IbisSoft ab
We analyze and visualize your business processes
www.ibissoft.se

Tutorial on:
Business Process Modeling as a Method of Requirements Engineering

Prepared for

ICEIS 2002

Instructor: **Ilia Bider**

(Director R&D)

IbisSoft AB
Box 19567
104 32 Stockholm
Tel: 08-15 10 10

Email: info@ibissoft.se
Fax: 08-15 10 50
www.ibissoft.se

IbisSoft AB
Fabriksgatan 10
412 50 Göteborg
Tel: 031-711 74 80

Extended abstract

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. This shift consists of moving from the command-based applications to the applications of workflow and groupware type. In other words, the shift can 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, as shown in table 1.

A new generation of applications is aimed at "controlling" business processes. Designing an appropriate "control" system presumes that the nature of the process that we want to control is fully understood. That is why in table 1, the traditional data modeling is substituted by process modeling.

Aspect	**Old generation**	**New generation**
Modeling	Data Modeling	Process Modeling
Data Base	Static and passive	Dynamic (history-minded) and active
User Interface	Functional (multilevel menus)	Navigational
Organizational aspect	Follows existing management schemes	Suggests new management schemes

Table 1. Aspects of system development.

According to the most general definition, a business process is a set of partially ordered activities aimed at reaching a well-defined goal, for example:

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

Each process engages a number of participants that can be roughly divided into two categories: passive participants, and active participants. Passive participants are the participants that are consumed, produced or changed 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 those participants that perform actions aimed at the passive participants.

A business process is a complex phenomenon, and there are different methods of representing the development of the process in time. The following views on the process development are the most common:

1. Input/output flow. The focus is on passive participants that are being consumed, produced, or changed by the activities.
2. Workflow. The focus is on order of activities in time.
3. Agent-related workflow. The focus is on order in which the agents get and perform their part of work.
4. State flow. The focus is on changes produced in the part of the world that embraces the given process.

The view to take depends on many factors, one of them being the mission of the system under development as shown in table 2.

System mission	**Process view**
Integrate existing systems	Input/output flow
Facilitate coordination / communication	Agent-related view
Introduce strict order in production-like processes	Workflow
Navigate each process to its goal	State flow

Table 2. How to choose process view

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 process 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.

In many business domains, the experts are not technicians, but may 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. To have some means to present the experts with a process model in an understandable for them form is essential for the success of the modeling work.

The means of representing a model for domain experts depend on the chosen view on business process dynamic. For the input/output, agent-related and workflow views, some form of a diagrammatic presentation is normally used. For the state flow view both diagrammatic, and non-diagrammatic presentations are possible. A non-diagrammatic presentation shows examples of the trajectories of the process in state space.

Requirements on a process support system can be divided into two categories: general requirements, and specific requirements. The general requirements are the same for all processes to be supported by the system. The specific requirements express the differences between the different kinds of business processes.

The following general requirements are the most important. The support system:

- Helps in execution of activities.
- Keeps track of what has been done.
- Helps to coordinate human participants synchronously (execution of 1 activity), as well as asynchronously (execution of different activities).
- Helps to plan new activities.
- Helps in distribution of resources.
- Reminds participants - what they have to do, resource managers - where resources are needed, and managers - if there is any problem.
- Provides easy access to current state of the process, the process's history, and old processes' histories.
- Provides communication channels between the participants of the same process, external participants including (e.g., customers).
- Helps in enforcing organizational policies.
- Provides with friendly and consistent user-interface.
- Insures flexibility in individual process handling.
- Permit flexibility in respect to organizational structure (who does what).

When a support system is designed, the general requirements should be translated into a development framework. The specific for each business process requirements should be translated into dialog screens, and subroutines that helps human beings to complete activities that are assigned to them.

At least two system layers should be implement in a general way:

- A historical database to store information that concern processes, their participants, and activities along with the histories of all objects stored in the database.
- User interface navigational system that allows the end-user to freely browse throw the information that concern processes' current state, past, and future (planned activities), and execute activities.

All other parts of the system can be hard-coded.

Preface

The material of the tutorial summarizes the theoretical and practical work in the field of business process analysis and application development conducted by IbisSoft AB (Stockholm, Sweden) together with its partners: Magnificent Seven (Moscow, Russia), and Department of Computer and System Sciences of Stockholm University/Royal Institute of Technology (Stockholm, Sweden).

Some materials have originated from the discussions at the international workshops devoted to business process modeling held in 1998, and 2000: Workshop on Object-Oriented Business Process Modeling at ECOOP'98, and Workshop on Practical Business Process Modeling at CaiSE*2000.

The scope of the tutorial is requirements engineering for business (enterprise) applications. The tutorial covers the following topics:

1. Paradigm shift in business application development.
2. What is business process?
3. How to analyze, visualize and specify business processes.
4. General requirements on the software that supports business processes.
5. How to use a business process model as a specification for the system aimed at supporting the process.

The learning objective of the tutorial is to understand the problems and solutions of building a new generation of business applications. The tutorial is worked out for the audience consisted of:

- Software engineers interested in building business (enterprise) applications.

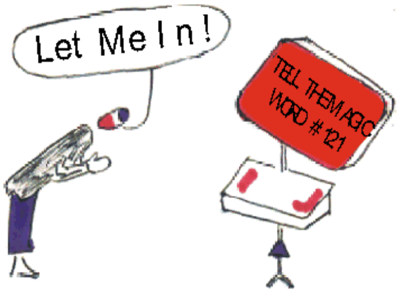
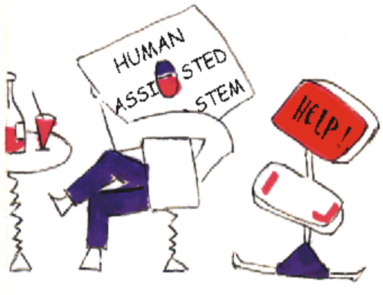
- Business analysts interested in requirements specification for business applications.

- Researchers interested in workflow, business process modeling, etc.

The material in this documented consists of two parts. The first part presents the main slides of presentation, while the second part (sections 2-7) discusses some issues in more details.

Table of Contents

Paradigm shift

Old Generation	New Generation
Human-Assisting System	Human-Assisted System
	
Powerful Toolkit	Assembly Line

Aspects of system development

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	Follow existing management schemes	Suggest new management schemes

Business process

is a set of partially ordered activities intended to reach a goal

- Reaching an agreement in business negotiations.
- Discharging a patient from the hospital in a (relatively) healthy state.
- Closing a sale.
- Building a software system according to specifications.

Figure 1

Basic notions

- Goal
- Activity
- Time
- State
- Change
- Event
- History
- Chronicle

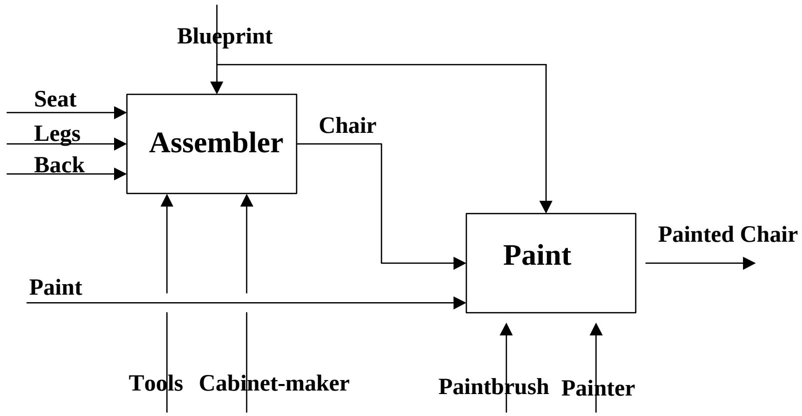
Business process

- Has participants
  - Active participants, or agents - perform activities
  - Passive participants – objects that are consumed, produced or changed when activities are performed
  - Both can be human, and nonhuman
- Develops in time – focus on dynamics

Four views on process dynamics

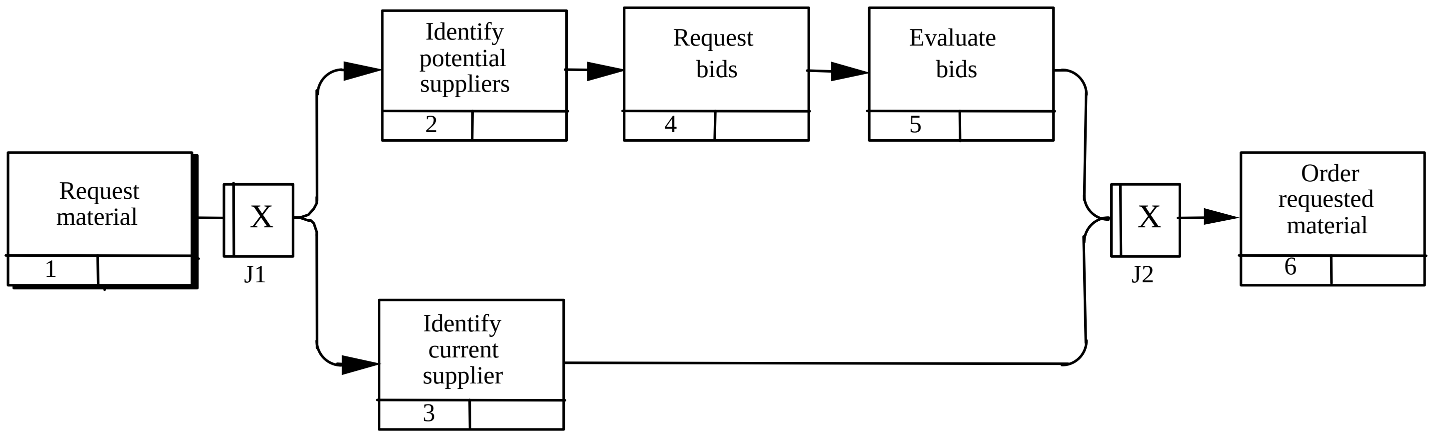
1. Input/output flow – focus on passive participants
2. Workflow – focus on order of activities
3. Agent-related workflow – focus on agents
4. State-flow – focus on changes produced by activities in the relevant part of the real world

