Yugoslav Journal of Operations Research
10 (2000), Number 1, 27-35

A METHODOLOGY FOR BUSINESS PROCESS MODELLING

Sini{a NE[KOVI], Branislav LAZAREVI]
Faculty of Organisational Sciences,
University of Belgrade, Yugoslavia

Abstract: This paper presents a methodology for business process modelling. The methodology recognizes and separates logical and physical models of business processes. The Logical Model represents collection of fundamental and atomic business functions (activities) mostly independent of organisational and technological constraints. The Physical Model, in the context of the given organisational and technological environment, defines a business process as a thread of specific implementations of specified fundamental activities. Structured System Analysis is used as a tool for logical business process modelling and Activity Diagrams as a tool for physical business process modelling.

Keywords: Business process modelling, functions, activities, logical model, environment model, physical model, structured system analysis, activity diagrams

1. INTRODUCTION

Recently, the problem of business process modelling has attracted a lot of attention, growing up into a new multidisciplinary research area. Business process modelling (BPM) is important for:

- Business process reengineering, i.e. the reorganization of business processes in order to improve their efficiency;
- Definition of business procedures in quality assurance studies;
- Information system development, i.e. the automation of business processes.

Since these three areas are closely related, the need for unique methodology for business process modelling is obvious.

A business process may be defined as a set of linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships [9]. A methodology for BPM comprises a collection of models and tools that enable an unambiguous, complete and formal description of business processes. A methodology of BPM is, also, a "business process" itself, i.e. a sequence of the activities that should be taken in order to obtain the BPM for the system under consideration.

There are a number of different models, tools and methodologies for BPM [1,3]. BP models are either based on the theory of the State Transition Diagrams [8], or the theory of Petri Nets [5]. Regardless of the formalism a model is based on, corresponding methodology usually assumes that the BPM of a given business process can be obtained by "reverse engineering", analysing and redesigning the current thread of activities it is composed of. This approach is partial in the sense that it analyses a current "physical" implementation of business processes only. The fact that an activity may be part of a number of different business processes is not explicitly taken into consideration.

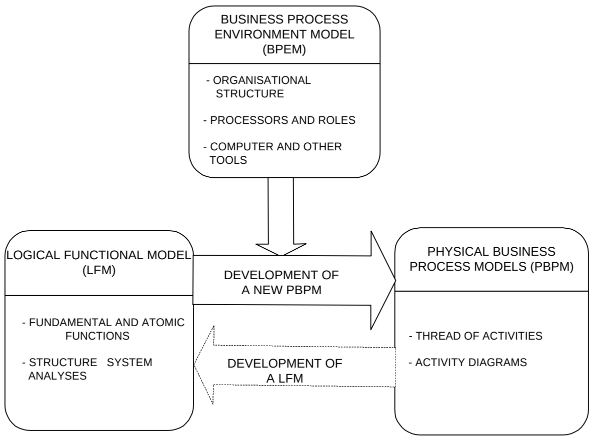

Figure 1: Logical and physical model

This paper suggests a different approach represented by the scheme in Fig. 1. The approach consists of the following steps:

- Development of the Logical Functional Model of the overall business system. The Logical Functional Model represents a collection of fundamental and atomic business functions of a system. "Fundamental" means that it is independent of current system organisational and technological constraints. "Atomic" means that it represents an atomic transaction or a "logical unit of work" that cannot be further decomposed into smaller meaningful units. A specific use of the Structured System Analysis provides a way to decompose a system into a collection of these fundamental and atomic functions [6]. "Reverse engineering" should be used as a method to develop a Logical Functional Model first, rather than a new Physical BPM directly. Fundamental atomic functions are also a functional specification of an information system that will support business processes.

- Development of the Business Process Environment Model (BPEM). The BPEM consists of a proper description of the organisational structure, with detailed specifications of "processors", their possible roles, as well as computer aided and other tools to perform assigned functions.

