

State-Oriented Business Process Modeling: Principles, Theory and Practice

(Overview of Doctoral Thesis: <http://media.lib.kth.se:8080/dissengrefhit.asp?dissnr=3375>)

Ilia Bider

IbisSoft, Sweden, email: ilia@ibissoft.se

Abstract: The progress in both modern IS development and business development in a high degree depends on our ability to understand, model, and control business processes. An approach to business process modeling and control is proposed based on the hypothesis that the ideas worked out in the Mathematical system theory for modeling and controlling physical processes can be successfully used for modeling and controlling business processes. One of the main ideas of mathematical system theory is to consider a process as a set of valid trajectories in a state space, and this idea is the keystone for the approach proposed. The paper discusses (a) how to represent a state of a business process, (b) how to represent direction of movement and speed in the state space, and (c) what type of formalism might be suitable for defining possible trajectories in the state space. The paper also overviews practical application of the the state-oriented approach. The method was tested in preparation of more than ten models of real business processes, and several support system has already been built based on it. The results presented in the paper summarize theoretical and experimental work mainly completed in an environment (a consulting company) that gave a possibility to observe the object of modeling, i.e., business processes, in its natural form, and test the results in practice.

Keywords: IS development, business development, business process, Mathematical system theory, business analysis, business modeling, object-oriented modeling, conceptual modeling, workflow

1 Motivation

1.1 IS development perspective

Today, at least half of the industrial software development is connected to business application development. During the past ten years, requirements on functionality of business applications have been slowly changing from command-based applications to the applications of workflow and groupware type. This change may be described as moving from the traditional, “*human-assisting*” systems, to a new generation of “*human-assisted*” systems.

A *human-assisting* system helps a human being only to perform certain activities, e.g. to write a letter, to print an invoice, to complete a transaction, etc. The relations between these activities, and the aim of the whole process are beyond the understanding of the system, but are a prerogative of the human participant. In a *human-assisted* system, the roles are reversed, the system has a complete picture of the process and is involved in all activities. Only when the system cannot perform some activity on its own, it will ask the human participant for assistance.

The difference between the old and new generations is essential, and it can be traced in all aspects of system development, user-interface including, see Fig. 1 below. A human-assisting system can be compared to a powerful tool set where the user should know exactly what tool to use and how to find it when it is needed. A human-assisted system functions like an assembly line conveyor bringing the user a task that he/she is to complete and a tool to complete the task with, i.e. a word processor.

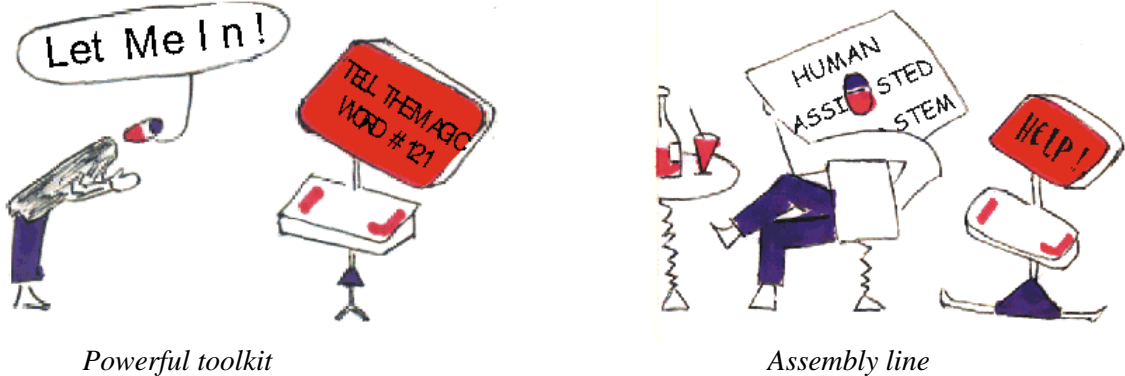


Fig 1. Shift in functionality of business applications

Note that the purpose of a human-assistant system is not to replace a human being, but to free him/her from routine, mechanical, uninteresting work. As opposed to AI, a human-assisted system takes care only on the formal part of the process, like bookkeeping, reminding, etc., leaving all intellectual tasks, like decision-making, to human beings.

The difference between the old and new generations is essential, and it can be traced in all aspects of system development, as shown in the Table 1 below.

Aspect	Old Generation (Human-Assisting Systems)	New Generation (Human-Assisted Systems)
Modeling	Data Modeling	Process Modeling
Data Base	Static and passive	Dynamic (history-minded) and active
User Interface	Functional (multilevel menus)	Navigational
Organizational aspect	Support existing way of running business	Suggests new, more effective way of achieving the business objectives

Table 1. Difference between two generations of business applications

All four factors are essential for building a new generation of business applications. However, a possibility of building a human-assisted system depends very much on whether we can understand, and model *business processes* that the system is supposed to support.

1.2 Business development perspective

The computer technology is becoming better with each day. The new technology, Internet in particular, creates great opportunities for companies and organizations to access global markets for products, services, skilled labor, and investment capital. However, the new technology creates only opportunities; to fully exploit them, it is not enough just to adopt the technology. In addition, a company or organization often needs to change its way of conducting business. For example, employing the Internet for accepting customer orders in combination with e-marketing may result in substantial increase in the amount of incoming orders and requests for information. If the company's internal organization has not been adjusted to handle the increased number of orders, the result will be frustrated customers who are not receiving their goods in time, receiving the wrong goods, or not receiving them at all.

The greatest advantage of the new technology is inexpensive channels of communication easily available even for small companies and organizations. These channels allow to arrange:

- Peer-to-peer communication between the company or organization and its customers, potential as well as existing, and suppliers.
- Mass communication with the external world, which allows to inform large groups of individuals and organizations about products and services delivered by a given company or organization.

