared)

Precondition: (the packed.p of ?Product is ?yes.or.no)

Subsequent.Activity: unkown

Most of the attributes are still uninstantiated. They will be instantiated shortly before execution time when the activity instance is activated. This is the time when information is most detailed. The two attributes Precondition and Goal refer to the control of the business process. In the activities' class definitions these attributes are filled with uninstantiated formulas determining the preconditions for executing the activity respectively the purpose, which the activity is applied for.

Referring to the meta model described in Fig. 4, information objects correspond to the resource objects of the meta model, the attributes Input.Information and Output.Information correspond to the relations between activity and resource objects, the attribute Potential.Actor depicts the relation of the activity to the organisational structure and the attributes Subsequent.Activity and Precondition depict the relation between activity and control objects of the meta model.

Information objects can be seen as documents or data. They are used for describing the states of the business process when it is executed. They are also described as classes that are instantiated at execution time.

class: Product

Superclass: Information.Object

Invoice: unkown

Send.to.customer.p: unkown

Packed.p: unkown

Quantity: unkown

Price: ...

...

Process instances describe global information about the business process like the information objects, the start time, the current activity, the goals etc.

class: Customer.Order
Superclass: Business Process
Information.Object: unkown
Goals: unkown
Start.time: unkown
Activity: unkown
Current.Activity: unkown
...

To start a business process the process initiator describes what he or she wants to do, i.e. he or she refers to a business process. Each business process is described by goals that must be satisfied during task execution. The goals determine milestones of the business process but do not give any information about how they are to be reached. These goals are set by firing generation rules. As the activity classes are also described by goals, a planning mechanism will instantiate activity classes that satisfy the goals. The preconditions of the instantiated activities are added to the goals of the business process and the system tries to satisfy them by instantiating activity classes with the corresponding goals. This is done iteratively until no more activity instances can be generated. After firing the sequencing rules the activity instances are ordered. Now the activity instances with true preconditions can be executed. So the execution rules are fired and afterwards the activities are delegated to the determined actors. The execution of activities can cause a manipulation of the information objects of the process. Then the generation rules are fired again, afterwards the execution rules are fired and so forth. Eventually the goals of the business process should have been reached.

4.3. Comparison

The two modelling approaches described before, can be both used for modelling business processes. In task-oriented approaches, the necessary changes are made directly and explicitly to the model and then validation algorithms as well as simulation and animation techniques may be used for evaluating the effect of the

changes to the process model. Changes to the business policy-oriented models are mainly incorporated by changing the appropriate rules. Then, execution algorithms and techniques like animation and simulation are used for evaluating the new business process models. In the following, each formalism is examined against the requirements which were set in section 2.2 .

The first requirement (i.e. to enable the modelling of all re-engineering related aspects of real world business processes) is satisfied by both approaches. However, business policy-oriented approaches seem more suitable for describing reality, as the descriptions are much simpler than the ones of task-oriented approaches which usually result in rather complex representations of the real-world. For example, when a business process needs four activities to be executed serially but in any order and a task-oriented approach is used, the result will be a very complex network (24 branches), if a control object is not offered for describing such situations. This situation is modelled by a business policy-oriented approach in a much simpler and compact way. The graphical nature of net-oriented approaches (which constitute the majority of task-oriented approaches) is an advantage which contributes in the enhancement of the understandability of the model by the users. The business policy-oriented approaches are usually not graphical, however, rule representations are usually closer to human perception of the real world. Furthermore, such descriptions could be used as an input to a tool which generates graphical net descriptions according to the situation at hand.

Regarding the reuse of modelling objects, the use of both approaches result in modelling objects and rules which can be reused in the modelling of many business processes. Concerning the application of validation algorithms to the models , there are a lot of such algorithms as well as simulation and animation techniques which can be applied to task-oriented approaches. For business policy-oriented approaches they are not many algorithms to be applied, however, techniques like simulation or animation can be easily used.

The transformation of the business process model to the implementation level is easy in the case of the task-oriented approaches, as the semantics are clear and the modelled objects are understandable by the users. The business policy-oriented

approaches have rich semantics, however, the transformation to the implementation level, for example to a model to be used by a Workflow Management System is not so easy, because existing Workflow Management Systems are mainly working according to task-oriented approaches.

5. Conclusion

The use of Re-engineering in an increasing number of enterprises, by focussing on business processes and by using IT as an enabler for the development of new products, has led to a revolutionary change of the ontology behind business matters. The task of business processes modelling within Re-engineering was the focus of this paper.

Summarising, Re-engineering was viewed as a process consisting of four subprocesses: *Goal Definition*, *Information Acquisition*, *Modelling* and *Evaluation*. Business process modelling objects were described and a set of requirements was set for modelling formalisms. Business process modelling approaches were classified under two modelling paradigms, depending on the philosophy behind them. These two modelling paradigms were analysed and two representative examples were described.

Each approach has a number of advantages and disadvantages and the selection of the modelling formalism is affected by a number of factors. One of these factors is the business process to re-engineer. For example, if the business process to re-engineer is a process at the management level which means that flexibility is important, a business policy-oriented approach is more appropriate. If the business process to re-engineer is one of the everyday processes which is very strict, then a task-oriented approach could be more feasible. The re-engineering goals could also influence the selection of the modelling approach. Thus, if the re-engineering goal is 'flexibility' during business process execution, a business policy-oriented approach seems more appropriate; when the main re-engineering goal is to improve time and cost of a strict and well-defined process, then a task-oriented approach should be used.