- Development of a collection of BPM Physical Models, each representing a specific business process as a thread of activities with processors and their roles to perform them. An activity represents a BPEM dependent implementation of one or more fundamental atomic function. Activity Diagrams (AD) from the UML standard [8] are used as a basic tool for physical BPM.

There is one important additional reason to separate logical and physical BPM. The LFM is much more "stable" than the PBPM. The LFM changes rarely in a given business system and often can be applied, fully or partially, in other business systems, too. The Physical Model is more frequently subject to change due to frequent changes in the BPEM. The separation of logical and physical BPM makes it possible to define two "life-cycle" models for BPM:

1. Basic life-cycle model: Analyses of current business processes -> Development of the LFM of a business system -> Development of the BPEM -> Development of the PBPM. The Basic life-cycle model needs only to be applied for the first BPM study or when a business system undergoes fundamental changes.

2. Adaptive life-cycle model: Analyses of current business processes ->Development of the new BPEM -> Development of the new PBPM. It is used to adapt a PBPM to business system organisational and technological changes.

In almost all other approaches to BPM (an extensive survey can be found in [1,3]) the adaptive life-cycle model is used, or more precisely, (physical) models of specific business processes are constructed directly, without any concern for common fundamental business activities. Section 2 and Section 3 describe tools and methods for BPM logical and physical modelling, respectively. Some experiences in applying the methodology and the state of the art of BPM "case tool", under development, are briefly discussed in the last section of the paper.

2. LOGICAL FUNCTIONAL MODEL

A Logical Functional Model (LFM) represents a collection of fundamental "logical" business functions which enables an enterprise to accomplish its mission and satisfy its goals. The LFM consists of a minimal complete set of business functions capable of fulfiling a system purpose, regardless of how the system is implemented. It is the functional specification model of a system with only true requirements.

There are different sources of false requirements [6]:

- Technological false requirements that originate through incorporation of either current or new system technological or organisational characteristics;
- Arbitrary false requirements, i.e. requirements making a system do more than is necessary to accomplish its purpose;
- The modelling methodology and modelling tools may impose some false requirements too.

Structured Systems Analysis (SSA) [2] is a well known and widely used method for the functional specification of information systems. It is also chosen here as a general methodological framework. In particular, the principles of so called "Essential Systems Analysis" [6] are used to obtain LFM with only essential, true functional requirements.

The SSA considers a business system as a function with a number of input and output data flows by which it communicates with external entities (interfaces). The SSA also provides a decomposition technique to represent a business system as the hierarchy of functions. Each level of decomposition is represented by a Data Flow Diagram (DFD). On the top level of the hierarchy, the DFD called "Context Diagram" consists of a function (overall business system), a collection of external entities connected to the function with data flows. A DFD on lower levels represents data flows between business functions (activities, processes), corresponding data stores and external entities. The decomposition is performed in the following way: a business function from one DFD can be decomposed and represented with a new DFD on the next lower level of the hierarchy.

The functions at each level should completely define the problem and should be mutually as independent as possible. If the resulting functions are mostly independent, each can be analysed separately from the others and decompose further, if necessary. Decomposition continues until atomic functions are identified. An atomic function is in fact an "atomic transaction" i.e. a "logical unit of work", the real system activity that is either completed or did not cause any change in the system state. In other words, an atomic function cannot be further decomposed into a collection of semantically significant, mostly independent functions.