Both types of communication are intrinsically linked to the internal business processes of the given company or organization.

Peer-to-peer communication is used for accepting orders, information requests, complaints, etc. Introducing electronic means for peer-to-peer communication, like email, and Web-based systems, can help to reduce the burden on call-centers and departments of technical support, sales, etc. However, a user of the fast communication channels, usually, has a presumption that fast channels ensure prompt responses. If this is not true, the users will be discouraged to use the new channels, and the burden on call centers will increase instead of decreasing.

To solve the problem, the business processes that are to be initiated via the new channels should be analyzed and optimized in order to fulfill the user's expectations. The incoming message should be promptly redirected to the responsible individual/department and go through all the relevant individuals/departments in a timely manner. It is also desirable to give the channel's users a possibility to follow-up their requests as they go through the organization.

Summarizing the above, there is no use in introduction of fast channels for peer-to-peer communication with the external world, until the company or organization introduces fast channels for internal communication.

Before appearance of the new technology, the channels for *mass communication* were affordable only for relatively large companies and organizations. With introduction of the new technology, such channels became available even for small organizations, via for example, email lists and Web sites. However, using such channels presumes understanding of basics of mass communication, which includes such questions as:

- Who your addressees are and what kind of information they might be interested in, and what kind of channels they are able to access?

- What do you want to achieve with each act of communication, e.g. remind of your existence, inform about new products?
- How to pre-process, and package various information messages in one communication sending?
- Where to get the relevant information and how to get it in a timely fashion?

All these questions are of equal importance. However, without solving the last one, the new technology will not work, as we will get Web sites in which the information is never updated. The readers will lose the interest to visit such sites, and thus all investment in the new technology will be lost.

Obtaining the relevant information in timely fashion is not a trivial task as, normally, it is not gathered in one place, but it is evenly spread through the whole organization. Some part of the relevant information might be found in the electronic form in various computer systems that are not connected to each other. Some relevant information may exist in the internal documents in various stages of preparation. The rest of relevant information might be in the heads of individuals participating in the business.

Introducing separate information-gathering procedures can be quite expensive and inefficient. A more efficient way is to direct this information to the information channel as soon as the information is produced in the frame of some business process. To arrange such “in-time” gathering we need:

- Identify all business processes that can serve as sources of information for mass communication.
- Describe and analyze them.
- Reengineer each process by introducing operations of information forwarding in the points of the process where this information is obtained or produced.

Introduce the redesigned processes in operational practice, which may require acquiring of an appropriate computerized support.

Summarizing the deliberations above, the progress in both modern system development and business development in a high degree depends on our ability to understand, model, and control business processes.

2 Why State-Oriented Approach

2.1 Basic notions

We follow the most general definition of business processes, see, for example, (Hammer&Champy, 1994), that defines a business process as a set of partially ordered activities aimed at reaching a well-defined goal. Some examples of goals are as follows:

- Reaching an agreement in business negotiations.
- Discharging a patient from the hospital in a (relatively) healthy state.
- Closing a sale.

This definition can be applied to main industrial processes that produce some “value” to the customers, as well as to support processes. Support processes are the processes that ensure that the main processes have enough resources to work problem free, e.g., hiring and firing personnel.

When discussing business processes, it is important to differentiate the process type from the process instance. The notion of *process type* is used to talk about the process in general, like:

- Sales process (in general),
- Processing insurance claims
- Decision-making

The notion of *process instance* is used to pinpoint a particular process, like:

- Processing a sales lead that concern a particular customer.
- Processing insurance claim #1345678.
- Passing an elderly care plan for 2002.

Two types of goals can be differentiated when discussing business processes: strategic and operational goals. Strategic goals, like customer satisfaction, growth, profit, etc. are associated with the process type. They explain why the process exists/should exist in the organization, and why it should be driven in a certain way. Analysis of the strategic goals results in the rules/procedures that dictate how the instances of the process should be run. All such rules for a given process type constitute a *process definition*.

Operational goals concern process instances, and they show when a process instance can be considered as finished. Examples of operational goals that correspond to the process types above are as follows:

- Understand the customer’s needs and make an offer (sales process).
- Insure that all basic documents that concern a particular insurance claim are collected and money are paid (processing insurance claims).
- Pass a decision on an elderly care plan based on the needs, available resources, and current legislation (decision-making)

Each process engages a number of participants, which can be roughly classified into artifacts, human beings (people) and organizations. The notion of *artifact* is used to represent any physical or abstract object like document, product, computer program, etc.

The notion of *human being* represents a person participating in the process. Very often, human beings participate in a business process not on their own, but on behalf of some company, political party, department, team, etc. Any of these unions can be represented by the concept of *organization*.

In relation to a particular process, the participants can be roughly divided into two categories: passive participants, and active participants. *Passive participants* are the participants that are subjected to transformation (or change) during the execution of activities, for example, a document being written, a car being assembled, a patient being treated in the hospital, an organization being reorganized.

Active participants, or *agents*, are the participants that perform actions aimed at changing passive participants. The active participants can be considered on the level of individual people participating in the activity, or on the level of organizations that they represent. Artifacts can also play the role of active participants, e.g., industrial equipment, robots, computers, etc. Both human and nonhuman active participants are often called resources, as they should be distributed among various activities and process instances.

