Evidence-Based Interactive Management of Change

Christian Stary*

Department of Business Information Systems – Communications
Engineering & Competence Centre on Knowledge Management
University of Linz
Freistädterstraße 315, A-4040 Linz, Austria
E-mail: Christian.Stary@JKU.AT

Albert Fleischmann

Metasonic AG
Münchner Str. 29 - Hettenshausen 85276 Pfaffenhofen, Germany
E-mail: albert.fleischmann@metasonic.de

*Corresponding author

Abstract: Evidence-based interactive management of change means hands-on experience of modified work processes, given evidence of change. For this kind of pro-active organizational development support we use an organisational process memory and a communication-based representation technique for role-specific and task-oriented process execution. Both are effective means for organizations becoming agile through interactively modelling the business at the process level and re-constructing or re-arranging process representations according to various needs. The tool allows experiencing role-specific workflows, as the communication-based refinement of work models allows for executable process specifications. When presenting the interactive processes to individuals involved in the business processes, changes can be explored interactively in a context-sensitive way before re-implementing business processes and information systems. The tool is based on a service-oriented architecture and a flexible representation scheme comprising the exchange of message between actors, business objects and actors (roles). The interactive execution of workflows does not only enable the individual reorganization of work but also changes at the level of the entire organization due to the represented interactions.

Keywords: Archetyping; Organizational Learning; Business Process Management; Subject-Oriented Business Process Modelling; Process Prototyping

Biographical notes: Chris Stary is Full Professor of the Department of Business Information Systems – Communications Engineering, and Head of the Competence Centre on Knowledge Management at the University of Linz. He received a diploma, PhD, and venia docendi in computer science from Vienna University of Technology, Austria. His research interest includes business process management, usability engineering, knowledge elicitation, modeling, representation, self-managed education and learning technologies..

Albert Fleischmann is head of the Supervisory Board of Metasonic. He received his diploma and Ph.D. degree in computer science from University of Erlangen, Germany. His research interests are system theory, business process modelling and management, and communications engineering.

1. Introduction

Currently, the competitiveness of enterprises is mainly driven by their capability to implementing process-driven information systems. They should enable structural flexibility, besides addressing quality, cost, and partner/customer relationships (Stephenson et al., 2007). The need for velocity popped up when cross-enterprise operations had become relevant in the course of economic globalization. Structural flexibility requires adapting to global supply chains or evolving networks albeit increasing operational efficiency and effectiveness (Haeckel et al., 1999). In this context process-driven business and information systems development is of crucial importance at the tactical and operational level (cf. Laudon et al., 2005).

Velocity Depends on Accurate Process Specifications.

For (cross-)organizational operation process specifications have to be tailored to a specific business (opportunity) at hand. Of particular importance is their coherence, when different organizational roles or units are involved in a business case. Coherence requires the consistent propagation of business objectives to operational structures, e.g., reducing engineering cycle times through reporting loops, both on the level of process specification, and on the level of processing them, e.g., using workflow management systems.

Each process can be considered as functional entity with dedicated objectives and particularities that have to be integrated with those of the enterprise network or networked enterprise. Intelligible and flexible representational schemes for process specifications and information systems architecting allow enterprises to keep up with the dynamics of change. Process design and redesign has to be tightly coupled with control flows of enterprise information systems (cf. Rouse, 2006). However, a variety of techniques is used to capture process elements and the flow of control. In most cases, for process specification (targeting towards implementation) a language switch occurs shifting from natural to formal languages. Effects of that shift are of economical (costs), social (conflicts, negotiations), and organizational scale (iterations, quality control), mostly in case of mismatches between business requirements and information systems features.

Continuous Change.

Those effects are of particular importance when dealing with all types of changes due to the snapshot nature of specifications and models (cf. Lewis et al., 2007, p.15). It is caused by cognitive mismatches when using particular notations for specification, and leads to incoherent transformation of information throughout analysis, design, and implementation. Current tools do not resolve semantic-gap issues (as identified in the realm of semantic web applications - cf. Ehrig, 2007) for developments driven by business process modelling languages, e.g., Rumpfhuber et al. (2002). However, such problems become substantial when tasks are increasingly pushed to users (Chakraborty, 2004), as the control flow of integrated processes has to be semantically coherent on the level of work tasks, and syntactically correct on the level of technology support. In case users are not supported along the workflows they are involved in doing their business, interactive developments can easily lead to

- reduced productivity,
- decreased quality of work,
- lack of efficiency in the course of task accomplishment,
- problematic management of tasks (cf. Preece, 1993).

Bridging Semantic Gaps.

For our approach we will use experiences from model-based technology development (Stary, 2000) and subject-oriented business process engineering (Fleischmann et al., 2009). The ultimate goal of development should be a role- and task-conform workflow in a way that each stakeholder is able to experience a certain task in an effective and efficient way from a mutually tuned global organizational and individual perspective. The latter might require the adaptation of existing work processes towards individual work styles and personal preferences. Thereby, the interactive process artefact reflects the world of tasks as perceived or envisioned by stakeholders or responsible managers. It takes into account skills and preferences, as required for task accomplishment and for mutual interaction. As such representations have to be negotiated before coming into operation on the organizational level. Task and role models need to be communicated. Semantic interaction requires some ontology and a notation supporting mutual understanding.

The Process Perspective.

Both characteristics, the representation of task knowledge and its communication are crucial in organizational learning processes (cf. Nonaka et al., 1999, Davenport, 1998, Senge, 1990), in particular when knowledge is considered as a production factor. Common to the development problem in systems design seems to be, for knowledge being a production factor or product that knowledge needs to be located, acquired, analyzed, and developed. Analogous to design knowledge organizational knowledge has to be considered as a product as well as a process. As a process it addresses the individual and group-wise processing of knowledge items, i.e. learning at both levels. Managing this transfer process has turned out to be a challenging organizational activity (Nonaka et al., 1995, Probst et al., 1997). For analysis and development knowledge needs to be represented and stored in organizational memories (Ackermann, 1994). According to Kühn et al. (1997) it captures 'accumulated know-how and other knowledge assets and makes them available to enhance the efficiency and effectiveness of knowledge-intensive work processes'.

Evidence-based Interactive Management of Change.

In the course of (continuous) improvement of business processes and changes in organizations stakeholders play an important role. Herrmann (2000) proposes to qualify stakeholders by letting them develop parts of business processes individually. Using the workflow modelling language SeeMe (Herrmann, 1998) stakeholders create or edit diagrams describing their particular view on the assigned work tasks and their specific procedures for task accomplishment. The diagrams should be linked directly to software applications supporting the task execution. Through activating those links the stakeholders might immediately get an impression of executing the specified workflow and using the assigned applications, eventually involving different applications and different user interfaces. This example demonstrates the utility of tools enabling hands-on-experience.

In order to execute workflow specifications in the course of organizational learning such tools have to capture the organizational process on an individual level and to ensure process visibility for other stakeholders, on the level of the organization. We achieve that in terms of executable specifications. For individual stakeholders and management we provide workflow definitions and executable interactive artefacts presenting the data and communication facilities according to individual workflow definitions.

As soon as stakeholders adapt business workflows to their specific view, individual learning is stimulated. The process changes can be made visible by prototypical process execution. Together with the process description, they are means of communication between stakeholders - an organizational learning step is initiated.

In order to implement the above mentioned concepts we have used a language for human-centred process modelling, and a corresponding tool to execute individual workflow specifications immediately. Hence, the following work is located at the interface of knowledge management, organizational learning, and business process engineering (see figure 1). The interdisciplinary approach is given by the nature of our research objectives: For evidence-based interactive management of change, we need to integrate business process engineering in knowledge management processes. In this way, procedures dealing with work process changes can be improved. In order to achieve these improvements both on the individual and organizational level we use concepts from organizational learning. As evidence we consider all inputs triggering learning steps. They might stem from market or production developments, organizational brainstorming sessions, idea management or continuous improvement activities set by individual stakeholders. Any evidence should be documented in process specifications for reflection and further processing. The latter enables the interactive and collaborative management of change.

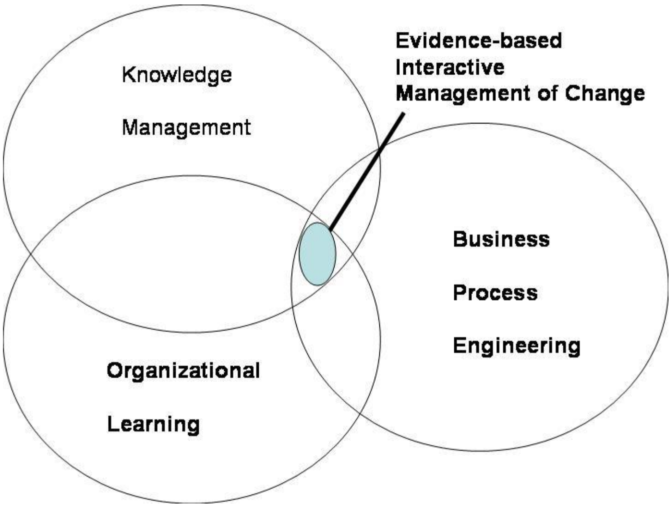


Figure 1. Interfacing knowledge management, organizational learning, and business process engineering

In the next section we describe the underlying conceptual framework stemming from organizational learning. In section 3 we elaborate the representation scheme for business process engineering and its interactive processing support for organizational knowledge creation. In section 4 we give the service-oriented architecture of the tool. In our conclusion (section 5) we summarize the objectives of our work, the presented achievements, and sketch our future research.

2. The Operational Frame of Reference

Organizational learning encompasses individual learning as well as learning on the organizational level and the transfer of knowledge between these levels. Brown and Duguid (1998) emphasize the importance of so-called *boundary objects* which help to transfer knowledge between individuals' or groups with different attitudes and presuppositions. For them, business processes are proper candidates for representing boundary objects because „business processes can enable productive cross-boundary relations as different groups within an organization negotiate and propagate a shared interpretation." For this reason, we have selected business process representations as enabler for individual and organizational learning processes and the transfer process (cf. process-based knowledge management in Abecker et al. (2002)).

In the following we describe how organizational learning processes occur and how they can be supported with business processes and business process modelling which is a technique for representing business processes encompassing structural and behavioural aspects.

In order to support stakeholders on the individual level, individual learning has to be stimulated. Kolb (1984) has proposed an experiential learning cycle which describes the learning process of individuals'. In figure 1 we incorporate the experiential learning cycle into our framework for organizational learning processes. Four steps can be used to explain "design" as activity for mutual discourse:

- *Observe*: individuals observe stimuli and their consequences from the environment
- *Assess*: the observations are assessed partly consciously, partly unconsciously
- *Design*: on basis of the assessment individuals form abstract concepts to react to the stimuli in an adequate way
- *Implement*: the developed concepts are implemented in real situations. The observation and reception of the effectiveness re-iterates the cycle.