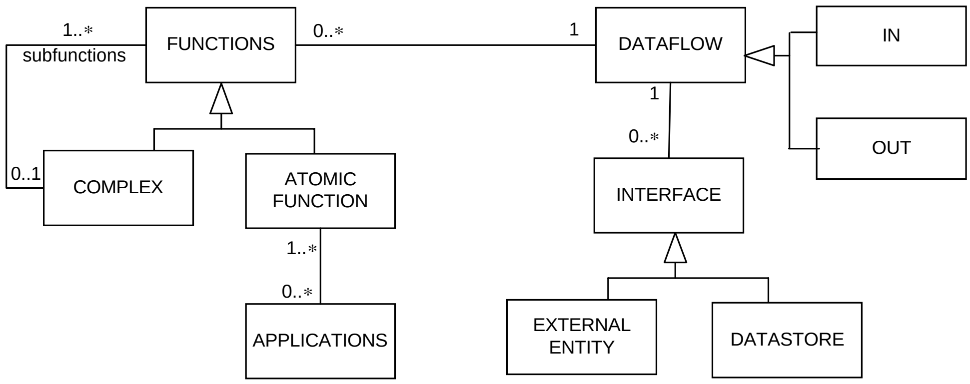

Figure 2: The LFM metamodel

In order to obtain fundamental atomic functions Essential Systems Analysis suggests the following:

- The functions a system is decomposed into, at any level of decomposition, interface each other only through data stores. This is in fact the test of the completeness of a function. When all the actions it consists of have been performed the system must become idle until inputs triggering it occur again.
- The functions a system is decomposed into have to be fundamental functions. A fundamental function consists of tasks that the system has to do even if it can be implemented using perfect technology. Perfect data processing technology consists of perfect processors and perfect memory. A perfect processor would be able to do everything instantly, while perfect memory would be able to hold an infinite amount of data instantly accessible to any processor. Both a perfect processor and a perfect memory would cost nothing, consume no energy, take no space, etc.
- The data stores are made of perfect memory. They contain minimal realisation of the system state, i.e., all the data the system has to remember in order to carry out the fundamental functions.

It is out of the scope of this paper to further discuss the methodology of SSA with Essential Systems Analysis recommendations. Instead, the metamodel of DFD following these recommendations is given in Fig. 2 and an example of LFM, to be used later in the paper, is given in Fig. 3.

The DFD metamodel is represented as a UML Class Diagram [8]. The DFD is a specific directed graph with functions and interfaces as the nodes and data flows as edges. The fact that functions cannot interface each other directly is represented with the separate association of the DATAFLOW class to the FUNCTION class and the INTERFACE class, both with a multiplicity of one. DATAFLOW subtypes IN and OUT represent incoming and outgoing data flows, respectively. The decomposition is represented using COMPLEX (which is decomposed into 1 or more "subfunctions") and ATOMIC subtypes of the class FUNCTION. It is assumed FUNCTION represent the fundamental functions and DATASTORE the prefect memory of a system. The occurrences of the ATOMIC FUNCTION class are functional specifications of the applications that would support activities of business processes.

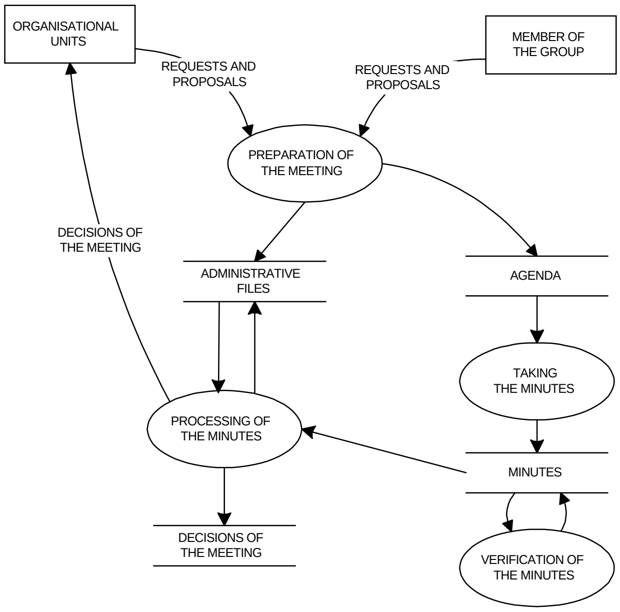

Figure 3: LFM of meeting preparation and processing

Fig. 3 represents the LFM for the preparation and processing of meetings. It is the LFM since it is aimed to specify the fundamental functions of any kind of meeting: meeting of a board of directors, meeting of a committee, meeting of a government and so on.