2.2 State of the art

Currently, business process modeling, reengineering and control are being studied in the frame of two research fields: BPR – Business Process Reengineering, and WFMS – Workflow Management Systems. A lot of valuable research and practical work has been done in both fields in the last 15 years, see, for example, typical literature (Elzinga et al., 1999, Workflow 1995). However, the focus in this research is mostly concentrated on ordering the flow of activities, i.e. the workflow. As a result:

1. A business process is represented as sequence of activities with branches, loops, etc., i.e. in a form of a graph. Quite often, some version of Petri nets are used to represent such a graph, see for example, (Deiters&Gruhn, 1998).
2. Process control is focused at not allowing deviations from the predefined activities flow.

The focus on activities ordering has its roots in modeling production processes where each process is aimed at producing more or less the same physical product in a (relatively) large quantity. The deviation from the standard pattern in such a process often results in discarding an erroneous product. The theories, practical methods, and software tools based on the workflow view work quite well for the production-like processes. However, application of these methods and tools to the less structured processes (that does not have a standard pattern of behavior, or often deviate from the standard pattern) gives rise to the famous problem of workflow flexibility. So far, this problem has not been solved in a general way.

2.3 Mathematical system theory

Process orientation of businesses is a relatively new phenomenon. However, the idea of process modeling and control is quite old in relation to the physical processes. The needs for controlling physical processes, like plants, airplanes, rockets, etc. gave birth to the mathematical system theory, see for example (Kalman et al., 1963). For 50 years of its existence, a considerable amount of research has been done in this field. Implementation of the results from this research in practice, have helped to drastically decrease the production costs. Most production processes are highly automated, and the use of robots in industry is growing.

We strongly believe that the conceptual (and partly formal) apparatus worked out in the mathematical system theory can be successfully applied to the field of business process modeling and control, which constitutes the main hypothesis of the thesis.

One of the main notions of the mathematical system theory is a notion of state, which for continuous processes, is often defined by a finite number of state variables accepting real values. A process in such case is represented as a trajectory in the state space. The most common definition of a continuous-time system is through a set of differential equations of the form:

$$F(x, \dot{x}, w) = 0,$$

where $\dot{\mathbf{x}}$ denotes the derivative of the vector of state variables \mathbf{x} with respect to time, and \mathbf{w} is a vector of environment variables that model interaction between the process and the environment in which it runs. Such equations are binding the direction and speed of movement of the system in the state space to the position of the system in the space and the state of the environment.

For the continuous processes, the idea of control conceptually is quite simple: to ensure reaching a certain point in the state space (goal) through forcing the system to move closely to a chosen trajectory. The control is realized via changing the environment variables and thus changing the speed and direction of movement of the system. With such definition, even considerable deviations from the chosen trajectory can be dealt with by finding a new trajectory that leads us to the goal from the point to which the system unexpectedly came.

To apply the conceptual apparatus of the mathematical system theory to business processes, we need to:

- Find a way for representing states of business processes.
- Understand how to represent speed and direction of movement.
- Find out how to connect speed and direction of movement to the position in the state space.
- Find a suitable formalism to substitute the differential calculus.

2.4 What has been done

The work presented in the thesis concerned both theory and practice of application of the ideas of mathematical system theory to business processes. In theory, we tried to answer the questions formulated in the previous subsection. In practice, we tried to use the ideas for both building models of real business processes, and for building computer systems that help to control business processes.

The results presented in the thesis summarize the author's theoretical and experimental work in the field of business process modeling during the period between 1989 and 2001. The main part of the work was completed at IbisSoft AB, a small Swedish consulting company that specializes in the borderland between business and software development. The environment in which the research was conducted gave the possibility to observe the object of modeling, i.e., business processes, in its natural form, and to test the results in practice. The formalism presented in the thesis is based on the results from the research project completed in 1984-1986 by a group of researchers that besides the author included Maxim Khomyakov and Eugene Pushchinsky, details see in (Bider et al., 2000).

The rest of this paper is organized in the following way. In Section 3, we informally answer the questions formulated in section 2.3. In Section 4, we give an overview of a possible formalism that might substitute the differential calculus for business processes. In Section 5, we shortly overview our practical work. Finally, in Section 6, we give a summary of results achieved.

3 Principles

Our work on application of the ideas from the Mathematical system theory to business process resulted in the following conceptual model of business processes.

A *business process type* is defined as a subset of all possible trajectories in some state space. Each trajectory in a subset constitutes a *business process instance*. Thus, an instance is represented by a

function from time into the set of all possible states. A *state* of the process is defined by a “construct” that reflects the relevant part of the “business world” at some moment of time. The internal structure of the state construct depends on the business process in question. An example, of such structure for a business process related to a customer order is represented on Figures 2. The state structure includes:

- attributes (variables), like *To pay*, *Paid*, *Ordered*, etc., and
- references to various human and non-human participants of the process, like customer, product, etc. (more on participants see Section 2.1).

The state may have a variable structure, e.g., it may include repeating groups, see the list of ordered products in Fig. 2.

F1		O R D E R		Ibis:HRS	
Deal Category:travel		50331650		Deal #:00001	
CUSTOMER					
Company Name: Travelshop		Reference:		Job:Manager	
Tel :(< >)08 -5809090		Firstname:Ivar		Lastname :Svensson	
Pos	Article#	Article name	Ordered	Deliv	Sum
1	CS6080GR	Suitcase 60x80 green	9	9	10800.00
2	CB4030BL	Computer bag 40x30 black	20	20	6000.00
3					
4					
Mark	Way of del.	Weight			
Notes	•Closed deals•	Payment in 15 days	Disc.	%	
•Events•	F3	UAT(y/n)y	Total	16800.00	
	•Plans•	Invoiced	Freight	Tax	
		Paid		4200.00	
			To pay	21000.00	
00-05-23-22:42					

Fig 2. State representation of order processing.