Observation and assessment lay ground to evidence for management of change in terms of (re-)designs. They trigger a continuous experiental learning cycle:

- *Design*: Stakeholders express their role-specific view onto the business processes according to their task assignments and experiences.
- *Implement*: The workflow specifications of the refined business processes can be executed. This way, an interactive artefact is generated that enables hands-on-experience for role-specific task accomplishment. The visualization and prototypical execution of the workflow specification allows to put the modified business processes into action.
- *Observe:* Stakeholders observe through the interactive artefact generation and workflow execution possible effects the executed processes have on their work and the organization of tasks.
- *Assess*: If the results fit the expectations of the stakeholders, the concerned process serves as input for the learning process on the organizational level. If further process refinements or modifications are required the cycle starts again.

In order to transfer the knowledge acquired by the stakeholder to the organizational level we designed the following learning steps (see figure 1):

- *Negotiation* of the business process with all stakeholders whose work is affected by the process: the stakeholder who changed the process presents the process. The artefact generation and workflow execution help to visualize the changes in a straight-forward way and to initiate discussions. Participants in the negotiation process can vote for or against the changes or propose modifications themselves. The negotiation ends when all participants have accepted or rejected the proposed changes. Negotiation parameters should be set at the beginning of the negotiation like required percentage of votes for an acceptance /a rejection, or a time limit which breaks off the negotiation. The moderator of the negotiation process should be the responsible stakeholder for the business process (process owner).

- The negotiated business process has to be documented in form of a business process model. The model has to be stored into the *organizational memory* where all stakeholders have access to. The organizational memory is represented through a business process repository which stores the original process model and all versions which were negotiated. The latest negotiated process model serves as basis for further changes.

The organizational learning step is complete if the modified business process represents the novel basis for the work of the stakeholders. Figure 1 shows the operational frame of reference for learning.

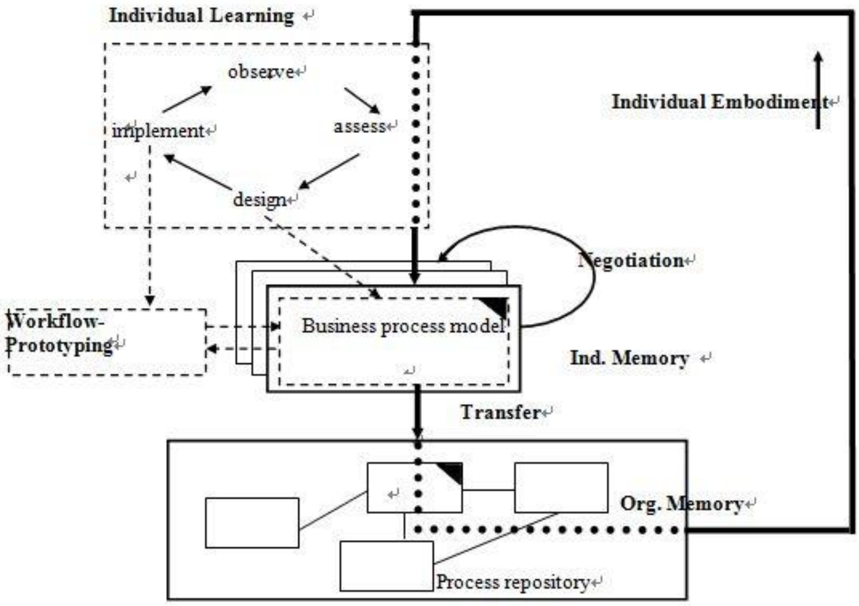


Figure 2. The Operational Frame of Reference of Interactive Change Management (according to Heftberger et al., 2004)

The directed link from the individualized business process model refers to the entire organizational memory, because all business processes of an organization are interrelated. The links between the business process model and workflow prototyping mean that experiences gained from using the interactive artefact generation and workflow execution influence the adaptation of business processes specifications, and vice versa, modified processes result in alternative actions (at the user interface) of the artefact.

The organizational learning cycle might be iterated as soon as stakeholders have used the newly acquired and specified knowledge on the level of the overall organization (denoted through the dotted line across the individual learning cycle). On the individual as well as on the organizational level, the ability can been acquired to put process specifications into practice. In this way, anticipated changes can be implemented prototypically, tested without disturbing the running business, and in case of acceptance incorporated. In case of rejecting the proposal no further implications occur.

3. Archtetyping

In the following we detail the representation scheme and the interactive tool support from a stakeholder perspective. We also give the procedure for proposing and experiencing workflow executions, i.e. initiating the organizational learning life cycle.

3.1. (Re-)Thinking Business in Terms of Socially Valid Processes

Understanding organizations as social systems we have defined a notation and specification language that

- is capable of describing an organization, therefore being as expressive as existing business process modelling languages, but provides an actor-specific and communication-oriented perspective
- makes use of semantic-net features to employ natural language expressions
- contains a minimal set of elements and relations in order to describe an organization as accurate as possible in an intelligible form
- provides descriptive items and relations to identify relevant codified (documented) knowledge for task accomplishment.

The language captures the

- tasks to be supported,
- users' perception of tasks in terms of involved actors,
- problem domain data required for task accomplishment, and
- exchange of messages to complete work tasks.

Lessons learnt from model-based design.

Designing information systems in a user-centered and task-driven way, several models could be distinghuised and explored in the development of the *ProcessLens* approach (cf. Dittmar et al., 2004; Stary, 2000):

- The *user model* sets up a role model by defining specific views on tasks and data (according to the functional roles of users).
- Business processes as such are modeled partially through the *task model*. Comprehensive processes can be composed of subprocesses. This step can be executed recursively until elementary task actions become evident. Additionally, temporal constraints can be set between tasks and processes. The task model can be designed according to the organization of work and the stakeholders' perception of work, usually being a part of the business intelligence representation.
- The problem domain or *data model* describes the data derived from the tasks and user organization in the problem domain. In contrast to traditional data modeling, in *ProcessLens* both aspects of the data required for task accomplishment are captured:

the static and the dynamic properties. Typically, objects of work are specified in the data model.

- Last but not least, the *interaction model* provides all interactive elements and interaction styles that are needed to make the business processes visible for stakeholders on the one hand and executable on the other hand.

ProcessLens provides dedicated relationships to ensure context-sensitive representations, and to prohibit models that can not be executed prototypically due to incoherent refinement and inconsistencies. Each of those relationships, such as ‘is handled’ between roles of a user model and subtasks of a task model, are checked through particular algorithms according to their semantics and syntax. UML (www.uml.org) is used for specifying the various models and their relationships. Each object relevant to actor behavior or task accomplishment is described using *attributes* and *stereotypes*, which indicate which category of model element is addressed, e.g. task, activity, role). Methods are described using a*ctivity diagrams*.

Activity diagrams model the behavior of actors, task accomplishment, data management, and user interaction with the information system. They contain activity states, (optional) transitions, fork and join vertices. Using these elements, either XOR, OR or AND relations can be modeled. Connecting states with transitions allows for constraints when operating a business. These conditions are triggers to reach the next state. Activity diagrams are prerequisites for the adaptation of business processes to the understanding of the stakeholders.

Behavior descriptions might be useless when the context of application is not intelligible. Therefore, a dedicated connection between activity diagrams had to be introduced. For instance, the activity diagram of ‘Order Entry’ has to be connected to the activity diagram of ‘Order Form (electronic)’ through synchronisation links, since the form (part of the interaction model) enables order entries (part of the task model). These links allow specifying the detailed flow of control. Hence, as soon as a certain state, such as ‘input order data’ is reached, the expected behavior continues with a state from the activity diagram of ‘Order Form (electronic)’ and returns when filling in the form has been completed. A synchronisation link is a special kind of transition, which had to be added to standard UML, in order to ensure the automated execution of task flows.

Besides the usefulness of keeping actor- and task-specific perspective on process specifications, the *ProcessLens* project revealed the benefits of automated execution in terms of immediate experience of task flows, not only for stakeholders, but also for customers, managers, and organization developers.

Utilizing an integrated, generic scheme for specification.

The integration of the various perspectives is enabled through communication links that synchronizes behaviour sequences (similar to *ProcessLens*). A respective template is used as representational frame of reference. The following figure shows a template of type 3, since it contains 3 involved parties. All parties exchange messages. In order to distinguish concrete persons from roles in a process we call the parties in a process subjects. Subjects are the acting elements in processes, like in sentences of natural languages where the subject is also the acting element. The subject which starts the message exchange is marked with a small white triangle (subject1).

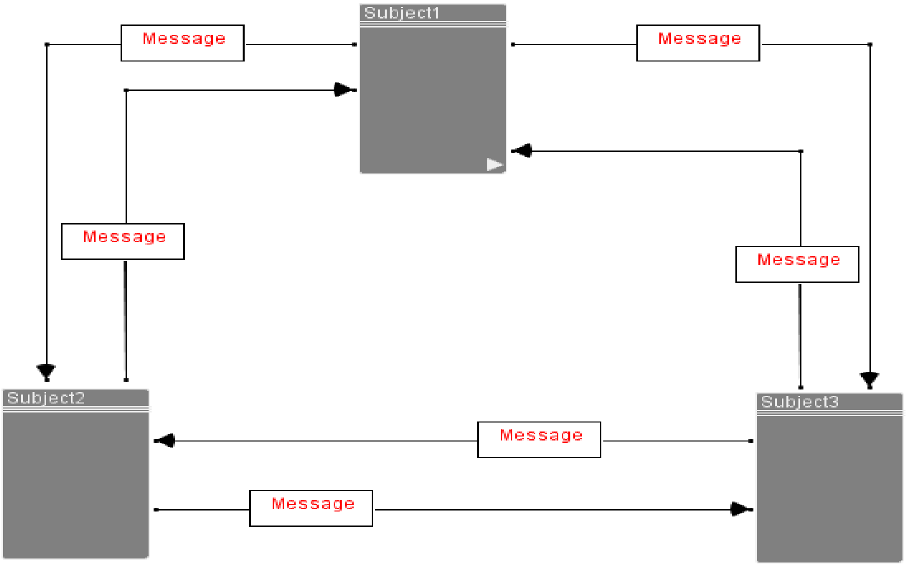


Figure 3. A process template with 3 involved parties

Each subject can send messages with the name Message to any other subject any time. Figure 4 shows the behaviour of the subject with the name ‘subject1’. Since subject1 is the subject which starts a process its ‘start’ state is the state ‘select’. The ‘start’ state is framed.