Currently there are few tools for supporting the execution of business policy-oriented models. The existing tools are mainly Workflow Management Systems which are based on task-oriented approaches. Thus, if the implementation of a business process is going to be performed using a Workflow Management System, then a task-oriented approach seems to be more suitable for the Re-engineering Process. Nevertheless, it seems that business policy-oriented approaches are becoming more and more important and it is believed that they will have an impact on future Workflow Management Systems (Abbott & Sarin, 1994).

Our future work focuses on the combination of task-oriented and business policy-oriented approaches. More specifically we are thinking of applying the business policy-oriented approach during *Information Acquisition* and generate afterwards a task-oriented model. We think that the use of a business policy-oriented approach will facilitate *Information Acquisition* and at a later stage, we can gain the advantages of the task-oriented approach for improving the business process.

Acknowledgments

The authors would like to thank the members of the BPMS group of the University of Vienna and Mr. Dimitris Gouscos, Ph.D. candidate at the University of Athens, for their contribution in this work.

References

- Abbott, K. R., & Sarin, S. K. (1994). Experiences with Workflow Management: Issues for the Next Generation. *Proceedings of the Conference of Computer Supported Cooperative Work (CSCW'94)*. ACM Press, 113-120.
- Brodie, M. L.; Mylopoulos, J. & Schmidt, J.W. (1984). *On Conceptual Modelling*. Springer-Verlag, New York.
- Croft, W.B., & Lefkowitz, L.S. (1988). A Goal-Based Representation of Office Work. In: W. Lamersdorf (Ed.), *Office Knowledge: Representation, Management, and Utilization*. Elsevier Science Publishers B.V. (North Holland), IFIP.

- Curtis, B., Keller, M.I., & Over, J. (1992). Process Modeling. *Communications of the ACM*, 35 (9), 75-90.
- Davenport, T. & Short, J.E. (1990). The New Industrial Engineering: Information Technology and Business Process Redesign. *Sloan Management Review*, Summer 1990, 11-27.
- Davenport, T. (1993). *Process Innovation*. Cambridge, MA: Harvard Business Press.
- Ellis, C.A., & Bernal, M. (1982). Officetalk-D: An experimental office information system. *Proc. ACM-SIGOA Conference on Office Information Systems* (June 1982), 131-140.
- Ellis, C., & Wainer, J. (1994). A Conceptual Model of Groupware. *Proceedings of the Conference of Computer Supported Cooperative Work (CSCW'94)*. ACM Press, 79-88.
- Fickas, S. & Finkelstein, A. (Eds.) (1993). *Proceedings of the IEEE International Symposium on Requirements Engineering*. San Diego, CA, 4-6 Jan. 1994. IEEE Computer Society Press.
- Hammer, M. (1990). Reengineering Work: Don't Automate, Obliterate. *Harvard Business Review*, July-August 1990, 104-112.
- Hammer, M. & Champy, J. (1993). *Reengineering the Corporation*. Harper Business, New York.
- Hinkelmann, K. & Karagiannis, D. (1992). Context-Sensitive Office Tasks: A Generative Approach. *Decision Support Systems* 8 (1992), 255-267.
- Hinkelmann, K. & Karagiannis, D. (1990). Vorgangsaspekte im Dienstleistungsbereich. In A. Reuter (Ed.), *Proc. GI-Jahrestagung*. Informatik Fachbericht No. 258, Springer (in German).
- Karagiannis, D. (1994, May). Towards Business Process Management Systems. *Tutorial at the International Conference on Cooperative Information Systems (CoopIS'94)*. Toronto, May 1994.

- Karagiannis, D. (1994, Sep.). Die Rolle von Workflow-Management beim Re-Engineering von Geschäftsprozessen. *dv Management* 3/94, 109-114 (in German).
- Karagiannis, D. (1995). BMPS: Business Process Management Systems. In this *SIGOIS special issue*.
- Medina-Mora, R., Winograd, T., Flores, R., & Flores, F. (1992). The Action Workflow Approach to Workflow Management Technology. *Proceedings of the Conference of Computer Supported Cooperative Work (CSCW'92)*. ACM Press, 281-288.
- Sarin, S. K., Abbott, K.R., & McCarthy, D. R. (1991). A Process Model and System for Supporting Collaborative Work, In: P. de Jong (Ed.), *Proc. of the Organizational Computing Systems Conference*, SIGOIS Bulletin, Vol. 12, No. 2,3.
- Soles, S. (1994). Work Reengineering and Workflows: Comparative Methods. In (White & Fischer, 1994), 70-105.
- Starke, G. (1994). Business Models and Their Description. In: G. Chroust, A. Benczur (Eds.), *Workflow Management - Challenges, Paradigms and Products / CON'94*, 134-147.
- Tsalgaidou, A., Gouscos, D. & Halatsis, C. (1994). Dynamic Process Modelling through Multi-level RBNs. *Procs. 4th Conf. on Dynamic Modelling of Inf. Systems (DYNMOD-IV)*, 10/1994, Noordwijkehout, The Netherlands.
- Wächter, H., & Reuter, A. (1992). The ConTract model. In: A.K. Elmagarmid (Ed.), *Database Transaction Models for Advanced Applications*, Morgan Kaufmann Publishers, San Mateo, CA.
- White, T. E., Fischer L. (Eds.) (1994). *The Workflow Paradigm*, Alameda.

Zisman, M.D. (1978). Use of Production Systems for Modeling Asynchronous Concurrent Processes. In: *Pattern-directed Inference Systems*, Academic Press Inc.