IDEF0 –Input/Output view



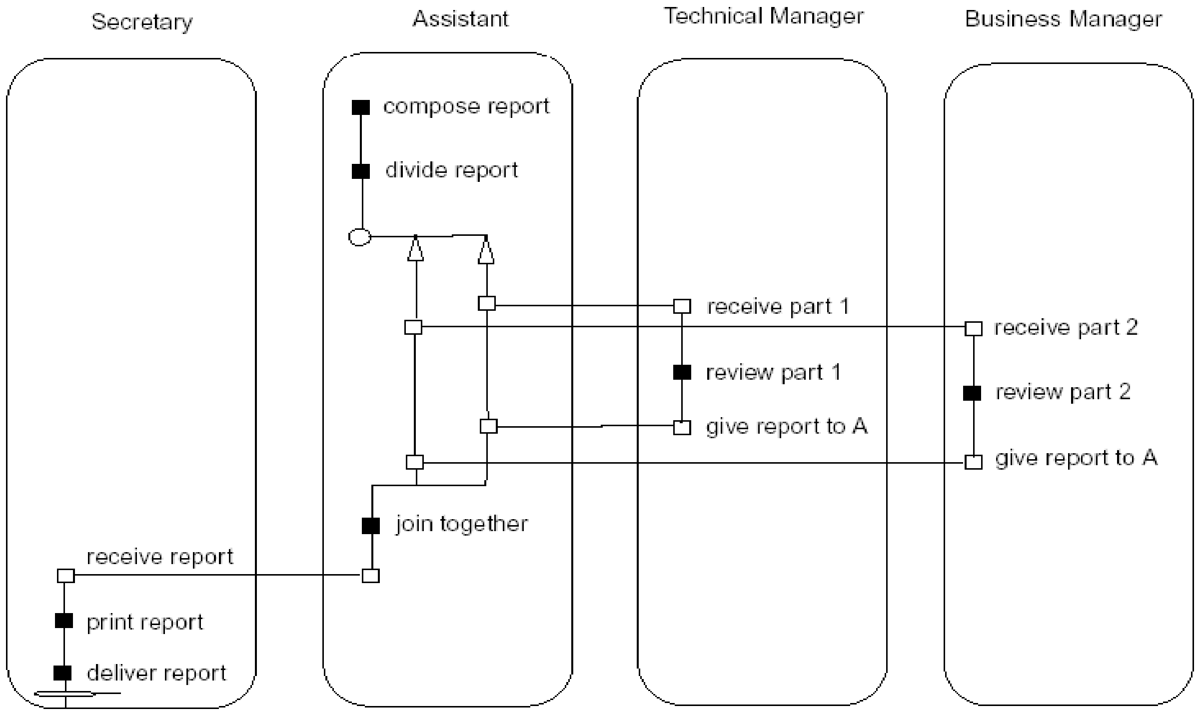
From: http://www.idef.com/Downloads/pdf/idef0.pdf
National Institute of Standards and Technology

IDEF3 - Workflow



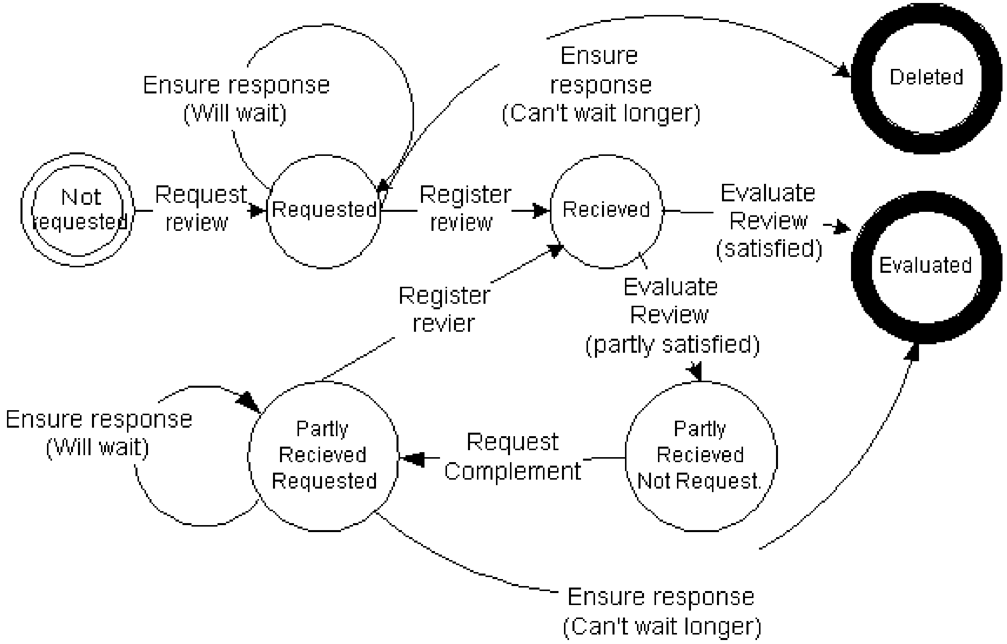
From Mayer et al.: http://www.idef.com/Downloads/pdf/Idef3_fn.pdf
KNOWLEDGE BASED SYSTEMS, Inc.

RAD- Agent-related view



From Kueng, P., and Kawalek, P. : http://www2-iiuf.unifr.ch/is/peter/KuKa96.pdf
Informatics Process Group, University of Manchester

State transition diagrams –State flow



From: Logistics Information Management 15(1), pp. 34-45

Which view to chose?

- Task to solve
  - Increasing process maturity (process identification)
  - Analysis and reengineering
  - Building computer support
- Background of human participants (technicians vs. non-technicians)
- Properties of the process

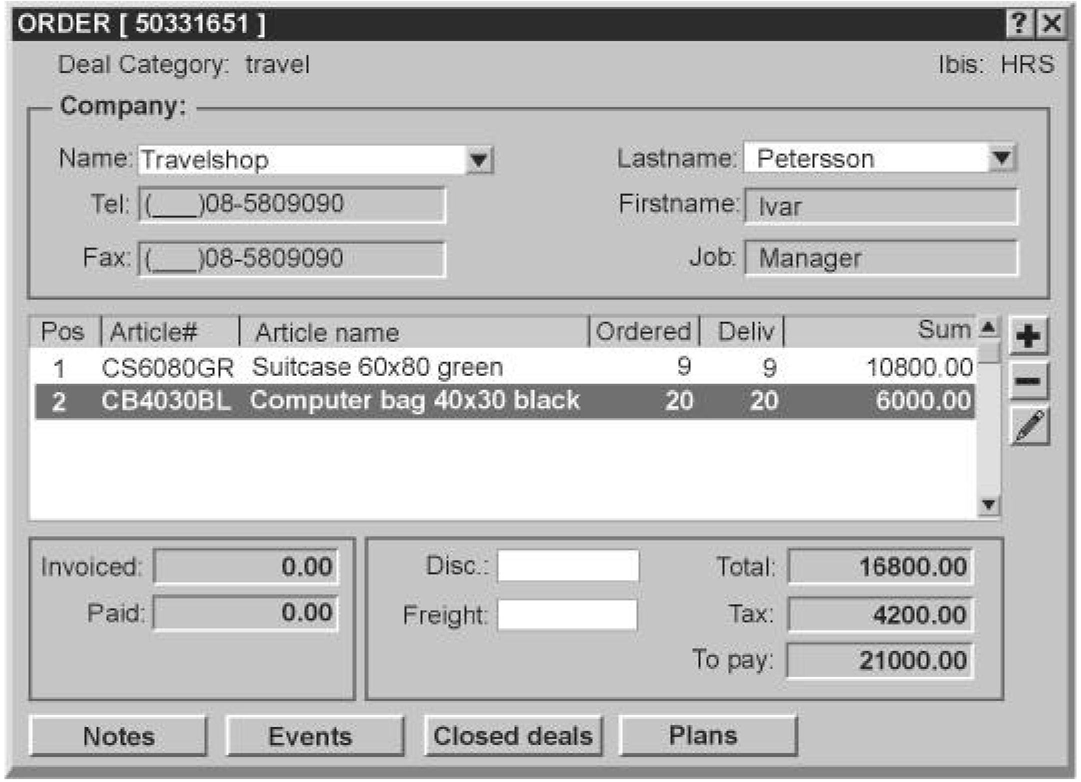
Which view to choose. Process properties

- Degree of physicalness and mobility of passive participants
- Level of specialization and mobility of active participants
- Degree of precision of operational goals
- Autonomy and friendliness of process environment
- Nature of activities
- Degree of orderliness of process flow

Which view to choose? System engineering perspective

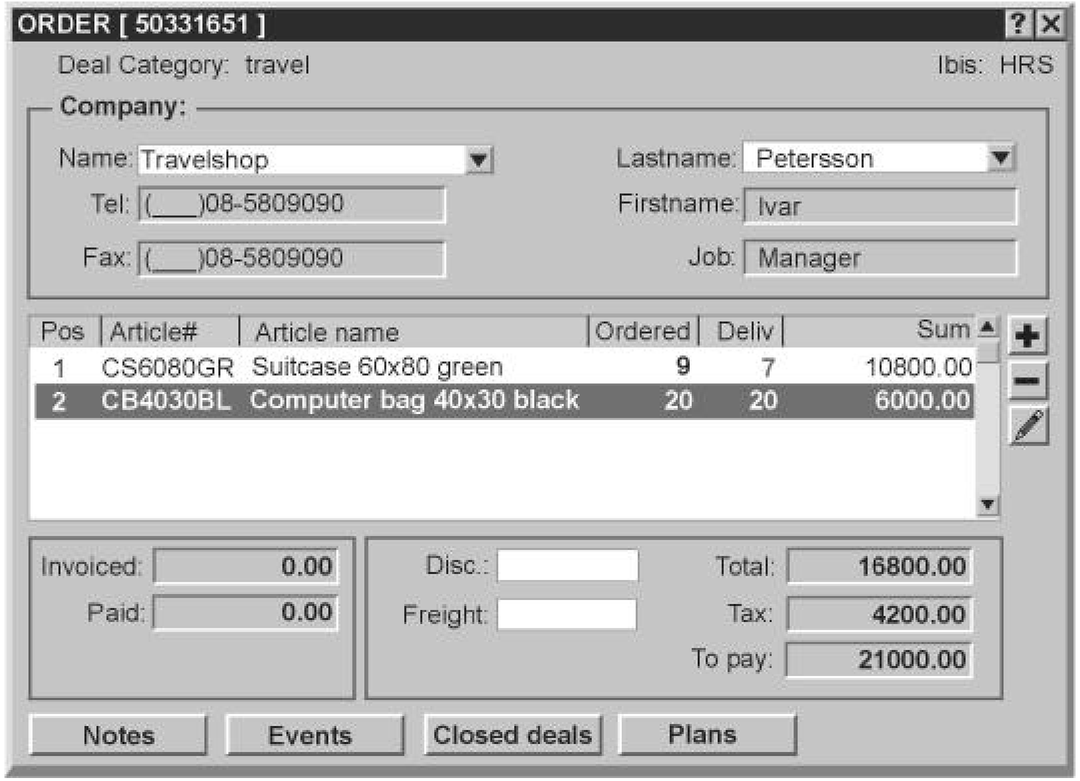
System mission	**Process view**
Integration of existing subsystems	Input/output flow
Facilitate coordination / communication	Agent-related view
Introduce strict order in production-like processes	Workflow
Navigate each process to its goal – Process Control System	State flow