The state ‘start’ and the transitions to the state ‘select’ are never executed in the start subject. This state is the ‘start’ state in all the other subjects. All the other subjects are waiting for a message from (all) other subjects. In this way each subject not being a start subject has to receive at least one message before they can start to send messages. The start subject sends a message to any other subject. The receiving subject can reach now the state ‘select’. In that state a subject can decide upon its next action without restriction. A subject which is in state ‘select’ can send a message to other subjects which are still in the state ‘start’. Now these subjects can also reach the ‘select’ state and can send messages. Finally, all subjects are in the state ‘select’ and can communicate whenever being addressed.

In the ‘select’ state the start subject decides whether it wants to send or to receive a message. In the beginning it does not make sense to receive a message since the other subjects are waiting for messages. All the other subjects are in the state ‘start’ which is a ‘receive’ state. This means the start subject will start with sending messages. After that mutual message exchange can start. When becoming active in the ‘select’ state a subject decides to use the ‘send’ transition. In the state ‘prepare message and select address’ the subject instantiates the business object that is transmitted by the message ‘message’. After that a subject decides to which subject the message with the business object as content will be sent.

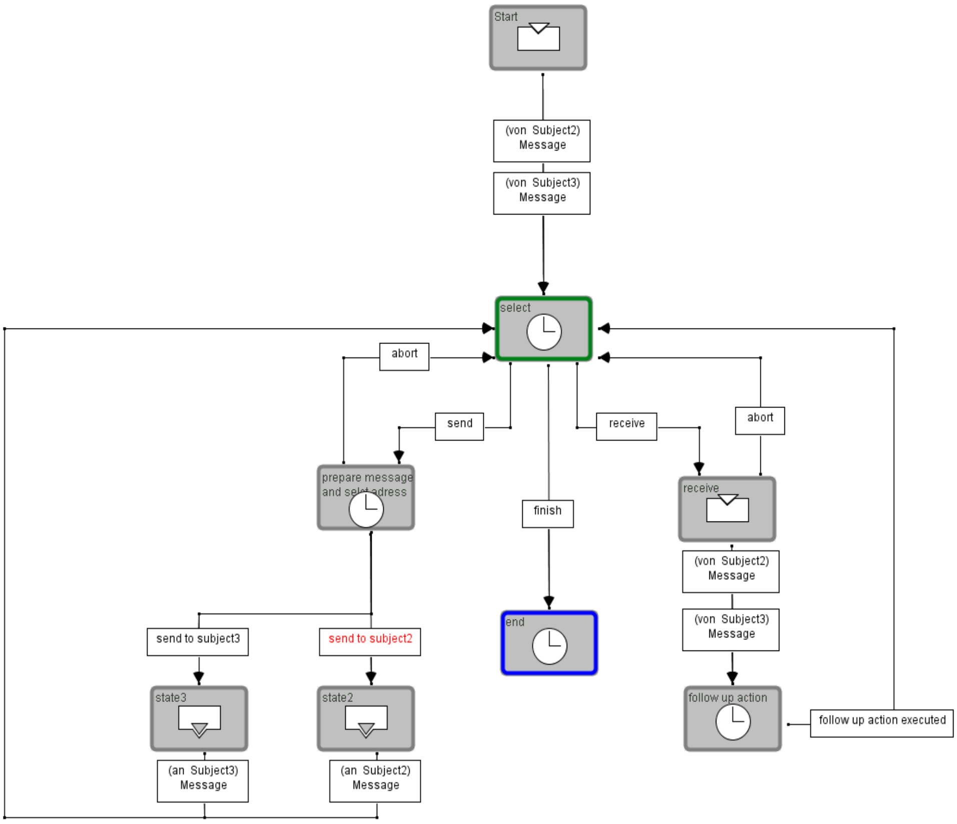


Figure 4. Behaviour of the start subject 'subject1'

In the 'select' state a subject can also decide whether it wants to receive a message. If a message from the expected subject is available the message can be accepted and a follow-up action can be executed. It is not specified what the follow up action is. It works like receiving an e-mail. The receiver is able to interpret the content of an e-mail and knows what the corresponding follow-up action is. The 'abort' transitions back to the 'select' state enable to step back in case a subject has made the wrong choice.

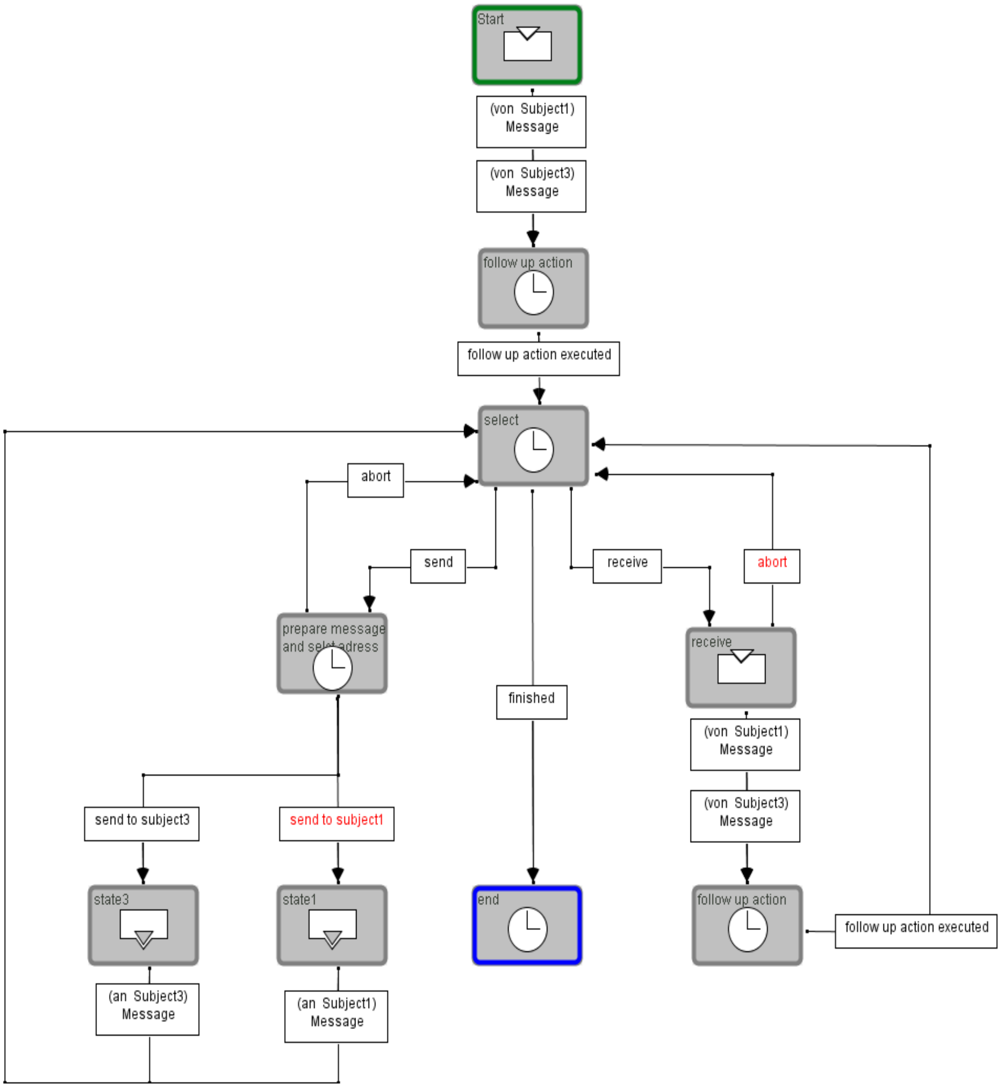


Figure 5. Behaviour of Subject2

Above we have shown a generic process scheme with three participants. Such types of schemes are universal, as they can be easily created for any number of participants. The following figure shows the generic structure of a process with four participants (type 4 template). The behaviour of each subject has to be adapted to the corresponding number of subjects in a generic process. Figure 7 shows the behaviour of the start subject ‘subject1’. The modeler or the tool needs to add the respective ‘send’ and ‘receive’ transitions between corresponding states. In the ‘send’ area transitions must be added to send a message to the new subject, as for the ‘receive’ area. In the ‘receive’ state a corresponding transition has to be added. Using that extension principle the behaviour for each type of generic process scheme can be generated automatically.

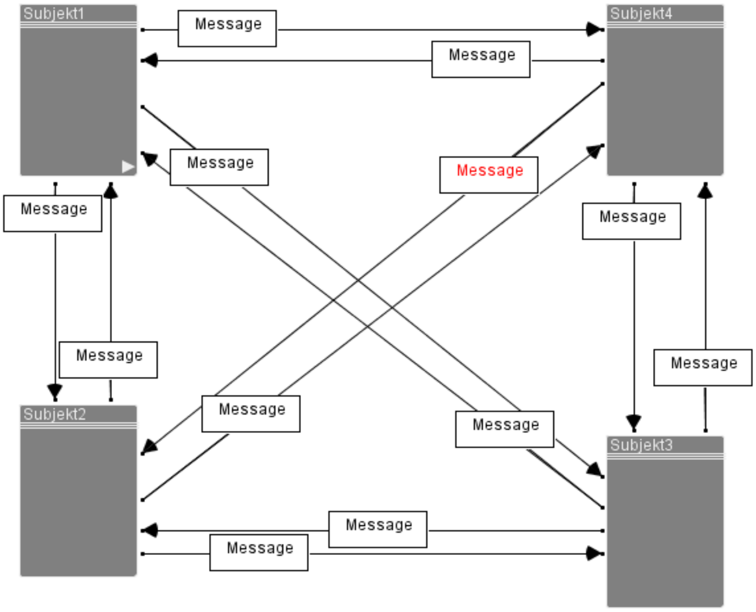


Figure 6. Process template for 4 participants

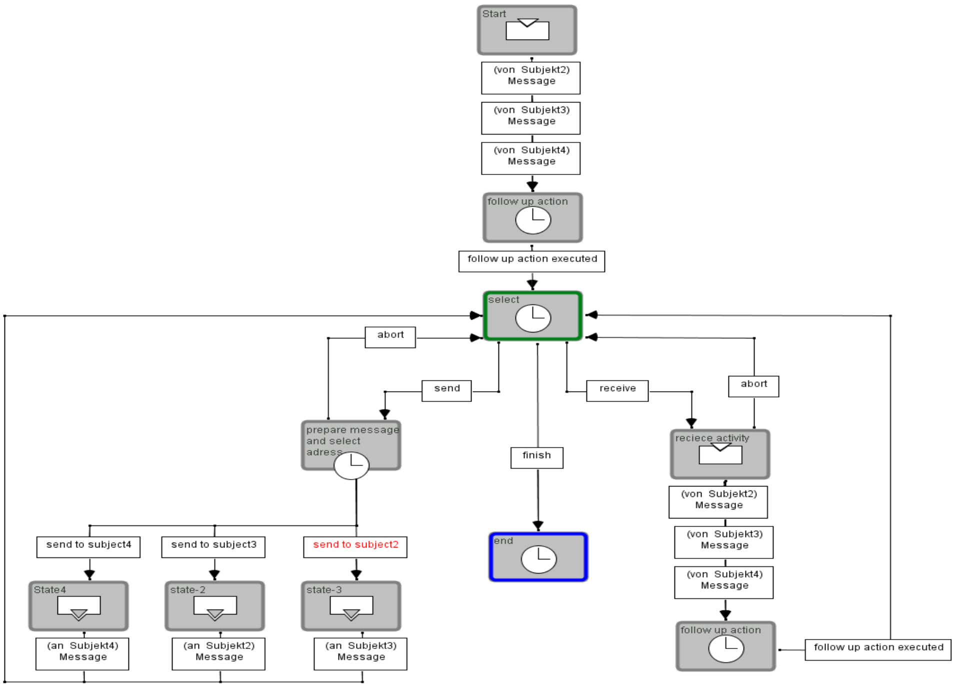


Figure 7. Generic behaviour pattern of start subject ‘subject1’ for a four party process