3. PHYSICAL BUSINESS PROCESS MODEL

As it was stated earlier, the Physical Business Process Model (PBPM) represents a thread of activities which collectively realise a business objective, within the context of an organisational structure defining functional roles and relationships. An activity can be defined as a piece of work (logical unit of work) that forms one logical step within a process [9]. An activity requires human and/or machine resources to support its execution.

The physical model defines the dynamics, i.e. the workflow, of a business process: the sequence of activities as business process steps as well as events and conditions for transition from one step (an activity) to another. The physical model also comprises processors (functional positions, roles, actors and agents) eligible for performing a specific business activity.

It seems that the Activity Diagram of the UML [6] is a most suitable tool for PBPM. The Activity Diagrams (AD) combines the ideas from the state transition diagrams [8] and Petri nets [7]. It is particularly useful in describing the behavior of complex business processes that have a lot of parallel activities. The metamodel for the AD used as the PBPM tool is given in Fig. 4.

The class NODE, together with the class TRANSITION is used to generally represent AD as a directed graph. One can distinguish three subtypes of NODE: (1) the ACTIVITY of a business process, (2) SYNCHRONISATION to represent the beginning and/or ending node of parallel activities and (3) DECISION ACTIVITY, which simply branches a transition to a "true" or "false" edge, depending on the value of the given condition.

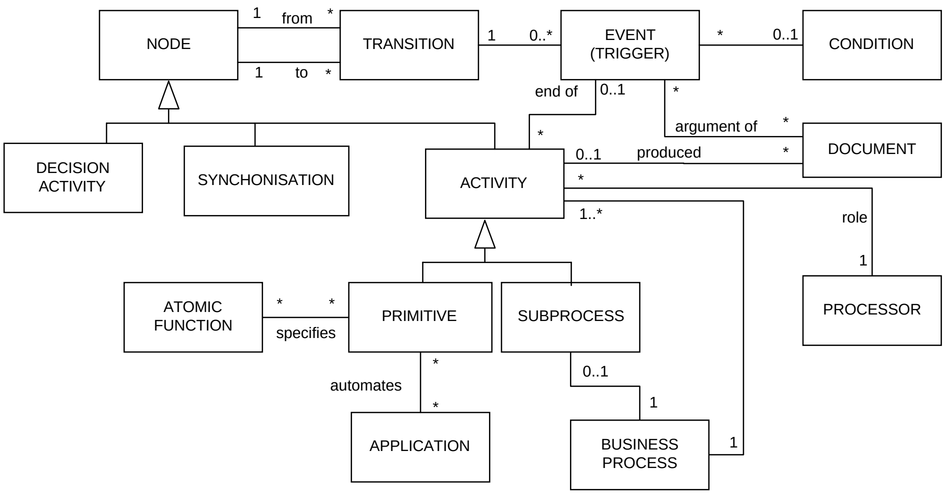

Figure 4: Physical Business Process Model

The TRANSITION from one business process state (step) to another is caused by an EVENT (trigger) and can be guarded by a specific CONDITION. Usually, an event represents the end of an activity. An ACTIVITY can be either PRIMITIVE or a SUBPROCESS, another business process which is represented by another AD. The representation of an activity as a subprocess provides the possibility to decompose the activity and represent it with a new, more detailed AD.

From the point of view of the methodology proposed in the paper it is most important to stress the association between PRIMITIVE (activity) and ATOMIC FUNCTION and PRIMITIVE and APPLICATION. This association shows that:

- An application may be used to automate a number of different activities, perhaps, only slightly modified for a specific processor;
- An fundamental atomic function can be used as an abstract specification for a number of activities. When some significant technological or organisational change takes place, only the new implementation of the activities has to be performed.