State flow: state and goal



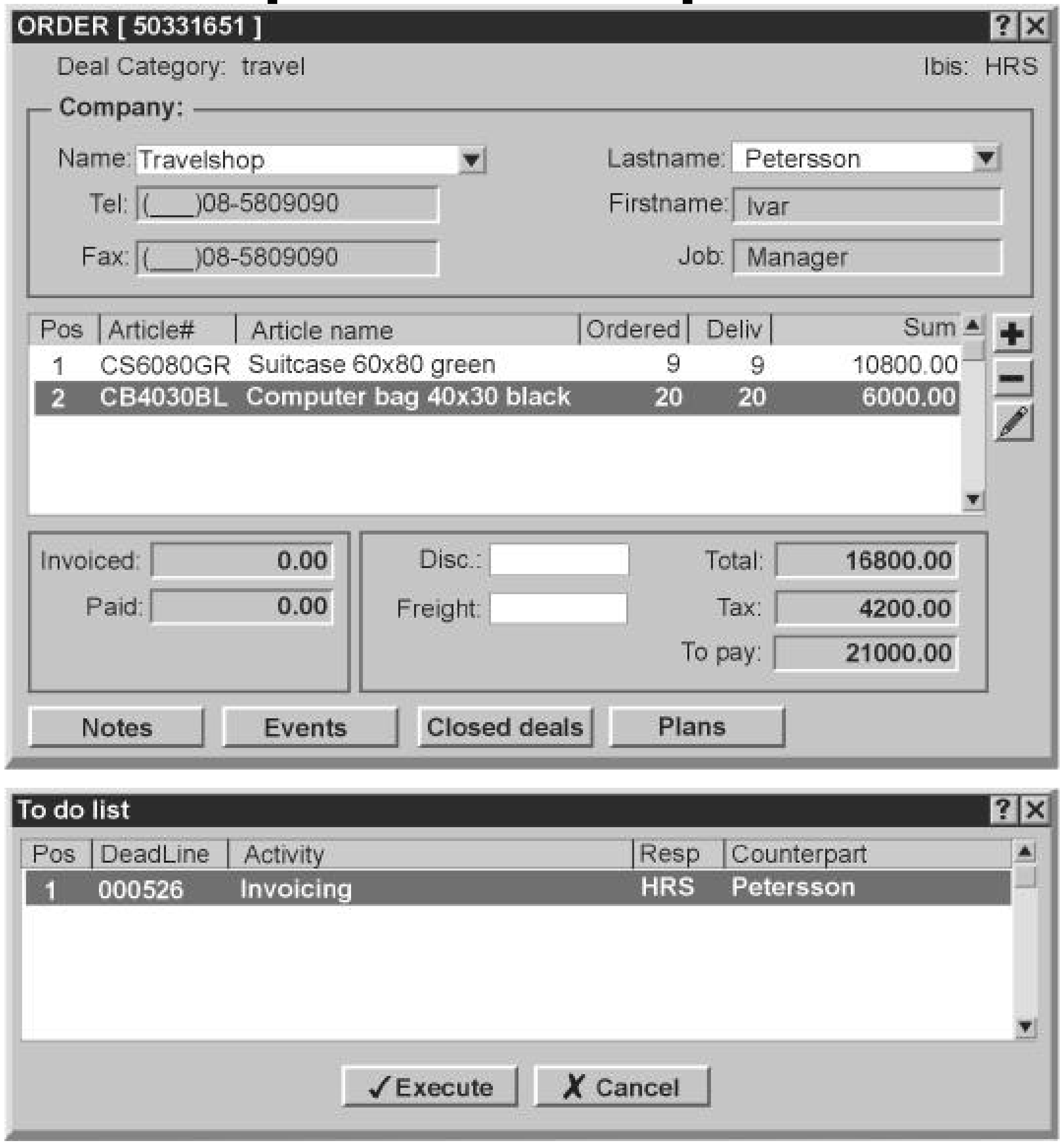
- For each item *Ordered = Delivered*
- *To pay = Total + Freight + Tax*
- *Invoiced = To pay*
- *Paid = Invoiced*

State flow: state and activities



- *Ordered > Delivered* ⇨ shipment
- *To pay > Invoiced* ⇨ invoicing

Integrated state = state + operative plan



State flow: introducing order via policies and dynamical planning

- Obligations
- Prohibitions
- Recommendations

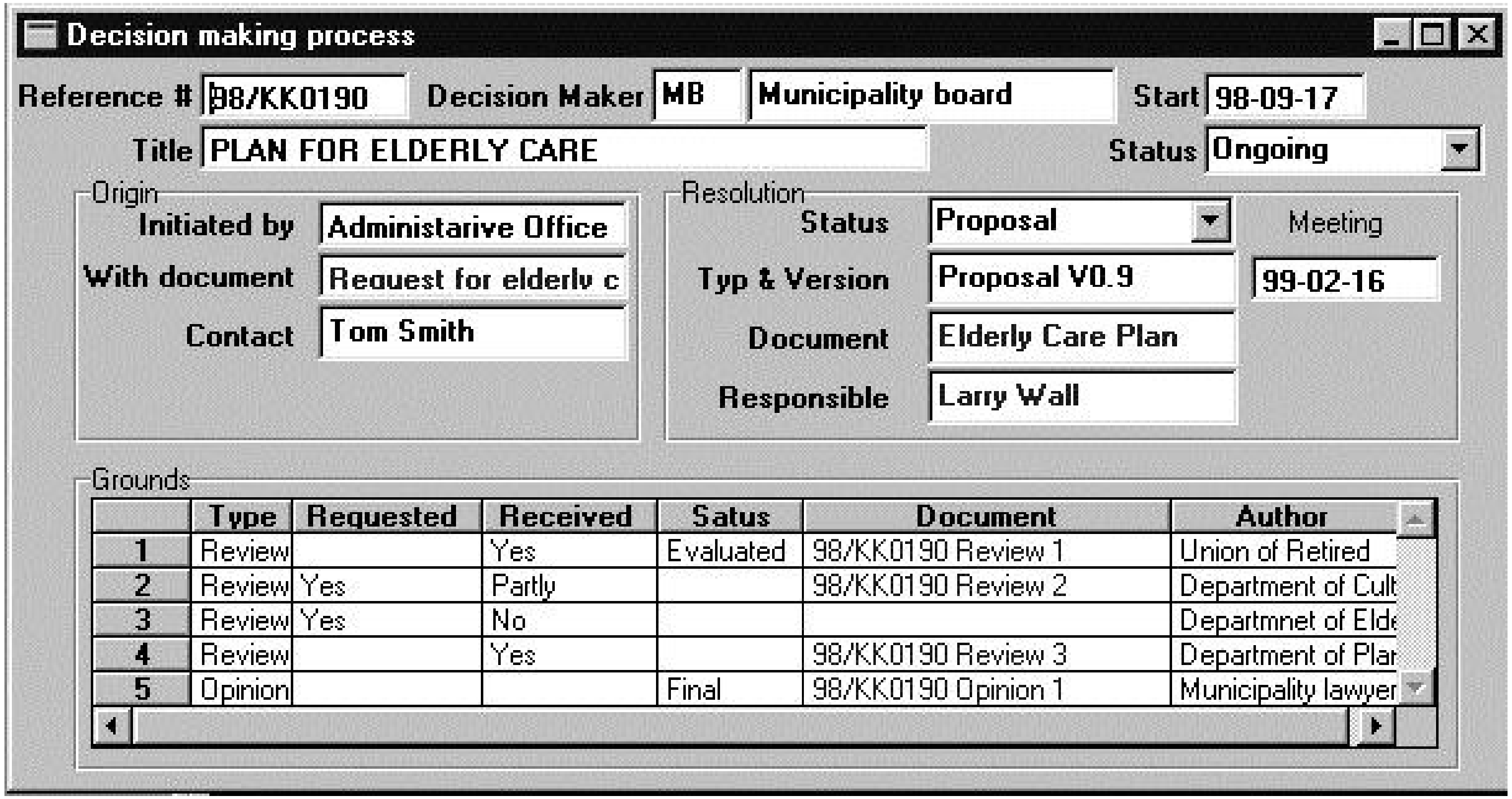
Two step planning

State flow: building business process model

- Building conceptual model
  - Defining static objects (documents, people, etc)
  - Constructing a state structure
- Creating a live demo
- Describing activities
- Giving recommendations on the support system

Example: Decision-making

Live demo – a series of triads: State, …



Decision making process

Reference # 98/KK0190 Decision Maker MB Municipality board Start 98-09-17

Title PLAN FOR ELDERLY CARE Status Ongoing

Origin

Initiated by Administarive Office

With document Request for elderly c

Contact Tom Smith

Resolution

Status Proposal

Typ & Version Proposal V0.9

Document Elderly Care Plan

Responsible Larry Wall

Meeting 99-02-16

Grounds

	Type	Requested	Received	Satus	Document	Author
1	Review		Yes	Evaluated	98/KK0190 Review 1	Union of Retired
2	Review	Yes	Partly		98/KK0190 Review 2	Department of Cult
3	Review	Yes	No			Departmnet of Elde
4	Review		Yes		98/KK0190 Review 3	Department of Plar
5	Opinion			Final	98/KK0190 Opinion 1	Municipality lawyer

From: Logistics Information Management 15(1), pp. 34-45© 2002
MCB University Press

Example: Decision-making

…, Plan, and …

Plan

Activities

	Start	Deadline	Activity	Responsible	Instructions	Document
1	98.10.22	99.01.25	Ensure response	Susan Show	Gard initial request from	Union of Retired
2	98.10.15	99.02.13	Prepare proposal	Larry Wall	Version 1.0	Elderly Care Plan
3	99.01.20	99.01.25	Ensure response	Susan Show	Gard supplement from	Department of Culture
4	99.01.10	99.01.25	Evaluate review/opi	Larry Wall		98/KK0190 Review 3
5	99.01.10	99.01.25	Evaluate review/opi	Larry Wall		98/KK0190 Opinion 1
6						
7						
8						

From: Logistics Information Management 15(1), pp. 34-45© 2002 MCB University Press