With the message ‘Message’ a corresponding business object is sent. The structure of this business object corresponds to the structure of a mail with some extensions like keyword and signature. The following figure shows the specification of the business object message in a XSD notation.

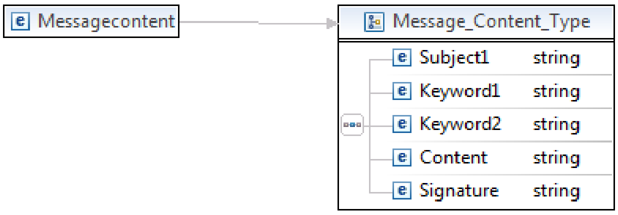


Figure 8. Structure of the Mail Business Object

Whenever a message ‘message’ is sent, such a business object is sent. The values for the components of the business message object correspond to the content of a traditional mail.

Adopting the Generic Scheme.

Subjects are abstract resources. They represent the parties involved in a process. For the specification of an actual workflow the various subjects of a process must be assigned to existing roles, persons or agents. The assignment of persons or agents to subjects means that a process is embedded into a concrete environment. The following example shows an application of a generic process scheme of type 3 in an actual work organization.

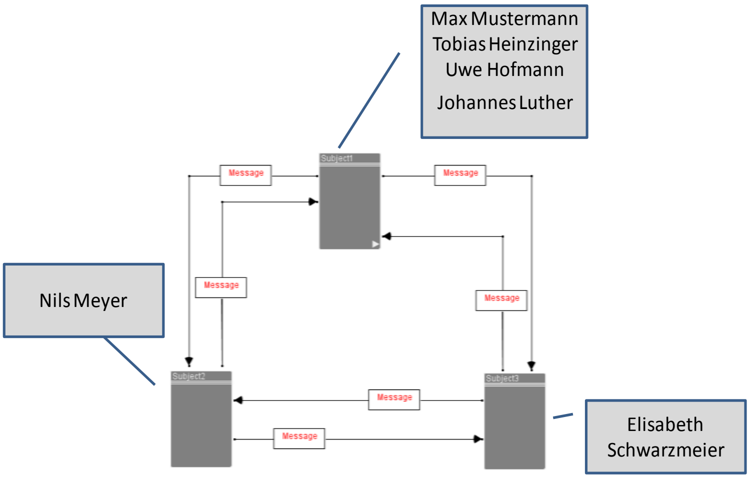


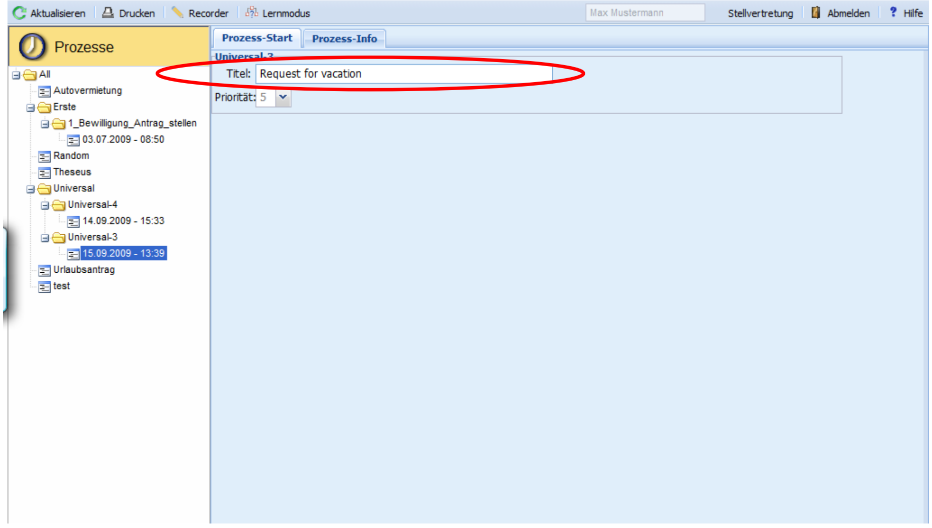
Figure 9. Embedding a process in its environment

The persons Max Mustermann, Tobias Heinzinger, Uwe Hofmann und Johannes Luther are assigned to subject ‘subject1’. Since these persons are assigned to the start subject all of them can start the process. For instance, Max Mustermann creates a message and sends it to Nils Meyer. Nils Meyer can accept that message and can send a message back to ‘subject1’ - the message is received by Max Mustermann. Max Mustermann receives the message because in his work environment or context the process is started. If another person assigned to 'subject1' starts a process this process instance is executed in his or her environment.

The embodiment of a process depends on the usage of a generic process specification. It depends on the business events to be handled. In our example vacation requests are handled. Nils Meyer as head of a development department is responsible for Max Mustermann, Tobias Heinzinger, Uwe Hofmann and Johannes Luther. Subject3 is assigned to Elisabeth Schwarzmeier. Subject3 represents the human resource department of the organization.

Evidence-based Interactive Experience.

The execution of work processes can be supported by an appropriate workflow system. In our research we use a suite developed for subject-oriented process specifications and workflows (Fleischmann et al., 2009). The following figure shows a screenshot as Max Mustermann creates a new instance of a generic process template for 3 parties (i.e. of type 3). This new process instance has the title 'Request for vacation' (see ellipsis in the figure).



Universal-3

Titel: Request for vacation

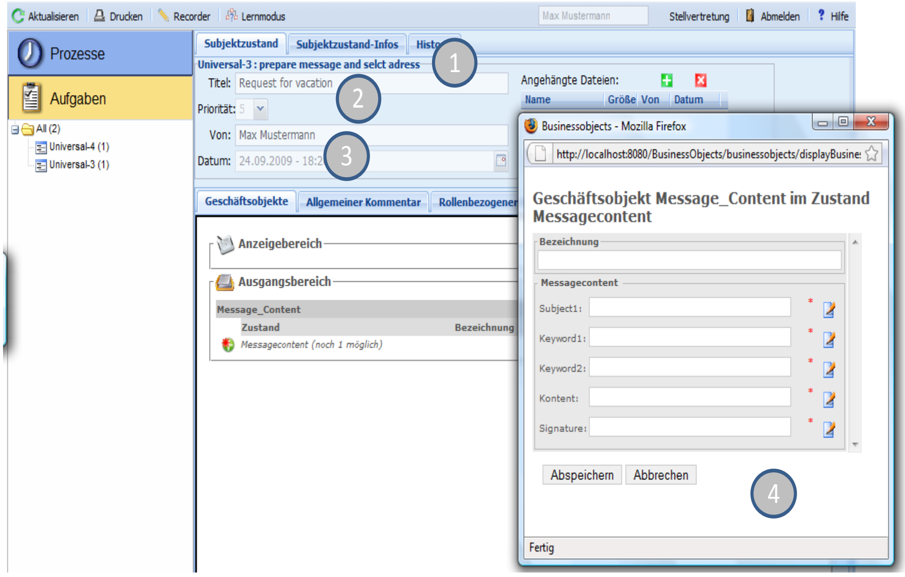
Priorität: 5

Figure 10. Max Mustermann creates a process instance with the title 'Request for Vacation'

After creating the process instance Max Mustermann is guided through the process. He is asked by the workflow system which transition he wants to follow. He knows that he has to fill in the business message form with the corresponding data and that form has to be sent to Nils Meyer. Consequently, Max Mustermann follows the transition 'send'. In the state 'prepare message and select address' following the transition 'send' he fills in business object data required for the application for vacation.

In the following figure the user interface of the workflow system and direct experiencing of the specification are shown.

- Number 1 in that figure shows the name of the current state: ‘Prepare message and select adress’.
- Number 2 shows the title of that process instance: ‘Request for vacation’.
- Number 3 shows the creation date of that process instance.
- Number 4 shows the form for instantiating the business object.



Universal-3 : prepare message and selct adress

Titel: Request for vacation

Priorität: 5

Von: Max Mustermann

Datum: 24.09.2009 - 18:2

Max Mustermann

Message_Content

Zustand	Bezeichnung
Messagecontent (noch 1 möglich)	

http://localhost:8080/BusinessObjects/businessobjects/displayBusine

Geschäftsobjekt Message_Content im Zustand Messagecontent

Subject1: *

Keyword1: *

Keyword2: *

Kontent: *

Signature: *

Figure 12. User Interface of the workflow system in state prepare message and select the person(s) to be addressed

Max Mustermann can put in all the required data for a vacation request to the business object and send it to Nils Meyer who is the owner of subject2. This is all Max Mustermann needs to know, since the behaviour description of his subject would allow sending the vacation request to the Human Resource department, conform to using a mail system. Each stakeholder needs to know to whom he/she has to send a mail in the course of task accomplishment. The workflow used in that example produces a protocol recording who has executed a certain action at a certain time. The following figure shows an example of an execution path for handling a vacation request with a generic process scheme. The steps executed in each subject are shown in a corresponding column with the subject name as head line.