The AD for an example of the PBPM is given in Fig 5. It illustrates how fundamental atomic functions of the LFM for the preparation and processing of meetings are multiply used to construct PBPM of the deferent, but mutually dependent and synchronised meetings. To prepare and organize a meeting of the Board of Directors and process its conclusions it is necessary to prepare proposals and suggestions of the Board of Financial Managers and the Board of Marketing Managers.

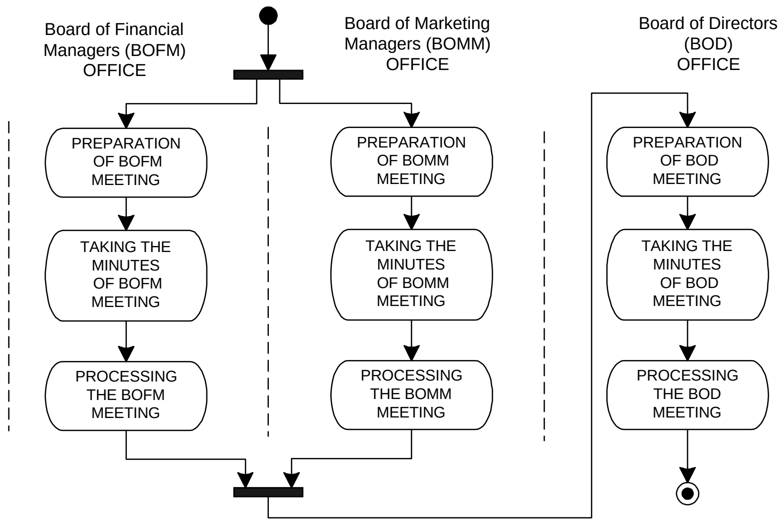

Figure 5: Example of BPM obtained from LFM

4. CONCLUSION

The paper has presented the methodology for business process modelling that separates logical and physical modelling and provides the tools and recommendations to build the Logical Functional Model and Physical Business Process Model of a business system. The motivations for this separation have been discussed in detailed.

The LFM and the PBPM are presented through their metamodels given as UML class diagrams. The clarity and compactness of the metamodels are not the only reasons to use them. The metamodels also represent the core of the CASE tool called BPM ARTIST. The prototype of the BPM ARTIST has been developed. It consists of the LFM module, the Structure System Analysis CASE tool (which is also part of the ARTIST - information systems development CASE tool [4]) and the PBPM module, a tool for Activity Diagram construction. The BPM ARTIST is aimed to generate Workflow Engine database to implement the BPM as an automated workflow.

The methodology and the BPM ARTIST has been successfully tested in a number of student projects and fully applied in the complex project of a Workflow system for a local government [5].

REFERENCES

[1] Bach, V., Brecht, L., Hess, T., and Osterle, H., Enabling Systematic Business Change: Integrated Methods and Software Tools for Business Process Redesign, Verlag Viewing, 1996.

[2] DeMarco, T., Structured Analysis and System Specification, Yourdon Press, New York, 1978.

[3] Ficher, L., Excellence in Practice: Innovation and Excellence in Workflow and Imaging, Future Strategies Inc., Lighthouse Point, Florida, 1997.

[4] Lazarevi}, B., and Ne{kovi}, S., "A CASE system for object-oriented development of information systems", Proceedings of ETRAN, 1992 (in Serbian).

[5] Lazarevi}, B. Ne{kovi}, S., and Skuli}, B., "A workflow system for administrative procedures in a local government", INFO Science, 1 (1) (1997) (in Serbian).

[6] McMenamin, S.M., and Palmer, J.F., Essential System Analysis, Yourdon Press, Englewood Cliffs, 1984.

[7] Muarata, T., "Petri nets: properties, analysis and applications", Proceedings of the IEEE, 77 (4) (1989).

[8] Rational Software Corp., Unified Modeling Language (UML), Version 1.1, 1997.

[9] WFMC, "The workflow reference model", Document Number TC00-1003 (http://www.aiai.ed.ac.uk/WfMC/).