Each business process has an *objective* or *goal*. The goal can be defined as a set of conditions that have to be fulfilled before a process instance can be considered as finished (end of the trajectory). A state that satisfies these conditions is called a *final state* of the process. The set of final states for the process in Fig. 2 can be defined as follows:

- for each ordered item $Ordered = Delivered$
- $To\ pay = Total + Freight + Tax$
- $Invoiced = To\ pay$
- $Paid = Invoiced$

The above conditions define a “surface” in the state space of this process.

Each process instance is driven forward towards the goal through activities that are executed either automatically or with a human assistance. An *activity* can be viewed as an *action* aimed at changing the process state in a special way. Execution of each activity results in a *change* in the process state, and thus a jump in the trajectory of the process instance.

A change in the process’s state can happen not only as a result of completing an activity, but also in many other, often non-anticipated, ways. For example, a customer may change, or cancel his/her order in the last moment (even when the goods have been delivered).

For many practical tasks, it is enough to present any change in the process state as happening at once. Thus, we can consider a function that represents a business process instance as having no segments with continuous changes. A moment of time when the process's state changes is called an *event* in the process's lifetime. Each completed activity results in an event, but as it has been pointed out above, events can happen unpredictably as well. The sequence of all events of the given process can be numbered in the ascending order to compose an *internal time axis* of the process. A sequence of the process's states taken after each event up to a certain moment of time forms the *history* of the process evolution (up to the chosen moment of time).

To link the internal time axis to the real or *external time*, event registration can be performed each time an activity is executed, or results of an unsolicited event are introduced to correct the state. A *registered event* is a record that links the change in the state of a process to the reality outside the process. For example, it can record the date-time when the event happened and/or was registered, name the responsible for the event, register comments on the event at the moment of registration (or even later), etc. A list of all events that were registered within the frame of a given process up to a certain moment of time constitutes the *chronicle* of the process, i.e. its written history. For example, the chronicle of the order processing up to the state illustrated on Fig. 2 can look like a list presented on Fig. 3.

F1		EVENTS		
Events in sale #		00001		
When	Deal	Resp	Activity	
000523 22:40	Deal	HRS	Call in	
000523 22:41	Deal	HRS	Pack list	
000523 22:42	Deal	HRS	Shipping	

F3 •Filter• •Unfilter• •Count• •Print• •Page 2•
00-05-23 22:42

Fig 3. Chronicle that corresponds the state on Fig. 2.

Activities can be planned first and executed later. A *planned activity* records such information as type of action (*goods shipment, compiling a program, sending a letter*), planned date and time, deadline, name of a person responsible for an action, etc. All activities planned, but not executed for a given process at a particular point of time constitute the process's *operative plan*. The plan should only list activities the execution of which will diminish the *distance* between the current state of the process instance and the *nearest* final state. The meaning of the term distance depends on the business process in question. Here, we use this term informally. For example, activities to plan for the process in Fig.2 can be defined in the following manner:

- If for some item *Ordered > Delivered*, shipment should be performed.
- If *To pay > Invoiced*, an invoice should be sent.
- If, on the other hand, *Invoiced > To pay*, a credit note should be issued.
- If *Invoiced > Paid*, steps should be undertaken to get money from the customer.

- If, on the other hand, $Invoiced < Paid$, money should be returned to the customer.

An example of an operative plan that corresponds to the state on Fig. 2 is presented on Fig. 4.

Conceptually, operative plan substitute the idea of derivates in continuous processes. Each activity on the least shows the “direction” of movement along some “axes” in the state space, and the speed of the movement (through planned date and deadline).

F1		D E A L ' S P L A N			Ibis:HRS
Company:Travelshop		50331650			Deal #:00001
Tel :(< 08 -5809090		Reference:			Firstname:Ivar
		Lastname :Svensson			
DeadLine	Activity	Resp	Counterpart	Document	
1 000526	Invoicing	HRS	Svensson		
2					
3					
4					
5					
6					
7					
8					
9					
10					
•Notes• F3		•Events•			
00-05-23 22:42					

Fig 4. Operative plan that complements the state from Fig. 2 and makes it valid.

A pair $\langle process_state, operative_plan \rangle$ is called a *generalized state*. A generalized state includes:

- *passive part* – attributes and references included in the state, and
- *active part or engine* – the operative plan that is responsible for moving the process forward.

Using the notion of generalized state, a process type (a subset of trajectories in the state space) can be defined as a subset of generalized states. The states included in this subset are called *valid*. Each trajectory in the state space that “goes” through the valid states constitutes a (valid) business process instance. The term “goes” means that:

- there exists an operative plan that complements a given state in such a way that the resulting generalized state is valid,
- the next change in the process state is “normally” caused by an execution of an activity from the complementing operative plan.

Note that the “abnormal” changes in the process state are due to unsolicited events, and there should be some constraints on what can happen during these changes.

Conceptually, defining the process through the set of valid states substitute the differential equations that are used for describing all possible trajectories of continuous processes.

A definition of a business process in the form of a set of valid states can be converted into an operational procedure called *rules of dynamic planning*, or *rules of planning* for short. Rules of planning specify what activities could/should be added to an invalid generalized state to make it valid. Using these rules, the process instance is driven forward in the following manner. First, an activity from the operative plan is executed and the state of the process is changed. Then, an operative plan is corrected to make the generalized state valid.

For example, executing the *invoicing* activity from the operative plan on Figure 4 can change the state of order processing shown on Figure 2 to the one shown on Figure 5. Applying the rules of dynamic planning will result in an operative plan shown on Figure 6. In addition, a new event is registered and added to the chronicle, see Figure 7.