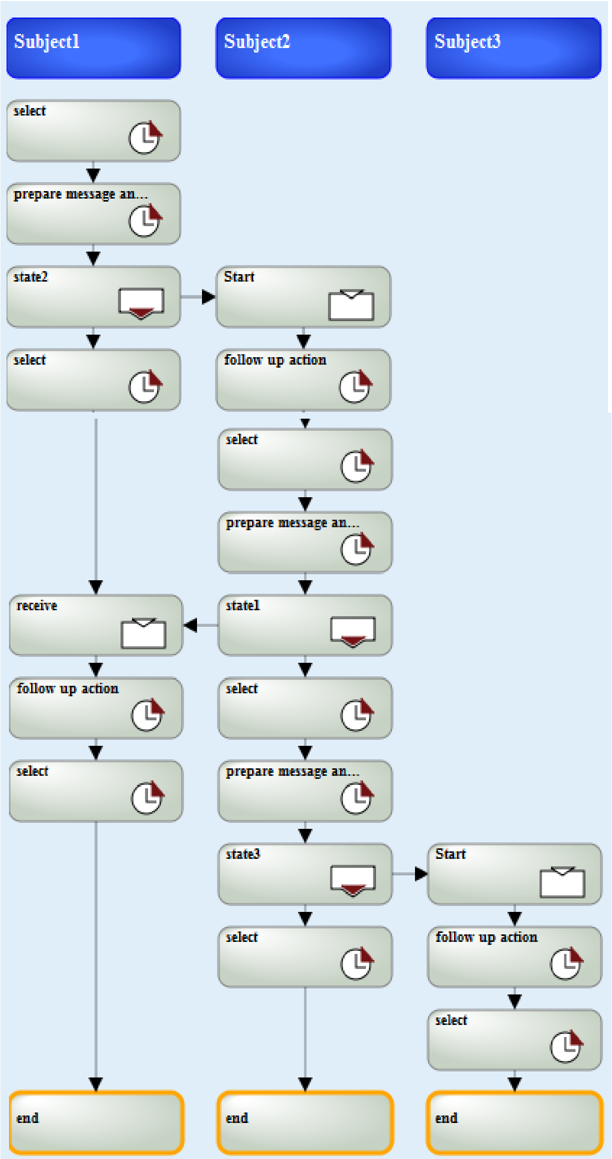


Figure 13. Execution path of a generic 3-party process scheme for a vacation application

Subject1 starts with the ‘select’ activity and selects the ‘send’ transition. Then the action ‘prepare message and select address’ is executed and in state ‘state2’ the message is sent to subject2. Now subject1 reaches again the state ‘select’. In state ‘start’ subject2 receives the message. In the subsequent state ‘follow up action’ the content of the received message is read and the corresponding action is executed by Nils Meyer who is the owner of Subject2. In the case of vacation application this follow-up action is Nils

Meyer´s decision whether the vacation application is accepted or denied. This decision must be sent to subject1. In state 'select' subject2 decides to follow the 'send' transition, prepares the message with the result of the decision, and sends it to subject1. This swim lane diagram shows which subject executes which actions in which sequence. If a subject sends a message the 'send' state is connected with the corresponding 'receive' state in the receiving subject. Subject1 sends a message to subject2 in state 'state2'. Subject2 receives that message in state 'start'.

A subject-oriented workflow system can guide each party involved in a process. It can be used to produce an execution protocol recording the sequence in which messages have been exchanged between the involved parties. Another advantage is that a workflow for a generic process scheme can be generated automatically. The only parameter that must be known is the number of involved parties besides the subject starting the process execution. Once completely detailed, a proposed business process can be experienced and immediately negotiated using the subject-oriented workflow system. We exemplify the required steps in the next sub section.

3.2. Submitting a Process Specification to an Organizational Memory

Detailing a generic process scheme can be described as omitting all items that are not required for task accomplishment, or 'refinement through restriction'. There are several restriction steps:

1. Remove message connections between subjects that are not required.
2. Name the subjects according to the application domain.
3. Name the messages and introduce message types according to the application domain.
4. Adapt specification to actual subject behaviour.
5. Refine the structure of the business objects transmitted by the various messages.

After each restriction step the process can be executed by a subject-oriented workflow system. With each restriction step the guidance for the subject holders is becoming more stringent to task accomplishment.

In the following the application of these restriction steps is exemplified for handling vacation requests, starting with a generic process scheme of type 3.

As the process specificiation can be embedded into the organization after each restriction step, the corresponding workflow can be used like shown above, according to the available resources.

Remove unnecessary communication paths.

We assume that Subject1 represents the stakeholder asking for vacation, Subject2 represents the manager and Subject3 the human resource department. When handling the process 'application for vacation' there is no communication between the employees and the HR department. Therefore the modeler needs to remove the corresponding exchange of messages. In addition, HR never sends a message to the manager. The modeler needs to remove the exchange of messages between Subject3 and Subject2. The following figure shows the resulting communication structure.

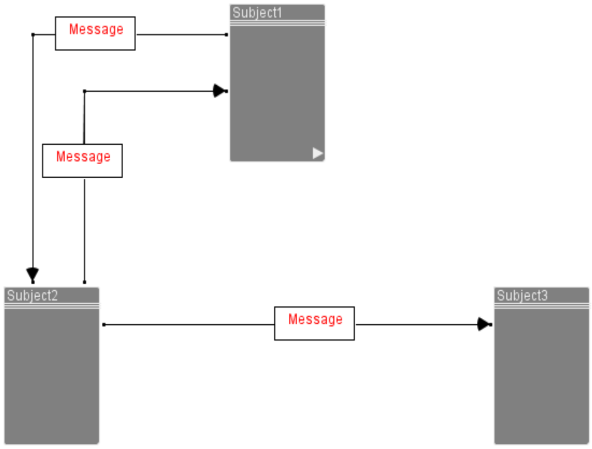


Figure 14. Restricting communication

According to the changes in the communication structure the behaviour of the subjects has to be adapted. The corresponding ‘send’ and ‘receive’ branches have to be removed. The following figure shows the adapted behaviour of Subject1. The ‘send’ branch to Subject3 and the ‘receive’ branch to Subject3 are removed.

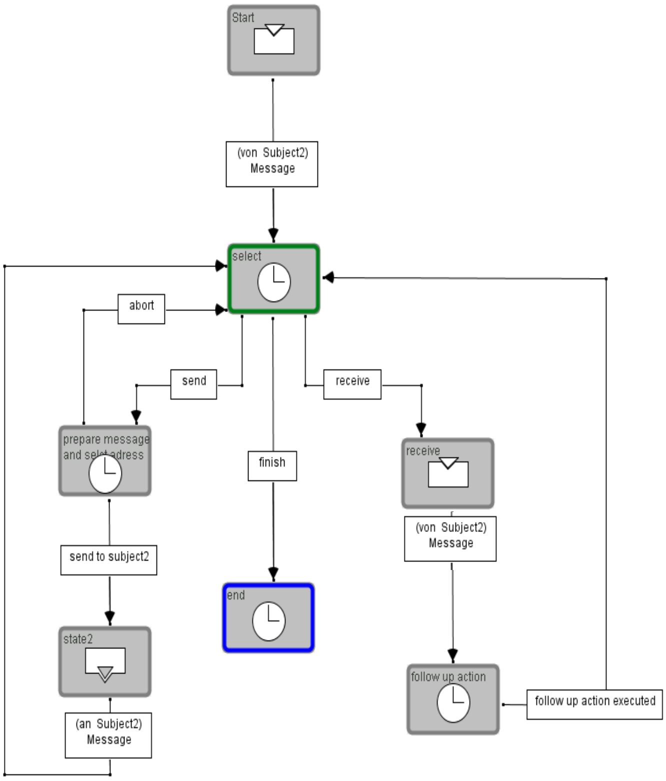


Figure 15. Communication to Subject3 is removed in Subject1

Analogously, the behaviour of Subject2 is adapted. The 'receive' transition for messages is removed from Subject3. The following figure shows the resulting behaviour.

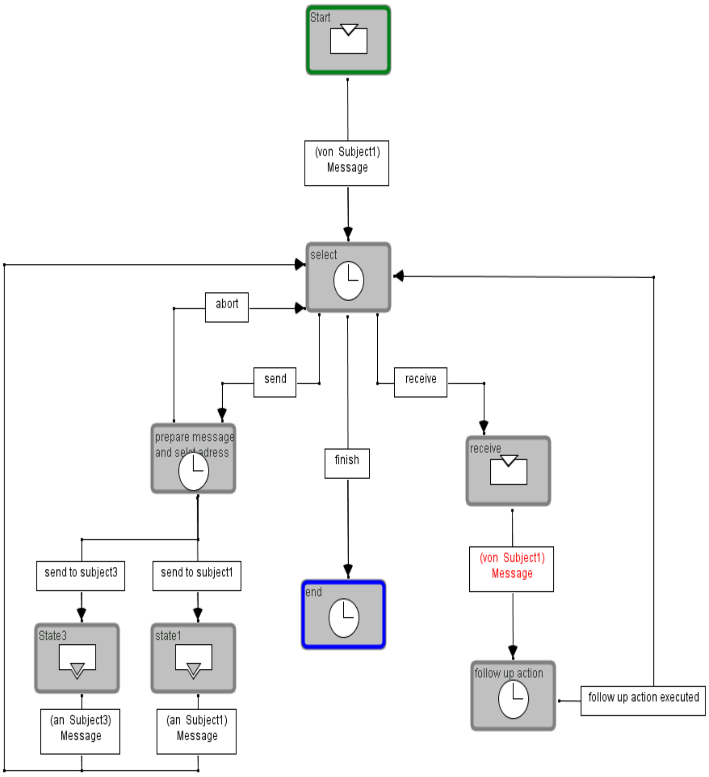


Figure 16. Subject3 does not send any messages

The changes in the workflow system restrict the possible interactions between subjects according to the vacation handling process, but there are still message exchanges allowed which do not meet the requirements of the anticipated vacation process. Subject1 can send another message with an application for vacation, Subject2 can send several messages with an answer to Subject1 etc.

Getting Concrete.

In a next step the subjects have to be named. Generic names need to be replaced to fit the application. Since the intention is to create a process and workflow for handling applications for vacation, the subjects are named according to that domain. Subject1 is renamed to 'employee', Subject2 is renamed to 'manager' and Subject3 is named 'HR'. The following figure shows the communication structure of the resulting specification with the domain-specific names of the subjects.