Example: Decision-making

…, and Chronicle

Chronicle

Events

	When	Event	Responsible	Comments	Document
1	98.09.17	Initiated	Susan Show		Request for elderly care plan
2	98.09.18	Grounds defined	Steve Smith		
3	98.09.22	Request sent	Susan Show		Request to Department of Culture
4	98.09.22	Request sent	Susan Show		Request to Departmnet of Elderly
5	98.09.22	Request sent	Susan Show		Request to Union of Retired
6	98.09.22	Request sent	Susan Show		Request to Department of Planing
7	98.10.15	Opinion written	Jim Smith		98/KK0190 Opinion 1
8	98.12.20	Review recieved	Susan Show		98/KK0190 Review 1
9	98.12.21	Review recieved	Susan Show		98/KK0190 Review 2
10	99.01.15	Review evaluated	Larry Wall		98/KK0190 Review 1
11	99.01.05	Review evaluated	Larry Wall		98/KK0190 Review 2
12	99.01.10	Request sent	Susan Show		Request to Department of Culture
13	99.01.10	Review recieved	Susan Show		98/KK0190 Review 3
14	99.01.11	Meeting set	Susan Smith		

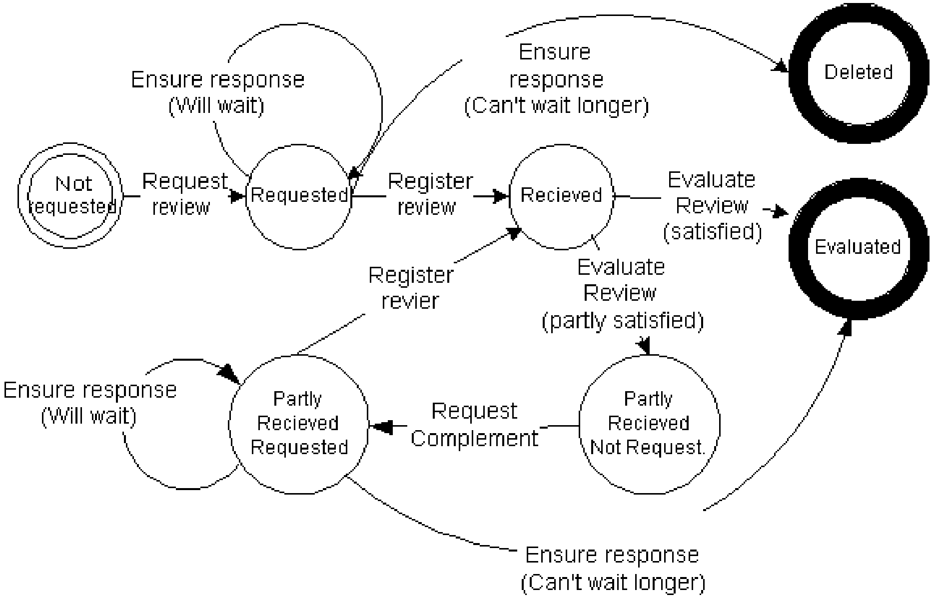
From: Logistics Information Management 15(1), pp. 34-45©
2002 MCB University Press

Describing activities

Define Grounds

Description	Define/redefine what documents should be included in the *Grounds*
Parameters	None
Instructions	Any hints on what should be included/deleted
Internal action	Modify the *Grounds* table, i.e. add/delete/modify rows. If a row is deleted, all activities connected to this row (i.e. *request review*) are deleted from the plan. Set *Status* to *Ongoing* (if *responsible* is set).
External action	None
Condition	The process should be initiated, i.e. should have a reference number and status *Initiated*. see *Initiate*, *Decide on permission*

State transition diagrams: Illustrating connection between activities via states



From: Logistics Information Management 15(1), pp. 34-45©
2002 MCB University Press

Support Systems
General Requirements:

- Functionality - What the system does
- Properties – How well it suits environment
- Quality – Tolerance to fluctuations

Functionality: support system

- Helps in execution of activities
- Keeps track of what has been done
- Helps to coordinate human participants
  - synchronously (execution of 1 activity)
  - asynchronously (execution of different act.)
- Helps to plan new activities
- Helps in distribution of resources

Functionality: support system

- Reminds
  - participants - what they have to do
  - resource managers - where resources are needed
  - managers - if there is any problem
- Provides easy access to:
  - current state of the process
  - process's history
  - old processes' histories

Functionality: support system

- Provides communication channels between:
  - participants of the same process,
  - external participants including (e.g.,customers)
- Helps in enforcing organizational policies

System properties

- Friendly and consistent user-interface
- Flexibility in individual process handling
- Flexibility in respect to organizational structure (who does what)

Quality: Reality tolerance

- We live in the fast changing world
- Changes concern both:
  - business reality, and
  - technology
- System should be adaptable to both types of changes

Quality: How to achieve?

- Have a sound conceptual model for representing business processes (redefining the processes does not change the model).
- Have a layered software architecture (layers can be changed one at a time when technology changes).
- Have the model and the software structure separated from each other as much as possible (changes in business and technology can be handled separately).

Three layers of process support system

1. Historical object-oriented database to store static objects, processes, and activities along with their histories – general layer
2. User-interface navigational system – general layer
3. Application dependent routines for activities execution and dynamic planning – can be hard-coded in the beginning

Pragmatic approach to development of control systems

- Don’t do everything yourself. Use third party tools, e.g., for storing information, and user interface.
- Build extra functionality before using the tools (the chosen tools should allow that).
- Don’t be afraid to hardcode the application-dependent parts if you can’t build them in a general way.

Analysis of examples while demonstrating the systems

- *DealDriver* – sales support
- *ReKo* – support for recruiting and communication
- *Utredaren* – support for inquiries/investigations

Additional reading

- Khomyakov M., and Bider, I. Achieving Workflow Flexibility through Taming the Chaos. *Journal of Conceptual Modeling*, August 2001: http://www.inconcept.com/JCM/
- Bider, I., Khomyakov, M. If You Wish to Change the World, Start with Yourself: An Alternative Metaphor for Objects Interaction. See *Proceedings of ICEIS 2002*
- More literature can be found on our website: www.ibissoft.com/english/index.htm

2 Why business process modeling

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. This shift consists of moving from the command-based applications to the applications of workflow and groupware type. In other words, the shift can 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.

A human-assisting system can be compared with 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.

The difference between the old and new generations is essential, and it can be traced in all aspects of system development, as shown in table 2.1

Nr	**Aspect**	**Old Generation (Human-Assisting Systems**	**New Generation (Human-Assisted Systems)**
1	**Modeling**	Data Modeling	Process Modeling
2	**Data Base**	Static and passive	Dynamic (history-minded) and active
3	**User Interface**	Functional (multilevel menus)	Navigational (free navigation in interconnected space of business objects)
4	**Organizational aspect**	Follows existing management schemes	Suggests new management schemes

Table 2.1. The difference between two generations of business applications.

A new generation of applications is aimed at "controlling" business processes. Designing an appropriate "control" system presumes that the nature of the process that we want to control is fully understood. That is why in table 2.1, the traditional data modeling is substituted by process modeling.

3 Examples of loosely-structured business processes

As *loosely-structured*, we consider business processes for which it is difficult to establish the order of activities/events. This term is introduced to differentiate this kind of processes, which are quite common in the office, from the *production-like* business processes, for which the sequence of events might be more or less predefined.

3.1 Decision making

Consider a situation with a decision-making body, e.g., a company board, or political organ, constantly making decisions prepared by some service department/organization, like the company's stab, or administrative office. In such an organizational structure the decision-making process maybe described in a uniform way independently of what kind of decisions are made by a particular decision-making body.

A decision-making process starts when a resolution is to be adopted by the decision-making body. A decision-making process usually originates by some document. This document can arrive from the external world, e.g., a motion from a political party, information on a new regulation passed by a parliament, etc. Alternatively, it can be produced internally, e.g., a record from the previous meeting. This document may be considered as an "order" for a decision-making.

Before the decision-making body can pass a resolution, all relevant basic information should be gathered, and assessed. Assessment results in a proposal for a resolution prepared by the service organization. The basis for decision-making may be considered as a collection of ground documents, e.g., documents containing external or internal opinions on the matter, calculations, laws and other regulations that should be taken into consideration, etc. At the start of the decision-making process, some of these other documents exist e.g., laws and regulations, others, e.g., internal or external opinions, are compiled during the process.

Based on the above, the decision-making process may be roughly represented in following steps: defining which ground documents are needed for a particular decision-making, acquiring each ground document, assessing ground documents, preparing proposal, passing resolution. In a particular case, the process may be much more complicated, e.g. the decision-making body decides that more ground information is needed, and it returns the proposal back to the service organization, etc.

The procedures of acquiring ground documents depend on the type of ground document in question. If it is an existing regulation, it should be found in an appropriate store (a bookshelf, a database, etc.). If it is an internal opinion on the matter, the task of writing the ground document should be assigned to an appropriate member of staff of the service organization, e.g. a lawyer. If it is an external opinion from some interested party, it should be requested from this party, etc.

The objectives of the decision-making process may be described as achieving the state of the business world in which the following two propositions are true:

- All ground documents are defined, collected, and assessed
- The proposal is prepared and passed by a decision-making body

3.2 Inquiries, Investigations, Inspections

For many types of decision-making, a special inquiry, investigation, or inspection should be conducted in order to get sufficient material for the decision. Inquiries can concern various issues, e.g.:

- Cause of an accident.
- Consequences of the previous decision.
- Probability of a given family providing a proper home for an adopted child.

Each type of inquiry may require a specialized set of measurements, calculations, interviews, and so on. However, there is a general pattern that can be applied to various types of inquiries.

A result of an inquiry is often a report with a fixed structure. It can be considered as a number of topics and/or questions that should be covered and/or answered. For many types of inquiries it is extremely important that all topics are covered (all questions are answered). Missing topics may, for example, result in losing a case in a court. Equally important is to see to that all the materials for the report are carefully gathered and analyzed.

Basically, there are three types of sources for information needed for writing a report:

- Measurements.
- Requesting and receiving ground documents, like a company's annual report, or a private person's health certificate.
- Interviewing persons relevant to the inquiry in question.

Measurements and ground documents may not give direct answers to the questions to be covered in the report, but they should be preprocessed. Two types of processing could be differentiated:

- calculation,
- review.

Calculations are based on certain formulas, or methods, which depend on the inquiry in question. The basis for calculations are values obtained through measurements, and/or found in the ground documents. Review is an intellectual process of going through the ground documents obtained and finding answers to the questions relevant to the given type of inquiries.

Any inquiry process may be divided in a number of general stages:

- Strategic planning – deciding what questions/topics should be answered/covered, what basic measurements should be taken, which ground documents should be obtained, what types of calculations should be done, how many interviews and with whom should be conducted.