F1		O R D E R		Ibis:HRS	
Deal Category:travel		50331650		Deal #:00001	
CUSTOMER					
Company Name: Travelshop		Reference:IvP		Job:Manager	
Tel :(< >08 -5809090_		Firstname:Ivar		Lastname :Petersson	
Pos	Article#	Article name	Ordered	Deliv	Sum
1	CS6080GR	Suitcase 60x80 green	9	9	10800.00
2	CB4030BL	Computer bag 40x30 black	20	20	6000.00
3					
4					
Mark	Way of del.	Weight			
Notes	•Closed deals•	Payment in 15 days	Disc.	%	
•Events•	F3	UAT(y/n)y 25.00 %	Total	16800.00	
	•Plans•	Invoiced 21000.00	Freight		
		Paid	Tax	4200.00	
			To pay	21000.00	
02-01-18 20:49					

Fig 5. The process's state after execution of the "invoicing" activity

F1		D E A L ' S P L A N		Ibis:HRS	
50331650				Deal #:00001	
Company:Travelshop			Reference:IvP		
Tel :(< >08 -5809090			Firstname:Ivar		
			Lastname :Petersson		
DeadLine	Activity	Resp	Counterpart	Document	
1	020227	Liquid. waitin	HRS Petersson		
2	020214	Thanks letter	HRS Petersson		
3					
4					
5					
6					
7					
8					
9					
10					
•Notes•	F3	•Events•			
02-01-18 22:14					

Fig 6. A new operative plan for the state on Fig. 5

F1		E U E N T S		
Events in the deal #50331650				
When		Resp	Activity	
000523	22:40	Deal	HRS	Call in
000523	22:41	Deal	HRS	Pack list
000523	22:42	Deal	HRS	Shipping
020118	20:49	Deal	HRS	Invoicing

F3 •Filter• •Unfilter• •Count• •Print• •Page 2•

02-01-18 22:17

Fig 7. Chronicle after executing the “invoicing” activity

Conceptually, dynamic planning realizes the idea of control for business process. By changing the operative plan, we may change the direction and speed of movement so that we can reach the process’s goal even in case of deviations.

4 Theory

4.1 What we were looking for

Our conceptual model of business processes was informally introduced in Section 3. This informal definition was quite enough for some practical tasks, e.g., building models of real business processes, and even designing process support systems. However, industrial application of the proposed approach, most probably, would be impossible without a rigorous formal model. Such a model should “substitute” the differential equations that are used for modeling physical processes.

Differential equations used for defining behavior of physical processes describe not only relationship between the state variables and their derivatives, but include also so-called *external variables*. The external variables are used to take into account the influence of the environment in which the process runs. When designing a formalism for describing behavior of business processes, we need to find some means to reflect the ways a business process can interact with its environment.

The most important point of communication between a business process and its environment is human beings who actively participate in the process. When looking for a suitable formalism, we formulated our task as finding a mathematical apparatus that can be used for describing computer system that works in a “symbiosis” with human beings.

4.2 Basic notions

Our model describes the dynamic world as consisting of:

- a set of *typed atoms* A ,
- a set of *objects* O ,
- a code of *laws* Λ , and

- a set of *connectors* CON , each connector hanging on a group of objects that must obey a certain law.

Atoms represent the usual notion of elementary data types, like integers, floats, strings, etc. We assume that each atom from A is assigned a type τ . By A_τ , we will denote the set of all atoms of the type τ .

Atoms are the only elements of the model that can be described independently of other notions. All other notions are interconnected. The objects are used to represent complexly structured entities. At each moment of time, an object can be characterized by its state. The state is not more than a set of connectors from CON included in the body of the given object.

Let $P(CON)$ be the power set of CON (a set of all subsets of connectors), and \perp a special symbol that represents the embryo state of the object. A function

$$st: O \rightarrow P(CON) \cup \{\perp\},$$

that assigns each object $o \in O$ its state is called a *state function*. A function w that maps the time axis T into the set of all possible state functions ST :

$$w: T \rightarrow ST$$

is called a *world*. The world determines the state $w(t)(o)$ of any object o at any point of time t . For simplicity, we consider the discrete time axis with starting point, i.e. it consists of all nonnegative integers called *time ticks*.

A *law* I is defined as a 5-tuple $\langle name, arity, type, condition, action \rangle$. The *name* assigns a name to the law. The *arity* is a positive integer $n > 0$ which defines the number of objects this particular law concerns. The law can be unary, binary, etc. The type of the form $\tau_1 \times \tau_2 \times \dots \times \tau_k$, $k \geq 0$, where τ_i is an atomic type, defines how many atoms and of which types can be taken into consideration by the law.

Let $I = \langle name, n, type, condition, action \rangle$ be a law. Consider a world w at a moment of time t . Let:

$$v = \langle v_1, \dots, v_k \rangle \text{ be a } k\text{-tuple of atoms of type } type, \text{ and}$$

$$z = \langle x_1, \dots, x_n \rangle \text{ be an } n\text{-tuple of objects from } O.$$

Then the *condition* is a predicate that given v, z, w , and t determines whether the law with parameters v holds for objects from z in world w at time t . The action provides the next state of objects from z for the points of time for which the law is violated. In other words, the action assigns a punishment for breaking the law. The simplest type of laws is a *trivial* law for which the *condition* is always true, and the *action* is always empty.

Finally, a connector is a triple $c = \langle I, v, z \rangle$, where:

$$I \text{ is an } n\text{-ary law of type } \tau_1 \times \tau_2 \times \dots \times \tau_k,$$

$$v = \langle v_1, \dots, v_k \rangle \text{ is a } k\text{-tuple of atoms such that } v_i \in A_{\tau_i}, i = 1, \dots, k, \text{ and}$$

$z = \langle x_1, \dots, x_n \rangle$ is an n -tuple of objects.