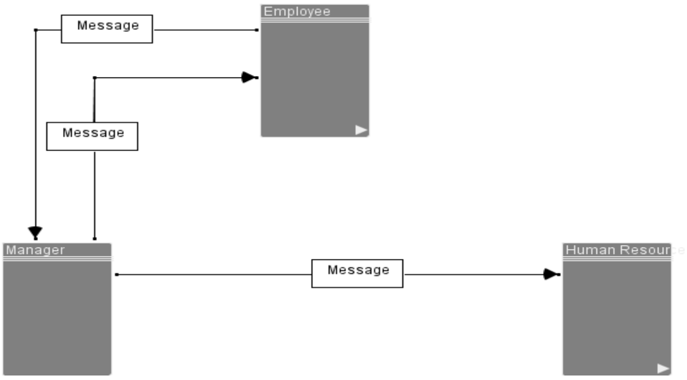


Figure 17. Communication structure with domain-specific subject names

The naming of the subjects has also some impacts on the communication behaviour. In the subject behaviour specification the corresponding sending or receiving subjects must be also adapted. The following figure shows the adapted behaviour of the subject 'employee'.

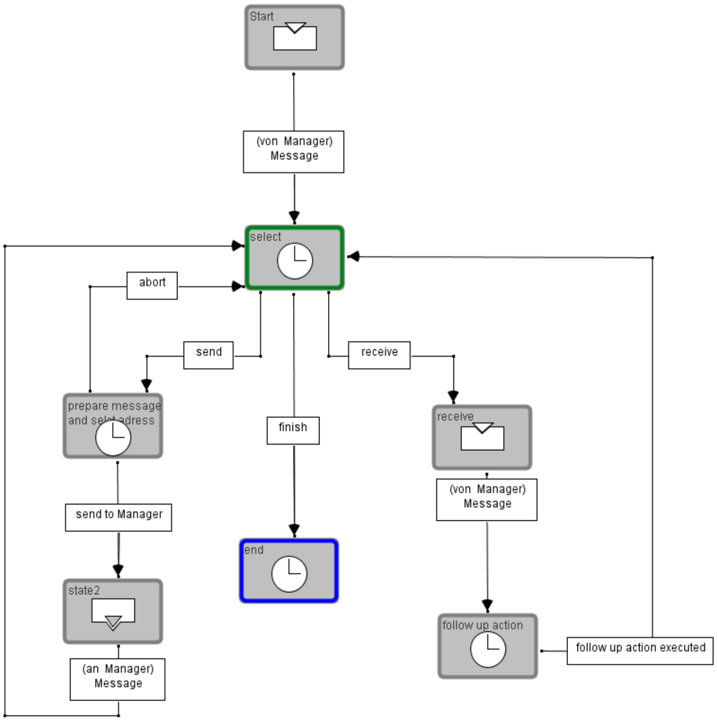


Figure 18. Behaviour of subject 'employee' with actual names of communication partners

The behaviour of the other subjects is adapted in an analogous way.

Instead of using the generic message name ‘message’ the modeler need to rename the messages in accordance to the application domain. He/She might introduce additional message types. The name of the messages exchanged between subjects can give some indications about their content and meaning. Instead of sending a message of the type ‘message’ informing the subject ‘manager’ only after reading that this is an application for vacation, the subject ‘employee’ can send a message with the name ‘application for vacation’. In this way the name of the message type indicates the intention of sending a message in the context of a certain process specification.

The following figure shows message types relevant for handling vaction requests. Subject ‘manager’ can send two different message types to subject ‘employee’, one message type for acceptance and the other for denial. The new message names improve also the readability of the process.

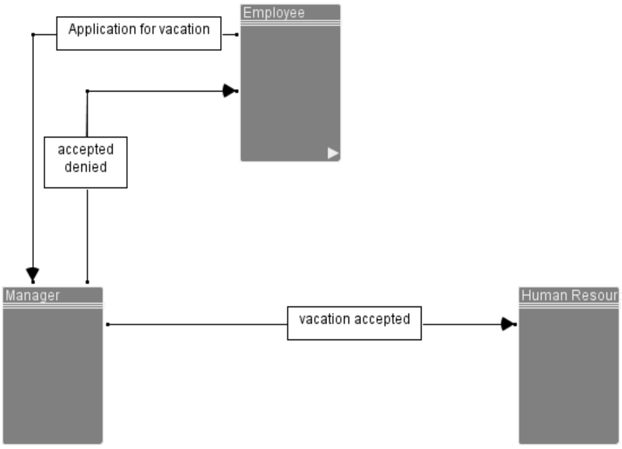


Figure 19. Communication structure with actual message types

The new message types have also some impacts on the behaviour specifications of all subjects. The message type ‘message’ must be replaced with the new message names and for the new message types additional ‘send’ and ‘receive’ transitions must be added. The following figures show the adapted behaviour of the subjects ‘employee’ and ‘manager’. The simple behaviour of subject HR can be adapted in a similar way.

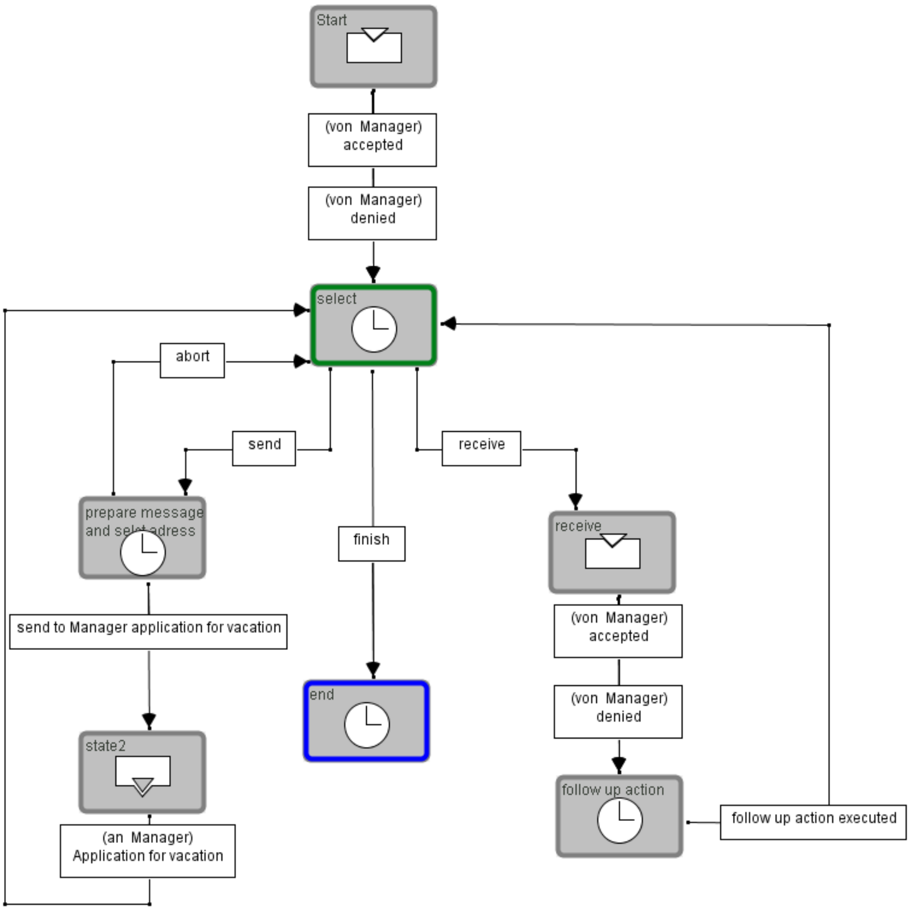


Figure 20. Behaviour of subject 'employee'

While renaming message types and adding new message types the process becomes more intelligible. Proper names for message types help users to grasp the meaning of a message type. But there are still the same problems we have after re-naming the subjects. Although the process can be better understood users might still send messages to subjects that are out of scope for the addressed task accomplishment, such as a second message 'application for vacation' to the manager. Still, it requires cognitive effort to organize effective and efficient communication having a certain process in mind.

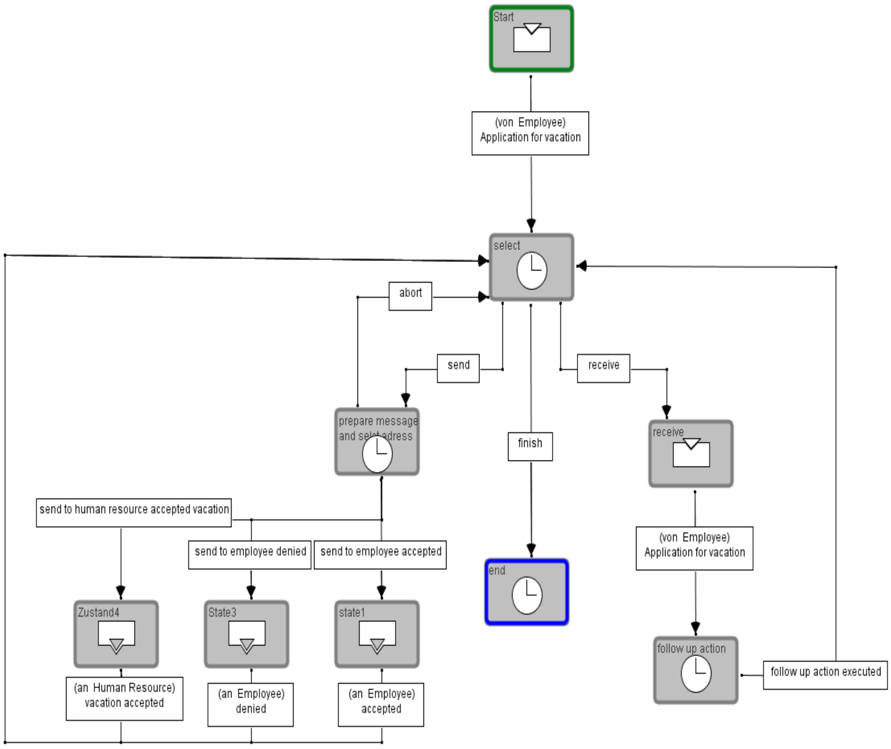


Figure 21. Behaviour of subject 'manager'

Up to now the already restricted behaviour of the involved subjects allows to send messages which are not in line with the intended handling of a business event. In order to achieve a coherent picture the behaviour of the involved subjects has to be restricted further. The following figure shows the modified behaviour specification of the subject 'employee'. According to that spefication the subject 'employee' can only send a single message 'application for vacation' to the subject 'manager'. After sending that message the employee has to wait for the answer from the manager. In case the subject 'employee' receives the message 'denied' from the manager the 'end' state is reached and the process execution is finished. In case the subject 'employee' receives the message 'approved' the state vacation is reached which means the action 'vacation' is executed. If this action is finished the 'end' state is reached.