- Obtaining information – making measurements, requesting and gathering ground documents, conducting interviews.

- Processing information and writing a report – making calculations, reviewing ground documents, going through the results of interviews, and at the same time refining various topics of the report.

These stages are not completed sequentially, but in parallel. For example, the ground documents could be reviewed as soon as they come, and the relevant parts of the report can be drafted before all required information has been gathered. The same is true with measurements and interviews. As soon as there is enough material to cover some topics in the report based on the results of these activities, the appropriate parts of the report can and should be written.

The completion of the stages above may be also done in a loop-wise manner. After having done some measurements, obtaining some ground materials, and conducting some interviews, a decision can be made to complete more complicated measurements, obtain more ground documents and conduct more interviews (with the same persons as before or with the different ones).

The objectives of the business process described above may be defines as achieving the state of the business world in which the following two propositions are true:

- All measurements are done and all calculations are performed

- All interviews are conducted

- All ground documents are defined, collected, and reviewed

- All questions are answered and the report is written

3.3 Lobbying – influencing decisions of others

Lobbying is an effective method of guarding rights and interests of a group in a democratic society. A lobbying process may be initiated in two ways, actively, and passively:

- Active initiation means that the interest group decides on its own to persuade, or press an authority to make a decision that lies in the interests of its members.

- Passively means that information has been received about a new decision planned, under discussion, or already taken. The decision might affect the interests of the group's members positively, or negatively, thus the interest group will oppose to or support the initiative.

Independently of active or passive initiation, a decision that the interest group wants to influence should be analyzed from several perspectives. After that, the main course of action, e.g., to promote the decision, to oppose it, or to find a compromise, could be decided on. Then, a proper strategy and tactics could be designed to pursue the chosen course.

It is impossible to choose a proper course of action without completing the consequence analysis. The proposed or taken decision should be analyzed from the point of how it would affect the members of the interest group. A number of parameters should be chosen for this end. These parameters depend on the nature of the group. They may be: standard of living, wages, taxation, etc. For each parameter, an evaluation should be made of how this parameter will be affected by the decision; whether its value would go up or down, and, if possible, how much. The evaluation should be supported by past experience, theoretical doctrines, etc. This kind of information can be very useful when trying to influence the decision in the chosen direction, create public opinion, etc.

In order to understand what channels to use when pursuing the course chosen, the decision should be analyzed from the following perspectives:

- Level of decision: international, national, municipality, company, alongside with what organ made, or is planning to make the decision.

- The area to which the decision in question is related: economy, democratic rights, etc.

- The status of decision-making: planned, under discussion, made, implemented, etc.

When the course of actions has been chosen, e.g., to oppose the decision, support from the group's members should be secured. An opinion poll might help for this end.

After the course of action has been chosen, a strategy and tactics could be designed. The *strategy* prescribes what channels to use when influencing the decision-maker(s), and in what order. The *tactics* is about how to use each channel.

Roughly, all channels to the decision-maker(s) can be divided into two groups: direct channels and indirect channels. As a direct channel, we consider a person, or a group connected to the organization that is participating in the decision-making, e.g., a member of the parliament, a high-level employee of the local authority, etc. By communicating with such a person in a chosen way (tactics), the group may affect the decision directly. Examples of tactics for direct channels are:

- inform about consequences (pretty neutral),

- ensure support, threaten, or

- look for a compromise.

Indirect channels should lead to organizations that are not involved in the decision-making, but have some influence on the decision-maker(s). The most important indirect channels are the ones that lead to mass media, e.g., daily press, weekly magazines, TV, and radio programs, etc. Examples of tactics for mass media channels are:

- inform wide public about the decision,

- initiate a debate,

- build up public opinion, and

- provoke the decision maker(s) to answer difficult questions.

Other indirect channels may lead to local, national, or international authorities that have some power to stop or promote the decision. Examples of tactics for such channels are:

- inform about the decision and its consequences,
- get support, and
- pursue to take actions.

Following the chosen for a given channel tactics means completing one or more acts of communication with the person(s) who represent this channel. Communication may be oral, e.g., meeting, press conference, phone conversation, etc., or written, e.g., fax, mail, email, etc. The basis for communication is information gathered during the analysis phase. After all communication activities via the channel have been completed, the result achieved should be understood: whether the tactical goal has been reached or not. By constant evaluation of the results, the overall strategy may be revised, new channels may be tried, or the tactics for already chosen channels may be changed.

During the communication through the channels, a lot of messages would be sent and received. It is important to keep track of such messages, and have operational procedures in place for reviewing them. This can help when writing the answers, revising the strategy, etc. If the bookkeeping works properly, the documents can also be reused, i.e., a document prepared for communication via one channel may be resent via other channels.

The objectives of he lobbying process may be defined as follows:

- The decision is classified according to the type of decision-making organ.
- Consequences are calculated according to the set of parameters that represents the interests of the group. Consequence analysis is supported by arguments based on past experience, doctrines, etc.
- The chosen course of actions is based on the results of consequence analysis.
- The support of majority of the group members has been acquired.
- The channels for influence were chosen according to the type of the decision-making organ.
- The tactics for each channel corresponds to the channel's nature.

The massages sent through each channel correspond to the tactics chosen for the channel, the nature of the channel, arguments gathered during the consequence analysis, and information received back through this channel (or maybe other channels).

4 Four views on business process dynamics

4.1 Main notions

There are many different definition of what the business process is. For example, review (Hutton, 200) lists the following definitions:

- A sequence of activities performed on one or more inputs to deliver an output to a customer.
- A set of (partially) ordered steps intended to reach some goal.
- A number of roles collaborating and interacting to achieve a goal.
- An organized collection of business behaviors that satisfies a defined business purpose and performs according to specified targets.
- A collection of business activities that create value for a customer.
- A systematic set of activities which take a 'business event' to a successful outcome.
- A way of linking people together.

We follow the most general definition of business processes, see, for example, (Hammer&Champy, 1994, Kueng&Kawalek, 1997), 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.

4.2 Classification

Reflecting the development of a process instance in time, i.e. business process dynamics, is considered to be one of the most important issues of business process modeling. There are many different approaches to representing process dynamics. However, in a simplified manner, we can classified all approaches into 4 categories according to the main view they take over the business process dynamics:

1. Input/output flow. The focus is on passive participants that are being consumed, produced, or changed by the activities. This flow can be represented as a diagram (graph), where activities serve as nodes. The arrows connect the activities in accordance to results of one activity are being used in one or another way by the next activity. Such a diagram does not reflect the order of activities directly, it reflects the causal order, i.e. the results of one activity are used by another activity. The causality establishes a partial order between activities indirectly, i.e. the results should be produced before they could be used. The most common approach to represent this kind of flow is IDEF0 (FIPS, 1993).

2. Workflow. The focus is on the order of activities in time. This flow can be represented as a diagram (graph) where arrows represent activities. Nodes show the results of one or more activities that end in a particular node. Typical notations for representing this kind of flow are IDEF3 process flow diagrams (Mayer et al., 1995), Activity Diagrams of UML (Eriksson & Penker, 2000), and Petri Nets (Aalst, 1999, Deiters&Gruhn, 1998). The Petri nets approach tries to combine the workflow with input-output flow, though the workflow is still a dominating view.

3. Agent-related view. The focus is on the order in which agents get and perform their part of work. The typical notation to represent this kind of flow is Role-Activity Diagrams – RAD (Ould, 1995), and collaboration diagrams of UML (Rumbaugh et al., 1999).

4. State flow. Each activity produces changes in the part of the real world that embraces the given process instance. Some changes may concern the state of passive participants, e.g., their form, shape, or physical location. Other changes may concern the state of active participants, e.g. the state of the mind of a human agent trying to find a solution for a complex problem. The focus of the state flow view is on changes produced in the part of the world that embraces the given process instance. When the state flow is used as a complementary view, as in IDEF3 (Mayer et al., 1995), the flow is described in form of state-transitions diagrams. A full exploitation of state flow see in (Khomyakov & Bider, 2000, Andersson et al., 2002).

The choice of the most appropriate view on the process dynamics depends on the nature of the processes being modeled, and a practical task that is going to be solved based on the model. The factors that can influence the choice of the view are discussed in the following sections.

4.3 Degree of physicalness and mobility of passive participants

In order to choose the right focus on the process dynamics, it is important to consider what kind of passive participants the process have. The passive participants can belong to one of the following two categories:

- Physical – parts of the car, a patient in the hospital, etc.

- Virtual – a decision being made, a company being reorganized (juridical, but not physical person), etc. Normally, a virtual object has some kind of physical representation. A decision being made is fixed in some document, first a proposal, then a law, regulation, order, etc. A company is represented by a number of documents that fixes its business, structure, etc. It may also have some physical office.

The degree of physicalness/virtualness determines the mobility of the passive participants. Heavy physical participants are not very mobile and need some efforts to be moved from one place to another. A virtual objects that are represented as documents are highly mobile, and with the current information technology they may be moved without any efforts.

If a process deals with immobile physical objects that should be passed around between different activities, then the input/output flow may be of great value as it can pinpoint all the logistical problems of the process. If the process deals mostly with the virtual objects, current technology may ensure that as soon as the object has been produced/changed it can be seen and used by whoever needs it. In this case, the input/output flow is of less interest. Workflow or state flow can be preferred.

4.4 Level of specialization and degree of mobility of active participants

Active participants are people and equipment that are required for completing activities. They can be classified according to two dimensions: level of specialization, and degree of mobility.

As far as specialization is concerned, the level can be from a totally specialized agent to a totally universal agent. Totally specialized agent can be used only in one activity, e.g., a person who was trained to do only one thing, or some specialized equipment. Totally universal agent can be used in any activity. A typical example of a universal human agent is an owner of a one-man company. A computer represents a universal agent of non-human kind.

If a process in question has a lot of highly specialized agents, than the agent-related workflow, which shows the flow of activities through the agents, might be of interest. It may help in optimizing the usage of specialized agents. If most of the agents are universal, then the normal workflow might be better suited to describe the process.

The degree of mobility describes how easy, far and fast the agents can travel. Here, we can have totally immobile agents, like heavy equipment, and totally mobile agents like software that can be executed on site. The mobility/immobility factor of agents should be compared with mobility/immobility factor of passive participants. For example, if all passive participants are mobile, than the mobility of the agents is not important. Consider another example where some agents are highly specialized and immobile, and passive participant that the agents consume, produce, or change are physical but relatively mobile. Than both agent-related workflow-view and input/output view are of interest when analyzing such a process.