If the law is trivial, the connector just represents a passive relationship between its operands, eventually assigning some properties to the relationship with the help of its k atoms (if any). This case corresponds to the usual notion of relationship from the ER model (Chen, 1976). If the law is nontrivial, the connector forces the objects' structure and behavior to be in sync with each other and, possibly, with the properties of the relationship (if there are any atoms). This is the case of active relationship.

If a connector's law holds on its operands, we say that the connector is *satisfied*, otherwise we say that it is *unsatisfied*.

The dynamics of the objects-connectors model can be defined by a machine in which a connector is regarded as a processing unit that monitors its operands. A connector

- *awakes* when one of its operands has been changed,
- *checks* whether the law still holds by reading the *condition*,
- *restores* it by applying the *action* when the law has been broken,
- *falls asleep*.

The machine has also a central unit that resolves conflicts when several connectors demand access to the same objects.

The logical semantics of the object-connectors model is defined in details in (Bider et al., 2000). Considerable extensions to this model and its procedural semantic are presented in the thesis. For informal discussion of the model see the thesis and (Bider&Khomyakov, 2002).

4.3 Role of the human-beings

The model described in the previous sections looks quite deterministic. The determinism would be justified if when modeling the reality, we could have the exact knowledge of all laws that act in it. This is not true in practice where we always model only a part of the reality. There are two sources of non-determinism when modeling business reality:

- Firstly, the behavior of some elements of business reality cannot be fully understood or controlled because they are external for the given business, for example, customers or suppliers (see the discussion of this topic in (Wegner, 1997)).
- Secondly, even the part of business reality that could be understood and controlled might be too complex to be described deterministically from the very beginning. Full understanding, if any, of this part of reality could be reached long after the business application has been designed and introduced in the operational practice. Thus, a strategy of stepwise refinement of the model is required (for more on this topic see (Kurki-Suonio&Mikkonen, 1999)).

To express the idea of insufficient knowledge of reality, we exploit a notion of non-deterministic law. We differentiate two types of non-determinism: firing non-determinism, and action non-determinism. For a law with *firing non-determinism*, an action can be defined even for the cases where the law holds. A connector that imposes such a law may be awoken even when none of its (visible inside the

model) operands has been changed. A law with *action non-determinism* can specify two different actions for the same behavior of its operands. A connector entrusted with such a law appears to be taking different courses of action in the same situation. Of course, a law can have non-determinism of both types, firing, and action.

Connectors with non-deterministic laws, which we call *boundary connectors*, are our means of modeling interaction with the external world, e.g., people, or electronic devices. As part of the law is known, the external world cannot function arbitrarily. The known part of the non-deterministic laws constitutes a harness (in terminology of (Wegner, 1997)) on the external world behavior.

Now, the human being's role becomes clear, he/she is to help the connectors with non-deterministic laws. Such connector may be thought of as having a terminal where a human being can help the connector to do its job. Let us assume that a connector's law has action non-determinism. Then it is up to the connector in case when its operands have been changed, to request the human assistance via the terminal by beeping, blinking, etc. After providing assistance, the human being may return to his other occupations, drinking coffee (see Fig. 1), for example, until the connector calls him again. Action non-determinism may vary from allowing any possible structural changes in the connector's operands to a selection from a list of predefined choices (menu).

Now, let a connector's law have firing non-determinism. Such connector can be represented as a big button that the user presses from time to time based on observations that lie beyond the boundaries of the computer system. Then it is up to the connector to do the rest of the job.

All other situations of non-determinism may be considered as a combination of the two cases above.

5 Practice

5.1 Business process modeling

To prove that a new approach to business process modeling is valid, we need to test it in practice by creating models of real business processes. Building a model of a real business process is a challenging task because:

- Business processes are not always clearly visible as they may go through the whole, often functionally structured organization.
- Written information about business processes is often non-existing or not reliable. The only practical way to obtain reliable information for creating a model of a real business process is by interviewing the people engaged in the process.

The complex nature of the business process modeling work is well known, and a number of methodologies have been introduced to solve the task of extracting the knowledge from the experts, see, for example, (Sharp&McDermott, 2001, Huckvale&Ould, 1995). However, each of this methodologies is connected to a specific view on business processes. The state-oriented approach is not commonly used in the business process modeling practice, thus we needed to invent our own methodology for extracting information from domain experts.

The modeling work consists of a number of iterations that in a simplified form can be presented as a sequence of the following steps:

1. Get unstructured information from domain experts.

2. Process information abstracting from the uninteresting details, and make a sketch of the model.
3. Present the sketch to the domain experts, discuss it, acquire new information, and go to the step 2 for refining the model.

Abstracting from details of the current way of handling the process is the most essential part of the modeling job. The material presented back to the domain experts should not be just a paraphrase of what they told during the interviews. It should contain new for them information. Without abstraction from details, it would be impossible to suggest alternative, more practical ways of handling the process. As a result, there won't be any way to verify the correctness of the model.

When the state-oriented approach is used, abstraction from details means concentrating, first of all, on what should be achieved during the process, before going into details of in what way it is done now, or how it could be done in a more efficient way. Translating into the terms of the conceptual model from Section 3, this means that the task of building the model is concentrated on designing a structure to represent the state of the process.

Step 3 above means that the model should be presented to domain experts, and discussed with them. In many business domains, the experts are not technicians, but can be professionals of any kind: doctors, nurses, teachers, lawyers, clerks, etc. For these professional, presentation of the model in some formal language, or complex diagrammatic notation would be inappropriate. We need some means to present a process model to the experts in an understandable for them form. We solved this task by presenting them a model in form of pictures, like Fig. 2 – 7.