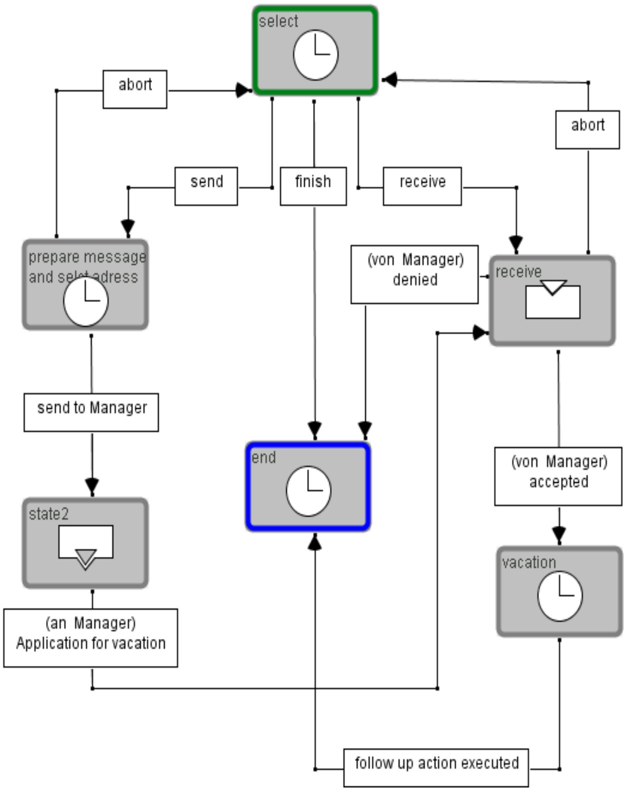


Figure 22. Complete behaviour refinement of subject 'employee'

The following figure shows the restricted behaviour of the subject 'manager'. After receiving the message 'application for vacation' the subject 'manager' decides whether he/she accepts or rejects that application. In the acceptance case the subject 'manager' sends the message 'accepted' to the subject 'employee', and after that the message 'accepted vacation application' is sent to the subject 'HR'. Then the subject 'manager' reaches the 'end' state. In the case of rejecting the application the subject 'manager' sends the message 'denied' to the subject 'employee' and reaches another 'end' state.

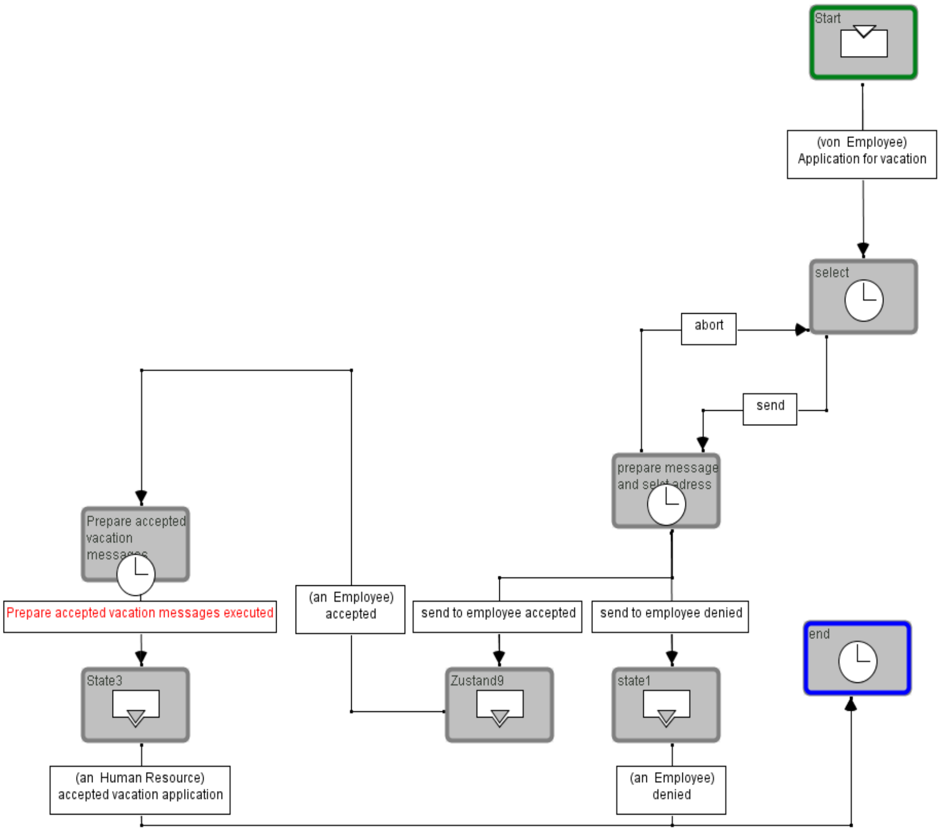

Start
(von Employee)
Application for vacation
select
abort
send
prepare message and select adress
Prepare accepted vacation messages
Prepare accepted vacation messages executed
(an Employee)
accepted
send to employee accepted
send to employee denied
end
State3
Zustand9
state1
(an Human Resource)
accepted vacation application
(an Employee)
denied

Figure 23. Final behaviour refinement of subject 'manager'

After those changes in the behaviour individuals can be guided through the process in a task-conform way without generating behaviour that is out of scope or context of the addressed task. There is only one issue remaining open. In the message specifications the subjects still use a generic business object for transferring the details about the vacation. This problem can be solved by defining a business object, and thus, representing the application domain of the process.

Finally, each addressed business object needs to be detailed, as shown for the vacation application form.

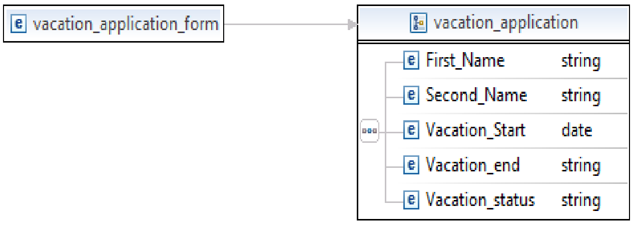


Figure 24. Sample Structure Specification

Then negotiating on the level of the organization is instantiated. It might lead to further adaptations. The versioning support of the organizational memory allows tracing the process of gathering inputs and compromising.

4. Service-Oriented Architecting of the Tool Support

Subject-oriented models have to be considered from the perspective of (distributed) software architecting for implementation. This conversion must be executed fast and at reasonable costs. Although functions can be derived through stepwise transforming model elements, service-oriented architecting is more effective and efficient. Users (i.e. subjects) could compose information systems based on components implementing subject activities. Such an assembling would not require programming at all. The concept of service-oriented architectures (SOA) has been designed to support such type of implementations.

In the following we demonstrate the straightforward use of subject-oriented models for SOA-implementations. The fundamental concept of SOA is the decomposition of software applications in coherent parts. Each part can be linked according to organizations, user's or application's needs. Such a composition allows generating applications along business processes in an effective and efficient way. Sequences of service calls replace change requests for existing applications involving IT and organization departments.

Using SOA requires business applications to be decomposed into functions or services, such as checking a vacation request, or updating a days-off sheet. These functions might stem from various platforms. Once functions become available as independent parts, developers might group them to establish novel functionalities. Neither the entire program nor the business logic is located in a single program, but rather distributed over several systems, including organizations.

A complete SOA requires control mechanisms assigning services to activities or tasks and implementing the overall flow of control. Each service has to be triggered accurately, in order to meet the requirements of the application. As these control mechanisms are the starting point of activities, they represent the subjects of SOA. For the definition of process control, services orchestration and service choreography can be used.

Service Orchestration and Service Choreography.

Orchestration describes the sequence of services as required for business operation. Orchestrating operations or service involves some kind of flow diagram mechanisms to specify the sequences of utilized service operations. BPEL has been defined as formal description language (Business Process Execution Language) – cf. Havey (2005). The execution of an orchestration requires central control to follow the defined flow of control. So far, the integration of user interactions is very costly, as BPEL targets for automated processes (workflows). Each process has at least one or two users, one triggering the process and one interested in the result.

Each activity is orchestrated, however, the activities synchronize themselves exchanging messages. The choreography of services is the description of direct interactions of the parties involved in a process, e.g., a sequence of messages. Each party involved in a process has its own flow of control, synchronized by exchanging messages. Hence, each user is responsible for executing process steps. In contrast to orchestration, when using choreography the system elements follow an agreed plan, however, pursuing individual paths to accomplish the involved tasks (initially promoted and then monitored by a controlling party). The following table summarizes the respective concepts.

Table 1. Ochestration and Choreography

Orchestration	Choreography
• Central control for process execution • Central process execution cannot be implemented for multi-threaded organizations • Tendency of serialization of process activities	• Each party is responsible for correct execution sequence • Multi-threaded organizations can be supported through decentralized process execution • Tendency of parallel process activities

Service-Oriented Architecture (SOA).

Service operations, their orchestration or choreography, and the recognition of users follow a 3-tier concept, as shown in the figure.

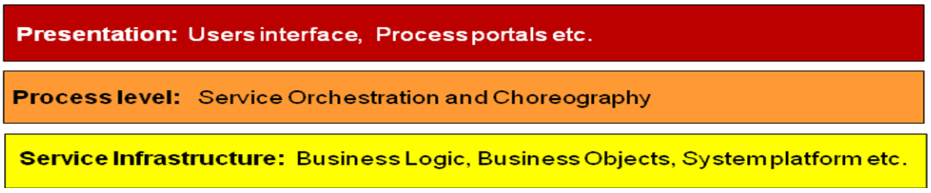

Presentation: Users interface, Process portals etc.
Process level: Service Orchestration and Choreography
Service Infrastructure: Business Logic, Business Objects, System platform etc.

Figure 25. Layers of SOA

The top level concerns the interaction with users, activating operations, providing inputs to functions of a process, and processing their results. Below the implementation of the business processes is addressed. In this level the sequence of functions (implemented in the bottom layer), or user inputs are expected (top level). As such, level

2 contains most of the business process logic. The bottom level provides all service operations available on level 2.

The subjects or actors correspond to the presentation level in SOA. Activities correspond to the operations assembled according to the orchestration. In Figure 26 the process description is given using EPC notation (IDS, 2000), orchestrating all services including user interactions. The service infrastructure contains objects. Their operations are used as activities (termed predicates in the figure).