4.5 Degree of precision of operational goals

The main parameter to classify the operational goals is the level of precision. The operational goal may be very precise, like manufacture a new car of make Volvo V-70 with such and such equipment, or less precise, like accept a plan for elderly care for year 2002. Less precise goals are specified in a functional manner. Examples:

- Accept a plan for elderly care for year 2002 with a total budget of 1 000 000 euro, and main directions as described in law no 789 passed by the parliament on 5 January 2002.

- Create a software system that supports the decision-making process in the municipality of NN. The maximum budget is 100 000 euro, the deadline is 31 December 2002.

The functional specification does not precisely define the result that the process is supposed to produce. It only states constraints that should be observed when delivering the result.

Even when an operational goal is defined functionally, there often exists a projected specification of what the result of the process should look like. This may take a form of proposals for the decision, or system specifications. The projected result may change in time, and sometimes the result may differ very much from the first projection. The degree of precision for functionally specified operational goals points to how much and how often the end-result can deviate from the initial projection.

When describing processes with functionally specified goals, a special consideration should be given to definition of operational goals in form of constraints. Very few methods of process representation give a clue of how to do it in a structural way. The state-oriented method described in this thesis presents a possibility for defining operational goals in form of constraints on the final states.

The degree of precision of operational goals has correlation with the nature of activities, and process flow, see sections 4.7, and 4.9. The more precise is the operational goal, the more exact the activities of reaching it could be defined. The same is true for the process flow.

4.6 Autonomy and characteristics of the process's environment

Each process interacts with its environment that is beyond the total control from the process. If the level of interaction is low, we speak of highly autonomous internal processes, for example, training personnel, pure manufacturing, etc. The flow of activities in this kind of processes normally follows the internal logic of the process, and it does not depend much on external events. The workflow approach to the description of the process dynamics will suit this kind of processes very well.

Processes that include external actors, like customers, supplies, etc., have less degree of autonomy. The order of activities for these processes depends not only on the internal logic of the process but also on the external, not always predictable events. For non-autonomous processes, it is important to understand if the environment in which they operate has a friendly collaborative nature, or a less friendly competitive nature.

In the collaborative environment, it is normal that external actors do their part of job as expected. For example, if during the purchasing process the vendor is asked to send a product description, we expect that he/she would do it with pleasure. A maximum disaster that can happen, he/she forgets, and will need a reminder. A process that operates in a collaborative environment can behave like an autonomous process, i.e., the order of activities performed follows the internal logic of the process.

Typical examples of processes that operate in a competitive environment are sales in the presence of competitors, lobbying, negotiations, etc. The activities to be completed in this kind of processes may, at large extent, depend on how the environment is being changed by the actors whose actions are outside the control of the process. The processes of this type are sometimes called event-driven. The workflow view is not very helpful in this case. A state flow approach suits better the task of modeling the event-driven business processes.

4.7 Nature of activities

Some activities may be described in the exact manner, i.e. as an order of simple operations. For example, activity *attach a wheel to the car being assembled* can be presented in simple operations, like *screw a nut*. Most of the manufacturing activities are of this kind.

Other activities may be described only from the perspective of what results they should produce. For example, a description of activity *review a document* can state that the result should be in form of comments, but it is impossible to dictate exactly how and on what basis the comments should be produced. Many of intellectual activities have non-exact nature.

Activities that can be described in the exact manner have an advantage that the time and other resources consumed by an activity can be easily established. These parameters belong to the activity description. For activities that cannot be defined in the exact manner, such parameters may only be estimated on the case-to-case basis. The estimation belongs not to the activity definition, but to the process instance for which the estimation is being made.

For processes with exactly defined activities, it is possible to make simulations of the type “what happens if we cut the time consumed by a particular activity by 30%”. For this kind of simulations, the workflow view on the process dynamics may be very useful.

For processes with many non-exactly defined activities, this kind of simulation is practically impossible. Thus, the advantages of time and resource related simulations are not of much importance for this kind of processes.

4.8 Orderliness of process flow

When choosing an appropriate view on the process dynamics, it is important to understand if the activities in the process flow follow each other in some exact predefined order, or only partial order can be established. The degree of orderliness correlates very much with other factors described in the previous sections, such as the precision of operational goals, degree of autonomy, friendliness of environment, nature of activities, etc.

For example, a highly autonomous process with a precise operational goal and exactly defined activities will have a high degree of orderliness. Such kind of processes can be properly represented by workflow diagrams. Suppose that in another example we have a non-autonomous process (like “sales”) operating in a competitive environment with a functionally defined goal (like “sell something to X”) and with non-exactly defined activities (like “analyze customer needs”). The degree of orderliness for such a process will be quite low, and it would be very difficult to present its dynamics with workflow diagrams. A state flow approach is more suitable for modeling such kind of processes.

4.9 Level of process maturity in the organization

The level of process maturity (McCormack&Johnson, 2001) characterizes the amount of knowledge an organization have about its own business processes. The following questions can help to determine the level of maturity:

- Are the processes identified?
- Have strategic and tactical goals been established for each process?

- Are personnel aware of in which processes they participate?

If the organization has a high level of process maturity, the process modeling can be done on the process-by-process basis. For each process, a group of experts who participate in the process can be assigned to investigate the details of workflow, state flow, etc.

If the level of process maturity is low, i.e. processes have not been identified yet, the first job is to find them in a functionally structured organization. One way to do this job might be by going through the organization on the department-by-department basis (and may be even on person-by-person basis). In this case, an input/output approach can be quite useful:

- First, identify what activities are performed in a particular department, wherefrom the input objects come, and where to the results are delivered.
- Sew together activities through their input/outputs.
- Select processes from the resulting cobweb.

For organizations that strictly define responsibilities for each position, the agent-related view may suit the task of identifying business processes. The identification process may start from listing activities for each position/role. Then points of cooperation/communication can be established, and one or several processes can be identified from the resulting net.

4.10 Professional background of human participants

There are two ways to ensure that a process model corresponds to what is going on in the real world. One is to observe the part of the world related to the process in question in real-time. The other one is to discuss the process with the people engaged in it, read operational manuals, etc. The first method is quite expensive (in terms of time), and it is possible to use it only with the processes that produce physical results (e.g., manufacturing). For most of the processes that include intellectual tasks, like design, decision-making, etc., only the second method can be applied.

When the second method is applied, only the people engaged in the process can give a confirmation that a process model corresponds to what happens in reality. Therefore, the process description should be understood by the majority of people who are engaged in the process. Confirmation from the management only is not enough, as the management may not know all details of the process. As far as operational manuals are concerned, they may be out of date, and thus they cannot be considered as a reliable source of confirmation.

The way of presenting a process should be chosen in accordance to what kind of people are engaged in the process, their background and current assignments. If the process team consists mainly of the people with technical background, e.g., engineers, system developers, etc., it is possible to use highly formalized notations, complicated diagrams, etc. These kind of people use formal definitions in their normal everyday work; thus, for them, there won’t be any problems to understand a formal description of business processes and find out what is incorrect.

If on the other hand, the people engaged in the process in question include specialists of non-technical professions, like office workers, doctors, nurses, lawyers, etc., the use of formal notations should be limited. Simple diagrams will help to understand the matter, but complicated many-pages diagrams are not suitable for this kind of professionals. They won't be able to detect errors.

The input/output flow, workflow, and agent related view, all use diagrammatic notation for presenting the model. To the best of our knowledge, there are no non-diagrammatic approaches related to these views. When working with non-technicians on modeling a complex business process, the diagrammatic presentation can create additional hinder in communication between business analysts and experts in the business domain.

When using the state-oriented approach, the focus of discussions is placed on the state structure. The state structure can be expressed not only as a formal structure, but also as a two-dimensional picture (see slides). As far as our own practice is concerned, the state pictures are quite understandable for not-technical professionals. If a diagrammatic presentation may create an obstacle in communication with non-technicians, choosing the state flow view may help in eliminating this obstacle.

4.11 Intended use of business process model

The choice of modeling methods depends also on objectives an organization have with the process-modeling project. Below, we present a list of usual objectives (the list is not full):

1. To increase the level of process maturity, for example, to make the staff goal and process-conscious, to improve cooperation between colleagues, to educate new employers, etc.

2. Create a basis for process analysis and reengineering.

3. Create a basis for building computer support systems.

4.11.1 Increasing process maturity

If increasing the process maturity is on the list of objectives, then we need to choose modeling methods that could help in communicating the process knowledge to all participants of the process. Here, it is particularly important to consider the background of the participants: technical/non-technical, and choose less formal methods for non-technicians.

4.11.2 Analysis and reeingineering

If analysis and reengineering is on the list of objectives, then first, we need to understand what kind of analysis should/could be completed for a given process. Two types of analysis can be applied to a business process:

- quantitative, or performance analysis,

- qualitative, or structural analysis.

The quantitative analysis means evaluation of the process based on numerical values of important parameters, like rate of success, calendar time required for reaching the goal, costs in time and money, number of activities performed, customer satisfaction, etc. A typical example of this kind of analysis is ABC (Activity-Based Costing).

The quantitative analysis requires statistically reliable information on the process activities. Often, this information exists only for the processes with exactly defined activities. For such activities it is possible to establish how much time, material, manpower, etc. is needed for their execution. Based on this information, it is possible to calculate the costs of the process instant, or the needs for some specialized agents, human or not human. For the first task, a workflow view on the process may be useful. For the second task, an agent-related view may be of great use.

For the processes that have a great deal of non-exactly defined activities, it is difficult to make the quantitative analysis based only on the process model. The model itself won't provide the statistically reliable information, see deliberation on the topic in section 4.7. This kind of information is difficult to obtain without first introducing some kind of computer support for identified processes.

Qualitative or structural analysis means evaluation of the process based on matching the activities included in it against the goal of the process, i.e. whether they contribute to the goal and in what way. The qualitative analysis can be performed based on a model of the business process. This type of analysis requires detailed representation of the process goal. For relatively complicated goals, the state flow view on the process dynamics may be the only option.

Process reengineering means suggesting a better way of handling the processes, for example, streamlining the process by eliminating activities that do not contribute to the goal. Results from both quantitative, and qualitative analysis can be used for reengineering. The results from the qualitative analysis can be used to detect the activities that do not lead to the goal (do not add value in other terminologies). Such activities can be eliminated. Another example is rearranging the order of activities execution in such a way that the activities directed at independent sub-goals are executed in parallel. This type of reengineering can be done based on the workflow view for the processes with high degree of orderliness, or based on the state flow view for the processes with low degree of orderliness (see section 4.8).

4.11.3 Building computer support

If building computer support is on the list of objectives, the following should be understood:

4. If a new system is to be created, or the existing systems are to be integrated?

5. If the system will support only execution of activities, or it will help in running the process. The latter includes: help in keeping track of what has been done, planning new activities, reminding what has to be done, etc.