A business process is a dynamic phenomenon. It is not enough to discuss the state structure; we need to discuss the possible trajectories in state-space. To visualize trajectories and present them to the experts, we use a concept of *State Flow* (SF). A state flow is an ordered sequence of triples, each of which visualizes the process at a particular time, and includes:

- process's state (see Fig. 2 and 5),
- operative plan (see Fig. 4 and 6),
- chronicle of the process from the beginning up to the given point of time (see Fig. 3 and 7).

Moving through the sequence forward, e.g., from Figures 2, 4 and 3 to Figures 5, 6, and 7 gives an impression of a live process where activities executed result in state changes, new activities being planned, and the chronicle filled with new events. To facilitate building state flows for various processes we created a simple software tool *ProVis* (Process Visualizer).

Our state-oriented approach has been tested by building models for 10 business processes on commission for four organizations:

- Municipality of Motala,
- Tenants Association in West Sweden,
- Swedish Church of Gothenburg,
- Municipality of Jonköping.

A list of business process we modeled included the following:

1. Administration of decision-making.
2. Inquiries, investigations, inspections.
3. Lobbying (influencing decisions of others).
4. Processing feedback from the external world.
5. Recruiting new members for an association.
6. Rent negotiations.
7. Providing legal assistance.
8. Funeral arrangement.
9. Project administration.
10. Delivering products/services.

More detailed description of our modeling practice see in the thesis, and in (Andersson et al., 2002).

5.2 System development

We made some tests whether the state-oriented approach can serve as a basis for engineering business process control systems. In this respect our ambitions were quite modest; we never had, and still do not have resources to build our systems from scratch using low-level programming. A pragmatic approach has been adopted for completing each project with resources available. We heavily relied on using third party tools and used hard coding where we had no time to solve a problem in a general way.

Each control system we developed is structured in three conceptual layers:

1. A historical object-oriented database to store all information related to the processes, see line 2 in Table 1. Software that implements this layer is of general nature, and it is independent of a particular process control system. This layer is built upon some third party database management system.
2. A user-interface navigation system that allows the end-user to freely browse through the information that concern processes' current state, past, and future (planned activities), see line 3 in Table 1. Again, software that implements this layer is of general nature, and it is independent of a particular process control system. This layer is built upon some third party user-interface management system.
3. Application-dependent routines for executing activities and dynamic planning. This layer is implemented in a general purpose programming language.

Our first experience in building process control systems goes back to 1989-1990, when we built a system to support sales and marketing activities of a trading company. The system was called *DealDriver* to highlight that it helped the workers to “drive” their deals from the beginning (e.g., getting an order) to the end (e.g., receiving payment). This system is being used internally at IbisSoft

AB since then. The pictures used to explain our state-oriented approach to modeling business processes in Section 3 (Figures 2 – 7) represent screen dumps from *DealDriver*.

A historical object-oriented database, and a navigation system built for the *DealDriver* project formed application development tools that could be used for building other process control applications. The biggest application, *SoftMotors* was built by our colleagues from Magnificent Seven (Moscow). *SoftMotors* supports management of all operations at a car dealership. It is used by 20 car dealerships, the biggest of which has more than 20 users. This application won the “Object Applications of the Year Awards 1997” in the group “Best object-based application developed using nonobject-oriented tools” (Object World Show in London, April 1997). More information on our early experience in system development can be found in the thesis, and in (Bider, 1997a,b, Bider&Khomyakov, 1998)

Recently, we developed two new process control systems. The first one, called *ReKo*, is aimed to support recruiting of new members and personal communication with already existing member of some interest group. This system has been built on commission for Tenants Association of West Sweden and is based on the process model described in (Andersson et al., 2002).

The second system, called *Utredaren*, is aimed to support inquiries, investigations, and inspections. This system has been built on commission for the Municipality of Jönköping. Currently the system supports only one type of inquiries, the inquiry that helps to decide on suitability of a given family to provide a home for an adoptive child.

Our work on building computerized support for business process continues. A new system for processing feedback from the outside world is under development, some others, e.g. support of lobbying activities, is already in the plans.

6 Summary of results

We started with the hypothesis that “*the ideas worked out in the Mathematical system theory for modeling and controlling physical processes can be successfully used for modeling and controlling business processes.*” The main idea that we borrowed from the mathematical system theory is representation of a process as a set of trajectories in some state-space, and defining these trajectories by means of equations/inequalities.

First, we showed conceptually what kind of state-spaces can be used when defining trajectories of business processes, and suggested the way of defining sets of trajectories via activities, and specially designed rules of planning. The latter are supposed to substitute the differential equations used in physical process modeling.

We suggested also an approach to the formal definition of business process dynamics by means of a model based on three basic concepts: object, law, and connector. This model, when fully developed, can serve as a means for formal definition of business processes, which can help in engineering control systems aimed to support these processes.

Based on the suggested state-oriented approach, we created a practical methodology for modeling and analysis of business processes. This methodology helps to extract the knowledge from the human participants of a business process and represent a model in a way understandable for non-technical professionals. The methodology has been tested in a number of customer-related projects and proved to be both useful for uncovering the structure of business processes, and efficient (in terms of time needed to create a model).

Most of the modeling projects were completed by the group that included the inventor of the methodology. However, we already completed the first methodology transfer to another group. This group has already finished their first modeling project, and, as a result, has got a commission to build a computer system to support this process.

We made some tests whether the state-oriented approach can serve as a basis for engineering business process control systems. For this end, we created a pragmatic methodology of building business applications upon third party development tools. Several systems have been built based on this methodology; some of them have already been used in practice for some time, others are in the stage of introduction in business practice.