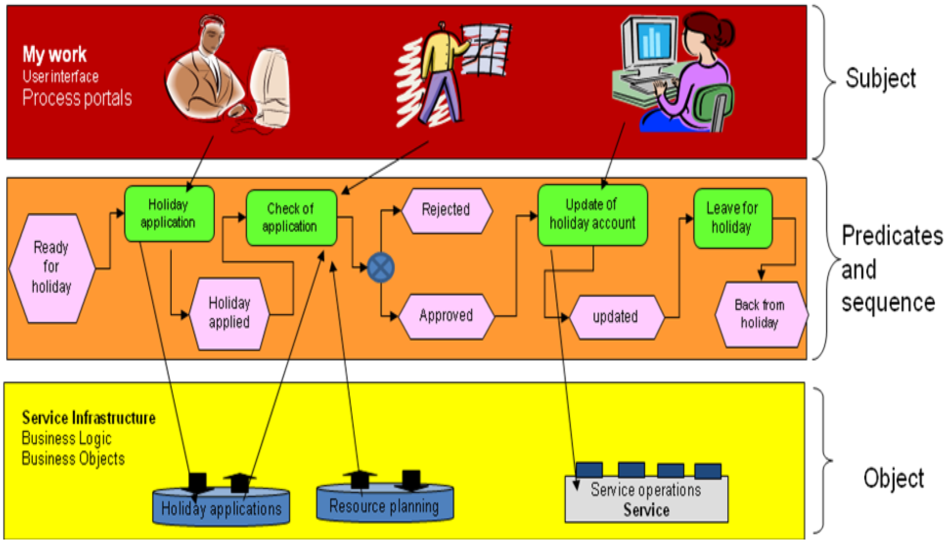


Figure 26. Orchestration and SOA

Using jPASS! (www.jcom1.com) operations can be used to provide meaningful subject-specific steps. Such services in use are triggered in a synchronous way. Hence, a subject switches to a subsequent state once all concerned services have been completed. This approach corresponds to orchestration. In the corresponding tool jPASS! subjects also comprise users besides technical elements. The services in use can either represent complex business-logic elements or elementary reading or writing operations.

The services assigned to the nodes might also contain user in/output as a part of their processing scheme. In this way users become part of the service of a process. A service can provide data sent via messages to other subjects, e.g., implemented by form-based user interaction. For instance, such an internal function would allow a user to fill in a holiday application form. The service meets the technical requirement to provide the relevant data for the holiday application. User inputs can also be combined with calculated or extracted information from data repositories. User services can be standardized constructing (process) portals that might serve as controller component for process or service execution.

Subjects themselves can be also considered services. A subject can request a service from another subject. Subjects can be considered as a service provider producing the requested services in a self-contained way. In these cases subjects offer so called ‚request services’ requested by messages. The requesting service might continue the service provision when a service provider accomplishes the requested task. The requested results delivered by the requested subject are accepted by the requesting service when suitable. Figure 27 visualizes the various aspects of subject-oriented process description, service-oriented architecture (SOA), and resulting portals.

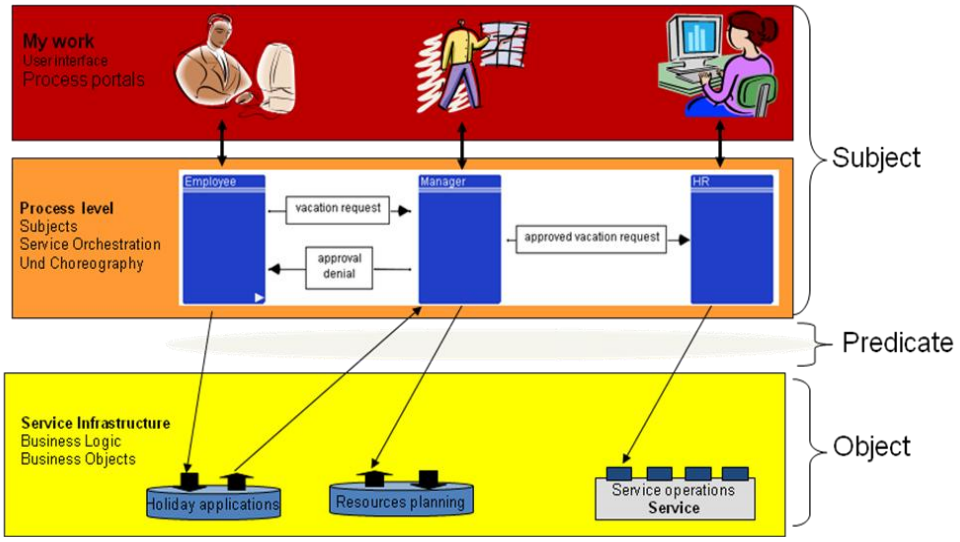


Figure 27. jPASS! and SOA

In the business context portals offer a user-specific view on information and relevant business processes. The subject-oriented perspective of processes reflects for all users their relevant part(s). By allocating subjects to users (stakeholders) all portal processes or services can be identified that are relevant for individual task accomplishment, streamlined with the task procedures of other members of the organization.

5. Conclusions

Organizations have to act and react according to environmental changes and internal developments. An inappropriate (re)action might lead to problematic situations. In our research we look at computer workplaces, the stakeholders' knowledge and the resulting potential for individual and organizational learning. If each person is enabled to adapt business processes, data and artefacts according to his/her needs and role characteristics, profound knowledge of the organization – seen as incorporation of all its members – can be stored and made tangible via workflow generation. It can also be challenged and modified in a participatory style of organizational development.

Our a research goal was and still is supporting stakeholders in a way that they can (re)define business processes according to their view on the organization of work in a consensual way. In doing so, they are able to contribute to organizational learning actively. The proposed subject-oriented representation and execution support captures the structure and dynamics of task accomplishment addressing actors (roles), problem domain data (objects), and interaction modalities (message exchanges) for task accomplishment. Each specification can be visualized and directly experienced by all involved parties.

What still needs to be done is to perform cross-organizational field studies to evaluate the impact of actor-specific and communication-driven workflow generation for networked organizational learning processes. It has to be investigated how stakeholders can be motivated to adapt business processes according to their experience in cross-organizational networks, and start organizational learning processes. A matter of particular interest is the observation of the evolution of business processes in the organizational memory as well as of the impact the individual views have on the organizational changes of the business processes.

References

1. Abecker, A., Hinkelmann, K., et al. (2008). *Geschäftsprozessorientiertes Wissensmanagement*. Heidelberg: Springer.

2. Ackermann, M. (1994). Definitional and Contextual Issues in Organizational and Group Memories. *Proc. HICSS*'94, IEEE.

3. Brown, J., & Duguid, P. (1998). Organizing Knowledge. *California Management Review, 40*(1), 90–111.

4. Chakraborthy, D. (2004). Extending the reach of business processes. *IEEE Computer, 37*(4), 78–80.

5. Davenport, T. H. (1993). *Process innovation: Reengineering work through information technology.* Boston, MA: Harvard Business School Press.

6. Davenport T., & Prusak, L. (1998). *Wenn Ihr Unternehmen wüßte, was es alles weiß.* Landsberg/Lech: Moderne Industrie.

7. Dittmar, A., Forbrig, P., Heftberger, S., & Stary, C. (2004). Tool support for task modelling – A 'constructive' exploration, *Proceedings EHCI-DSVIS'04*, 11th International Conference on 'Design, Specification, and Verification of Interactive Systems', Hamburg, *Lecture Notes on Computer Science, Vol. 3425,* Springer, pp. 59–74, 2005.

8. Ehrig, M. (2007). *Ontology alignment*. Bridging the semantic gap. Berlin: Springer.

9. Fleischmann, A., Lippe, S., & Meyer, N. (2009). Coherent Task Modeling and Execution based on Subject-oriented Representations. Proceedings *TAMODIA'09,* Task Modeling and Diagrams for User Interface Design, Berlin: Springer.

10. Haeckel, S. S. (1999). *Adaptive enterprise: Creating and leading sense-AND-respond organizations.* Cambridge, MA: Harvard Business School Press.

11. Havey, M. (2005). *Essential business process modelling.* New York: O´Reilly.

12. Heftberger, S. & Stary, C. (2004). *Partizipatives organisationales Lernen – Ein prozessorientierter Ansatz*. Wiesbaden: DUV.

13. Herrmann, T. (1998). Sozio-orientierte und semi-strukturierte Modellierung mit SeeMe, *Proc. MobIS´98*. Informationssystem Architekturen. Rundbrief des GI-Fachausschusses 5.2. 5. Jg., Heft 2. (pp. 15–22).

14. Herrmann, T. (2000). *Lernendes Workflow aus: Verbesserung von Geschäftsprozessen mit flexiblen Workflow-Management-Systemen* (Band 4). Heidelberg: Physica.

15. Havey, M. (2005). *Essential Business Process Modeling*. Beijing: O'Reilly.

16. IDS Scheer AG. (2001). *Handbuch zur ARIS-Methode.* Version 5, Saarbrücken.

17. Kolb, D. (1984). *Experiential learning: experience as the source of learning and development*. Englewood Cliffs, NJ: Prentice-Hall.

18. Kühn, O., & Abecker, A. (1997). Corporate Memories for Knowledge Management in Industrial Practice: Prospects and Challenges. *Journal of Universal Computer Science, (3).*

19. Laudon, K.-C., & Laudon, J. P. (2005). *Essentials of Management Information Systems.* Managing the Digital Firm, 6th edition, Upper Saddle River, NJ: Pearson.

20. Lewis, M., Young, B., Mathiassen, L., Rai, A., & Welke, R. (2007). Business process innovation based on stakeholder perceptions, *Information Knowledge Systems Management*, 6, 7–17.

21. Nonaka I., & Takeuchi, H. (1995). *The knowledge-creating company*. New York: Oxford Univ. Press.

22. Preece, J. (1993): *A guide to usability, human factors in computing*. New York: Addison-Wesley.

23. Probst, G. et al. (1997). *Wissen managen.* Wiesbaden: Gabler.

24. Stary, C. (2000). TADEUS: Seamless Development of Task-Based and User-Oriented Interfaces, *IEEE Transactions on Systems, Man, and Cybernetics,* 30, 509–525.

25. Senge, P. (1990). *The fifth Diszipline, The Art and Practice of the Learning Organization*. New York: Doubleday.

26. Rouse, W. B. (ed.) (2006). *Enterprise transformation: Understanding and enabling fundamental change*. Hoboken, NJ: Wiley.

27. Rumpfhuber, M., Fichtenbauer, C., & Stary, C. (2002). Sprachgerechte unternehmensnahe Modellierung von ereignisgesteuerten Prozessketten – Zur adäquaten Aus- und Weiterbildung von ModelliererInnen. in: *Proceedings EPK-2002*, Workshop 'Ereignisgesteuerte Prozessketten', GI, Springer Lecture Notes, (pp. 109–118)..

28. Scheer, A. W. (2001). *ARIS - Modellierungsmethoden, Metamodelle, Anwendungen(4th Edition).* Berlin: Springer

29. Stephenson, S. V., & Sage, A. (2007). Architecting for enterprise resource planning. *Information Knowledge Systems Management, 6*, 81–121.