6. If a new system will impose a strict order in activities execution, division of responsibilities, etc., or it will allow to choose the course of action dependent on how the process is developing in time.

If an integration project is in view, then the input/output flow may help to understand how to connect various systems used in the organization. If the system should impose strict rules, the workflow view, or even agent-related view could be very helpful in building such a system. If the system should allow a high-level degree of flexibility, then the state flow approach can be the most appropriate. The result of deliberation above is summarized in the table 4.1.

System mission	**Process view**
Integrate existing systems	Input/output flow
Facilitate coordination / communication	Agent-related view
Introduce strict order in production-like processes	Workflow
Navigate each process to its goal	State flow

Table 4.1. How to choose a view on process dynamics.

4.12 Which view is the best?

Based on the discussions in the previous sections, we can conclude that there is no universal approach that would be equally suitable for all types of business processes, all types of objectives, and all levels of process maturity. An organization might need to choose several different methods to apply at different stages of the process modeling work, and for different objectives.

As we have seen in the previous sections, many factors should be taken into consideration when choosing suitable methods. Summarizing the discussion, we can give some general recommendations of what methods to choose.

- If an organization is functionally structured and processes are not identified, it is suitable to use input/output view (for example, IDEF0), or agent-related view (for example, RAD).

  Input/output view suit mostly organizations that have formal ways of internal communications via some objects, like documents, files, etc. Then the processes can be discovered by following the movement of these objects inside the organization.

  Agent-related view suit mostly organizations that strictly define responsibilities for each position. The communication channels may be informal, like phone calls, informal meetings, etc. Then the identification process may start from listing activities for each role.

- When the processes are identified the workflow view, or state-oriented view, or both should be applied, as they are better suited for expressing details of each process. Workflow can be used only if there is some normal order in which activities are completed one after another. If this order is difficult to establish, the state-oriented view should be used.

  In which way the end-result should be documented, depends on the tasks in which this result will be used, e.g., analysis and reengineering, building computer support, etc. Several views may be needed, each one for its own use.

- There is no need to work sequentially, i.e. first get a full input/output view, then identify the processes, than describe each process. As soon as (after initial analysis) some process has been suspected, a different method can be applied to map this process. The work with input/output view can continue to identify other processes. We just span a new process-mapping project, and continue with identification.

The same is true when modeling a particular process. Suppose, for example, we want to represent results in a state-oriented way, but it is easier to start with workflow diagramming. We do not need to completely finish the workflow description before going over to the state-oriented description. As soon as we have enough information to make a sketch of the process state structure, we can continue processes mapping using the state-oriented approach.

4.13 References

(Aalst, 1999): Aalst, W.M.P. van der. *How to Handle Dynamic Change and Capture Management Information? An Approach Based on Generic Workflow Models*. http://wwwis.win.tue.nl/~wsinwa/genwf.ps, 1999.

(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.

(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

(Eriksson & M.Penker, 2000): Eriksson, H.E., and M.Penker, M. *Business Modeling with UML, Business process at work*, John Wiley&Sons, 2000.

(FIPS, 1993): *Standard for Integration Definition for Function Modeling* (*IDEF0*), National Institute of Standards and Technology (NIST), FIPS publication, 183, December 1993.

(Hammer&Champy, 1994): Hammer, M., and Champy, J. *Reengineering the Corporation – A Manifesto for Business Revolution*, Nicholas Brealey Publishing, London, 1994.

(Hutton, 2000): Hutton G., *Behavioural* (*Process*) *Modelling Techniques For Organisational Design.* Enix Consulting Limited, 2000: http://www.enix.co.uk/behmod.htm.

(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.

(Kueng&Kawalek, 1997): Kueng, P., and Kawalek, P. (1997), Goal-Based Business Process Models: Creation and Evaluation. *Business Process Management Journal*, 3(1), pp.17-38, 1997.

(Mayer et al., 1995): Mayer, R.J., Painter, M.K., Menzel, C.P., Perakath, deWitte, B.P.S., Blinn, T. *Information integration for concurrent engineering* (*IICE*) *IDEF3 process description capture method report.* KNOWLEDGE BASED SYSTEMS, INCORPORATED, http://www.idef.com, 1995.

(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.

(Rumbaugh et al., 1999): Rumbaugh, J., Jacobson, I., Booch, G. *The unified modeling language reference manual,* Addison Wesley Longman Inc., 1999.

5 The state flow view in more details

5.1 Main idea

The general definition of the business process given in the previous section is based on the notion of goal. This notion presumes that at any moment of the process's life, we can tell whether the process’s goal is achieved or not. If it is not achieved, we would like to be able to tell how far is it from the process’s goal. This leads us to the notion of the process’s *state*. The state can be *final*, i.e. the goal has been reached, or *intermediate*, i.e. the goal has not been reached yet. As an example, an intermediate state of order processing in a retail store is shown on fig. 5.1.

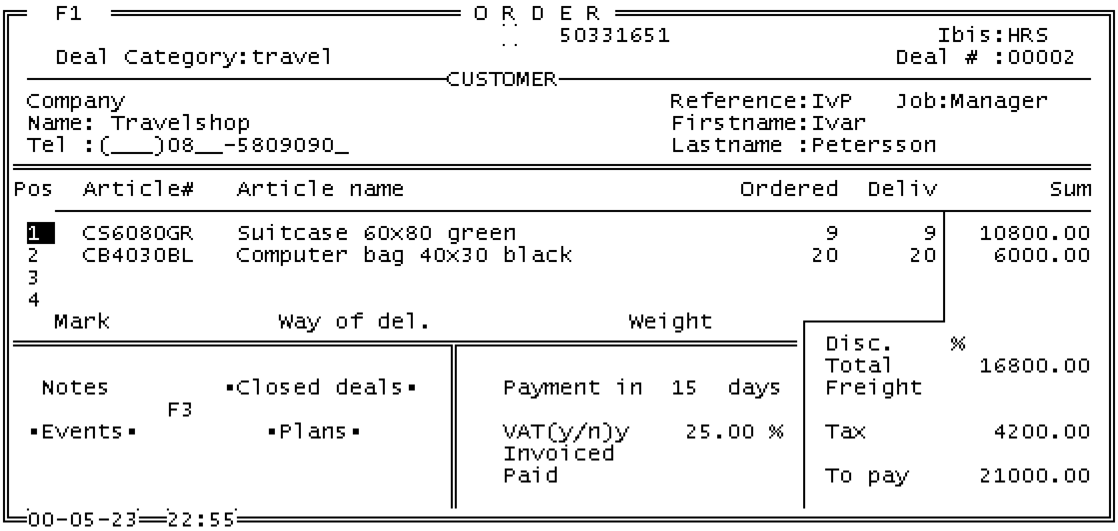


Fig. 5.1 Example of a state representation

The notion of state allows us to consider a business process as a dynamical system that moves in the space of all possible states until it reaches the final state (the goal). The notion of state and trajectory in the state space are basic in the mathematical system theory that is aimed at modeling physical processes. In physical processes the movement in the state space is defined by differential equations for continues systems, state-transition diagrams for discrete systems, or a combination of both in hybrid systems.

In a business process, movement forward (to the goal) is done via activities execution, e.g. ship the goods. However, activities are not the only possible source of movement. An unpredictable external event can move the process backwards or sidewards in the state space, e.g. the customer changing his order.

The presence of external influence makes it impossible to define the process’s goal just as a point in the process's state. The goal should rather be defined as a set of final states with a criterion of which final states are reachable, and which ones can be considered as nearest to the given intermediate state.

5.2 Imposing restrictions

In the mathematical system theory, the concept of state has been introduced to reduce the need of considering the history of inputs when calculating the outputs. When the concept of state is used, the input changes the state, and the output is fully determined by the new state.

The same principle can be applied to the business processes. The current state of a process normally contains enough information to guess what activities need to be executed in order to move the process forward. Consider, for example, the state of order processing on fig. 1. The set of final states for this process can be defined as follows:

- For each ordered item *Ordered = Delivered*
- *To pay = Total + Freight + Tax*
- *Invoiced = To pay*
- *Paid = Invoiced*

The set of activities that should be executed depends on how the current state differs from the projected final one. Examples:

- If for some item *Ordered* > *Delivered*, shipment should be performed.
- If, on the other hand, *Ordered* < *Delivered*, steps should be undertaken to get back excessive items.
- 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, etc.

Execution of each activity changes the state of the process, moving it nearer to the final state. For example, when shipping is completed, *Ordered* may become equal to *Delivered*, at least for some items. Obviously, the activities required for reaching the goal can't be executed in an arbitrary order. Some way of imposing restrictions is required.

In the real business life, activities are planned first and executed later. Planning is needed for many reasons, e.g.:

- to reserve resources that are required for execution of the activity,
- to ensure the flow of work between various departments of the organization,
- to pay attention of the individual workers what they are supposed to do, etc.

Planning can also be used as a tool of execution control. Let us for a particular process's state create a list of activities that *should* and are *allowed to* be executed in this state. Obviously, this list can't include two activities such that the presence or/and execution of one of them is based on the outcome of the other. The list constitutes an operative plan of the current process.

As soon as an activity from the process's plan has been completed, new activities can be planned based on the new process's state. Thus, the restriction on the order of execution can be defined as the rules of *dynamic planning*. To formulate these rules, we use the following method. We regard the process's plan as being an integral part of the process's state (more precisely, state representation). This allows us to define a notion of *valid* state in addition to the notion of final state. To be valid, the state should include all activities *required* and *allowed* for moving the process to the next stipulated state. See, for example, plan on fig. 5.2 that complements the process state from fig. 5.1 and makes it valid.

	DeadLine	Activity	Resp	Counterpart
1	000526	Invoicing	HRS	Petersson
2				
3				
4				
5				

Fig 5.2. Operative plan

Naturally, all final states with the empty plan belong to the set of valid states. All intermediate states with the empty plan do not belong to the set of valid states, as well as any final state with a nonempty plan. Treating the plan as a part of the process's state has the following advantage, changing the plan becomes a normal operation of changing the process's state during the completion of an activity. Thus, any regular activity, which removes itself from the currents state as its last step, no longer differs (at least conceptually) from the "planning" activity that just changes the plan.

Based on the notion of valid state, the order of execution can be defined in the form of rules that given an invalid state, correct it in a way that it becomes valid. The state is corrected by changing the plan, i.e. via adding and/or removing activities. The correction can be done by one general rule that observes the whole complex state, or with the help of many local rules, each of which watches a limited part of the state structure. The last option allows introducing the notion of sub-process, but in a *declarative* way.