We made a deductive analysis of our modeling experience and the existing literature in order to find out in which circumstances our approach suit practical tasks better than other approaches to business process modeling. We found that the state-oriented approach is better adjusted for modeling so-called loosely-structured business processes, i.e. the processes for which it is difficult (if ever possible) to establish a predefined sequence of events. It does not mean that the state-oriented approach cannot be used for other types of business processes; just there are no practical or theoretical evidences that it might be better for other types of processes than other approaches.

When using the state-oriented approach in practice, we have not found any types of business processes where this approach cannot be applied, except in situations where the business processes had not been identified. In such cases, we used other views on business process dynamics (input/output flow, or agent-related view). However, as soon as the processes had been identified, we immediately switched to the state-oriented view.

We consider the material of this thesis as presenting enough evidence that the state-oriented approach can be useful in practice of business process modeling, analysis and support. This statement does not suggest more than the right for our approach to exist until some evidence is found that shows that it does not cover some essential properties of business processes, and, what is more important, it cannot be adjusted to cover these properties. Considering the great success of application of the mathematical system theory to controlling physical processes, we believe that chances of finding such evidences are not great. If such evidence is found, we think that it will lead to creating a new approach to business process modeling, not going back to already known ones.

As far as an ultimate proof of validness and efficiency is concerned, we do not believe that it is possible to obtain one in this particular application field. Normally, such a proof requires experimental comparison of several approaches to see which one gives better results, in terms of accuracy, efficiency, etc. Such comparison requires applying each approach in exactly the same conditions, which is impossible to achieve for business process modeling. Something will always be different: the business processes themselves, the groups of domain experts, or the level of experience of the business analysts.

Summarizing the above, we believe that for the time being, it is worthwhile to continue research and practical work on improving the state-oriented method of business process modeling. Both the theory and practical methodology require improvements that are discussed in the last chapter of the thesis.

Note that due to the size limitation we have not given here any comparison of the state-oriented approach with other approaches to business process modeling. This topic is covered in the thesis, especially in Chapter 5. See also some literature coverage in (Khomyakov&Bider, 2000).

7 References

- (Andersson et al, 2002): Andersson, T., Andersson-Ceder, A., and Bider, I. State Flow as a Way of Analyzing Business Processes – Case Studies. *Logistics Information Management*, Vol.15 (1), pp. 34-45, 2002. See also: <http://www.ibissoft.se/English/Cases.pdf>
- (Bider, 1997a): Bider, I. Developing Tool Support for Process Oriented Management. *Data Base Management* 26-01-30, p. 18, Auerbach, 1997.
- (Bider, 1997b): Bider, I. Object Driver: A Method for Analysis, Design, and Implementation of Interactive Applications. *Data Base Management*, 32-10-25, p.18, Auerbach, 1997.
- (Bider&Khomyakov, 1998): Bider, I. and Khomyakov, M. Object-Oriented Model for Representing Software Production Processes. *ECOOP'97 Workshop Reader*, LNCS 1357, pp. 319-322, Springer, 1998.
- (Bider&Khomyakov, 2002): Bider, I., Khomyakov, M. If You Wish to Change the World, Start with Yourself: An Alternative Metaphor for Objects Interaction. In: Piattini, M., Filipe, J., and Braz, J., eds. *Proceedings of ICEIS 2002 - the Fourth Conference on Enterprise Information Systems*, Vol. 2, pp. 732-742, ICEIS Press, 2002. See also: <http://www.ibissoft.se/English/tango/tango.pdf>
- (Bider et al., 2000): Bider, I., Khomyakov, M. and Pushchinsky, E. Logic of Change: Semantics of Object Systems with Active Relations. *Automated Software Engineering*, 7(1), pp. 9-37, Kluwer, 2000.
- (Deiters&Gruhn, 1998): Deiters W. and Gruhn V. Process Management in Practice. Applying the FUNSOFT Net Approach to Large-Scale Processes. *Automated Software Engineering*, 5(1), pp. 7-25, 1998.
- (Elzinga et al., 1999): Elzinga, D.J., Gulledge, T., R., Lee C.-Y., eds. *Business Process Engineering: Advancing the State of the Art*, Kluwer, 1999
- (Hammer&Champy, 1994): Hammer, M., and Champy, J. *Reengineering the Corporation – A Manifesto for Business Revolution*, Nicholas Brealey Publishing, London, 1994.
- (Kalman et al, 1969): Kalman R.E., Falb P.L., and Arbib, M.A. *Topics in Mathematical System Theory*. McGraw-Hill, 1969.
- (Khomyakov&Bider, 2000): Khomyakov M., and Bider, I. Achieving Workflow Flexibility through Taming the Chaos. *OOIS 2000 - 6th international conference on object oriented information systems*, pp.85-92, Springer, 2000. See also: <http://www.inconcept.com/JCM/August2001/bider.html>
- (Kurki-Suonio&Mikkonen, 1999): Harnessing the power of interaction. *Information k H.Jaakkola, H. Kangassalo, E. Kawaguchi eds Modelling and Knowledge Bases*, pp. 1-11, IOS Press, 1999.
- (McCormack&Johnson, 2001): McCormack, K.P., and Johnson, W.C. *Business Process Orientation: Gaining the E-Business: Competitive Advantage*. St. Lucie Press, 2001.
- (Ould, 1995): Ould, M., *Business Processes – Modelling and Analysis for Re-engineering and Improvement*, John Willey & Sons, Chichester, 1995.

(Wegner, 1997): Wegner, P. Why interaction is more powerful than algorithms. *Communication of the ACM*, 40(5), pp. 80-91, 1997.

(Workflow, 1995): Workflow Management Coalition. *Reference Model - The Workflow Reference Model*. WFMC-TC-1003, 19-Jan-95.