5.3 Acquiring controlled flexibility

Dependent on what kinds of rules are introduced for controlling the processes, we can get everything from predefined control to almost full chaos. The latter occurs when all activities are planned manually, the rules requiring not more than "something is planned" when the process is not in a final state. If the plan is empty, the general activity *plan something* is added.

To get the order while allowing a certain degree of deviation, we need to structure the rules of planning in some way. We choose to structure the rules according to the idea of policies. Policies are usually divided into three groups: obligations, prohibitions, and permissions. Policies can govern various aspects of the business process. For example, a policy that defines who has permission and/or obligations to complete a particular type of activities is related to the field of resource management. In this paper, we are focused on execution control, and thus we are interested in the policies that regulate this part of the process's life. Examples of control policies are as follows:

- No sale can be closed without the product being first demonstrated to the customer (to ensure the customer knows what he is buying).

- The product is not demonstrated to the same customer more than twice (as it is too expensive relatively to the cost of the product).

Applying the concept of policies to our idea of dynamic planning, we group the rules of planning into three categories. The first two run as usual:

1. *Obligations*. Based on the current state and possibly the process's history some activities must be present in the process's plan. In case of absence, they are added.

2. *Prohibitions*. Based on the current state and possibly the process's history some activities can't be present in the process's plan. In case of presence, they are removed.

In respect to execution control, the concept of permission does not make much sense. Everything that is not dictated by obligations and is not prevented by prohibitions is permitted. However, another type of policies could be useful here, i.e.:

3. *Recommendations*. Based on the current state and possibly the process's history some activities are normally present in the process's plan. In case of absence, they are suggested (strongly or weekly) for inclusion.

Classification of rules into 3 groups above implies a two steps scheme of planning after executing an activity. First, the state is corrected automatically using all rules, the suggested activities being marked in a special way. Then, the responsible person may change the resulting plan, provided that he/she is prevented from breaking any obligation or prohibition.

The groups of rules described above give a hint on how gradual tuning of a process support system can be done. We can start with no rules, totally relying on manual planning. On the next stage, when the nature of the process is better understood, recommendations are added. As the last step, some recommendations are promoted to obligations and some prohibitions are added. It is very important to have a strategy of gradual tuning as it is practically impossible to acquire all information on business processes before the system has been installed.

6 Modeling business processes - challenges to meet

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 environment in which a business process modeler works is similar to the environment in which a field linguist works when studying a natural language that exists only in the spoken form. He gets oral information from the native speakers, and confirms his/her theories via getting confirmation from the native speakers.

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 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.

A typical business process analysis project starts with building a project group. The group usually consists of two business analysts (BA), and from 3 to 7 field experts. Field experts should represent all kinds of human participants of the given business process; just having managers is not enough as the managers may not know all details of the process. There are no prerequisites that the business analysts should be experts in the given business domain, they should be just experts in business analysis. There are no prerequisites that the experts in the field should know anything about modeling or business processes. They may not be aware that they participate in any business process at all.

Sometimes BA's can acquire some knowledge on the business domain in question through receiving internal documents from the customer. However, such documents are considered as not reliable. Besides, they are often written in a professional jargon, which leave outsiders no possibility to get insights into the business process before meeting the experts in the field.

The role of the experts in the field is to provide information, and verify the suggested business model. The role of BA's is to initiate discussions, ask right questions and provide the drafts of the model for discussions.

7 Building computer support

7.1 Functionality, properties, quality

Any business application, or in fact, any software system can be evaluated from the three major aspects: functionality, properties, and quality. Functionality determines the usefulness of the system to the end-user in terms of what kind of help it provides. Properties determine how well it suits the given business and technical environment. There are many definitions of quality of software systems, but we believe that for a business application, the most important aspect of quality is the system's tolerance to changes in business environment and computer technology

7.1.1 Functionality - what the system does

Requirements on functionality of a process support system can be divided into two categories: general requirements, and specific requirements. The general requirements are the same for all processes to be supported by the system. The specific requirements express the differences between the different kinds of business processes. The most important general requirements on functionality are as follows:

1. The system should store in one place all information that concerns ongoing and finished processes. This information should include the process's current (or the last for the finished processes) state, the process history (e.g., previous states) and the process chronicle (written history). Information on the ongoing processes is used to drive them forward. Information on finished processes can be used for statistical and other types of analysis, for learning by example, etc.

2. All planned activities assigned to a given member of staff should appear immediately in his/her personal calendar, so everybody knows exactly what activities he/she has to complete and when. This will remove redundant communication aimed only to inform people what they are supposed to do.

3. The system should help to execute activities. Some steps of the activity execution should be done entirely by the system, e.g., saving the history. Other steps could be done in cooperation with the end-user, such as choosing the next activities to plan. Providing some help in executing each activity is essential for motivating people to use the system. Without motivation the staff may leave some of the activities they completed unregistered, which would make the information on which the supporting system rely *unreliable.*

4. The system should be integrated with the tools needed for executing activities, such as word-processor, email, Internet browser, etc. This will allow competing many types of activities without having to leave the system.

5. The system should provide easy and flexible access to all information stored. For example, from an external organization name, it should be easy to find all the ongoing processes in which this organization participates. This will make it easier to sort the incoming mail (or other type of communication) that is not clearly marked with a process reference number. Access should be provided to current information, as well as to the historic one.

7.1.2 Properties - how well it suits the environment

The most important general properties are as follows:

1. The system should offer a user-friendly interface. This implies among other things the possibility of unhampered browsing through the information related to various processes, and the execution of activities while browsing. The latter applies to both planned activities, and the activities that are executed on the fly, i.e., without preliminary planning. The user-interface should also allow executing several planned activities at one go.

   The user interface should be *consistent* which implies that the standard techniques are used for browsing through all stored information, for planning, for starting the execution of activities, etc. These techniques should be uniform for all types of processes and activities. This is important for minimizing the initial learning efforts. It will also make it easier to introduce changes in process definitions, and support for new types of processes.

2. The control that the system exercises over the processes should be flexible with respect to individual processes, and to the organizational structure. The flexibility of the first kind allows to provide the high degree of automation when a process follows the standard pattern, without impending the possibility of manual control when the process deviates from the standard. The flexibility of the second kind makes it easier to introduce structural changes, e.g., when the company expands its operations.

   To achieve flexibility in respect to the organizational structure, it is important to separate what should be done from whom should do the task. The way of achieving flexibility in respect to individual processes depends on the view on the process dynamics chosen for system implementation

7.1.3 Quality - how well it handles fluctuations

We live in the world that undergoes constant changes. The changes concern both business reality and technology. As the changes happen during the system lifetime, the system should possess a degree of “reality tolerance”, i.e. adaptability to both kinds of changes. To reach proper adaptability, the control system should be built upon:

1. A sound conceptual model for representing business processes that allows one to redefine processes when the business reality changes without major changes in the model (i.e. without introducing a lot of new concepts).

2. Properly layered software structure, so that in case of advances in technology, only one layer needs to be changed at a time.

3. Conceptual model being logically as much as possible separated from the software structure, so that changes in the software structure and business model could be made independently of each other.

7.2 Pragmatic approach to system development

When a support system is designed, the general requirements on functionality, properties and quality should be translated into a development framework. The specific for each business process requirements on functionality should be translated into dialog screens, and subroutines that helps human beings to complete activities that are assigned to them.

A support system as a minimum should be structured in three conceptual layers:

1. A historical object-oriented database to store all information related to the processes, see line 2 in table 4.1. Software that implements this layer is of general nature, and it is independent of a particular process control system. This layer can be built upon some third party database management system.

2. A user-interface navigation system that allows the end-user to freely browse throw the information that concern processes' current state, past, and future (planned activities), see line 3 in table 4.1. Again, software that implements this layer is of general nature, and it is independent of a particular process control system. This layer can be built upon some third party user-interface management system.

3. Application-dependent routines for executing activities and controlling business processes. This layer can be implemented in a general purpose programming language (i.e. hard coded).

A moderate level of adoptability of the process control system to changes in business reality is achieved by the fact that only the third layer is domain and business dependent. It consists of a set of relatively small routines that are called by the general layers. Changes in business reality requires only reprogramming these routines.

A moderate level of adaptability to changes in technology is achieved by building general layers upon third party development tools. When the technology changes this layers are rebuilt based on other third party tools, the ones that provide solutions based on new technology.

7.3 Experience

7.4 Sales support

In our practice of building computer support for business processes, we use state flow view on business processes. Our first experience in building process support 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* since then.

The system was developed for the character-based environment and run under DOS on any LAN available on the market. The *historical database layer* was implemented upon the *Btrieve* record manager (now part of *Pervasive Sql* from Pervasive, Inc.). This layer was programmed in C, and it got the name Hi-base. The *user-interface navigation layer* was built upon JAM (JYACC Application manager) from JAYCC, Inc. (now Prolifics, Inc.) The layer was also programmed in C, and it got the name Navi. The *application-dependent routines* were written in C.

Hi-base and *Navi* 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).

7.5 Support for recruiting and investigations

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.

The second system, called *Utredaren*, is aimed to support inquiries, investigations, and inspections. This system has been built on commission for one of Swedish municipalities. Informal description of the business processes it is aimed to support is given in section 3.2. 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.

By the time we started the development of first *ReKo*, and then *Utredaren*, the technological base (*Hibase* and *Navi*) that was developed for the *DealDriver* project became old-fashioned. For these two systems, we have developed a new technological base.

The *historical database layer* has been implemented upon SQL database management systems: *ReKo* uses *Sql Server* from Microsoft, and *Utredaren* uses *Oracle*. Several programming tools were used for building this layer. Object and history structure is supported by a set of stored procedures and triggers implemented in the native for the given DBMS dialect of SQL. As far as object structure is concerned, we use the same way to define the database scheme as in the *DealDriver* project. This scheme is then translated into SQL tables and triggers definitions by a program written in Perl. Run-time mapping from objects into relational tables is done with the help of Transaction Manager (TM) from Panther, the tool we use for building user interface (see below). The tuning of TM to our historical object structure has been done by programming in C.

The *user-interface navigation layer* has been built upon Panther from Prolifics, Inc, which is the successor of JAM used for *DealDriver*. Currently, this layer works in the graphical environment under MS Windows; creating a Web browser version is planned for the future. The user-interface layer is implemented partly in C, partly in JPL, the proprietary language included in Panther.

The *application-dependent layer* has been written as stored procedures in the chosen SQL dialect, and as a set of JPL routines (the proprietary language of Panther).

Currently, both *ReKo* and *Utredaren* are being introduced in the business practice of organizations that ordered it. The details of these two projects will be published later. Here we would like to mention that our pragmatic approach to system development permitted us to change the technological base in a relatively short period of time and with limited resources (in terms of money and manpower).