Albert Fleischmann
Stefan Oppl · Werner Schmidt
Christian Stary

Contextual Process Digitalization

Changing Perspectives – Design Thinking – Value-Led Design

Springer Open

Contextual Process Digitalization

Albert Fleischmann • Stefan Oppl •
Werner Schmidt • Christian Stary

Contextual Process Digitalization

Changing Perspectives – Design Thinking – Value-Led Design

Springer Open

Albert Fleischmann
Dr. Albert Fleischmann & Partner
InterAktiv Unternehmensberatung
Pfaffenhofen a.d.Ilm, Germany

Stefan Oppl
Department for Continuing Education Research
and Educational Technologies
Danube University Krems
Krems, Austria

Werner Schmidt
Technische Hochschule Ingolstadt
Business School
Ingolstadt, Germany

Christian Stary
Department of Business
Informatics-Communications Engineering
Johannes Kepler University Linz
Linz, Austria

ISBN 978-3-030-38299-5 ISBN 978-3-030-38300-8 (eBook)
https://doi.org/10.1007/978-3-030-38300-8

This book is an open access publication.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

"The most important innovations are those that change our thinking."

We are pursuing this claim with this book, following the aphorism of Hans-Jürgen Quadbeck-Seeger, a German chemist. Our ingredients:

"Comprehensibility is the skill of an expert." Again, we have tried to be faithful to this aphorism. Our work should inspire everyone interested in novel forms of process design in the course of digitalization. Therefore, we attempted to keep explanations as simple as possible.

"Luxury = Cultivating the Unnecessary." We want to address all those who want to grasp the essence of processes and their utilization in practical action, without intensive modeling language and application studies, but rather underpinned with graspable concepts.

Students may appreciate the textbook character of the book, practitioners the examples, and researchers and developers the conceptual representations and theoretical achievements.

"The bigger the project, the quieter it will be buried." For more than a decade, the concept and missionary goal of renewing process management has existed. The result has led to the idea of appreciating simplicity and clarity, without neglecting complexity. As novel drivers of process management have evolved, it is time to look at the digitalization of processes from the perspective of subject orientation.

"Adventure tourists are attracted to new and exciting places." The experience is worthwhile, because it opens a view of the world that comes close to our perception of reality, conclusively extending the existing, and thus providing new space for adaptation. Our followers are also adventurers—welcome all!

Special thanks go to:

- Christoph Moser—with his insights on organizational practice
- Edith Rieß and Christoph Bawart—for their help in shaping the format
- Sabine Kathke, Heike Jung, Sybille Thelen of Springer Vieweg, and Ralf Gerstner of Springer for their support from the publishing house for implementing our ideas
- Jerome Geyer-Klingeberg of Celonis SE—for clarifying the process practice

- Richard Wright for his essential support in increasing the quality of the book. He transformed the German English of the authors into proper English. With his professional expertise and in-depth understanding of the subject matter he also contributed significantly to the readability of the book. We would like to express our sincere thanks to Richard for his achievement in improving the book.

"Innovations are not natural events, we have to seriously want and enforce them"—Ad multos multiplicatores, not only in Ingolstadt, Pfaffenhofen, Steyr, and Vienna.

Pfaffenhofen a.d.Ilm, Germany	Albert Fleischmann
Krems, Austria	Stefan Oppl
Ingolstadt, Germany	Werner Schmidt
Linz, Austria	Christian Stary

Contents

Check for updates

Motivation 1

1.1 Business Processes and Business Process Management

There is no organization without processes. When people want to collaborate, they use the necessary tools and coordinate their activities to reach the desired result. Since such activities can not only be carried out by humans, but also by machines and computers, their activities must also be included when aligning human requirements and technical capabilities. In particular, different types of actors are involved in at least partially automated processes.

A process is triggered by an event that may originate inside or outside the organization, such as a travel request or customer order. Coordinated and targeted action in response to such an event is called a process. In case the organization is a company, this is referred to as business process.

There is no company without business processes. There are only differences in their level of maturity. The reactions of an organization to certain business events can always be coordinated anew when these events occur, or a procedure is defined that is then executed in such cases. Events of the same type, such as purchase orders, are referred to as event classes. A predefined procedure for an event class is called a process model. The execution of the activity sequences defined in the model as a reaction to an identified concrete event, e.g., the book order of customer Huber from May 20, is termed a process instance.

Every company, irrespective of its type of business, has certain standardized processes that can be designed and tailored to the individual company. For instance, every company has an order-to-cash process designed to react to business events, ranging from the customer order to the receipt of payment, and to document these through booking. Conversely, a procurement process will exist with purchase orders to satisfy individual requirements, the concrete reference (for example, receipt of goods and storage), and the payment of vendors. Other examples are processes for recruitment or logistics. A common classification categorizes processes according to their character into management, core and support processes. The classification is company-specific and depends, among other things, on the industry sector.

A. Fleischmann et al., *Contextual Process Digitalization*,
https://doi.org/10.1007/978-3-030-38300-8_1

The more clearly a company defines its business processes and the more consistently it implements them in its daily operations, the more efficient it will be. For many companies, their competitiveness is not (or no longer) based solely on the uniqueness of their products, but on the quality of their business processes. For example, while a publisher's business is primarily determined by its books, at Amazon the customer experience in searching, selecting, purchasing, paying, delivering and returning products, i.e., the smooth, customer-centric process, is the key to success.

The models for such processes must be continuously adapted or completely redesigned because the reactions to an event class can change, or additional reactions to new event classes can become necessary. The resulting specifications must also be implemented in the organization and IT infrastructure so that employees can work through instances of the processes in day-to-day business. In doing so, underlying conditions, such as effectiveness, efficiency and compliance, i.e., the requirements to deliver the desired result with the lowest possible expenditure of resources and in compliance with valid external and internal regulations (e.g., laws), must be taken into account. Business Process Management (BPM) has established itself to handle these tasks. It describes an integrated management approach for analysis, design, optimization, implementation, control, monitoring, and further development of the management, core and support processes in a company. From a technical point of view, it also includes IT support for these subtasks through corresponding tools, e.g., for modeling or execution (such as process engines) or more comprehensive Business Process Management Systems (BPMS).

In Business Process Management, a company and its immediate environment are regarded as a selected part of reality for modeling and executing. In this dedicated part of the world, one party wants a deliverable from another party in the form of a physical product, a service, or a combination of both. The deliverable should be provided in accordance with associated requirements; the desire for it is the business event to which the company should react as perceived in the defined process model.

In Business Process Management, it is therefore necessary to define a model for the provision of services and apply it to the processing of business cases. This means adapting reality according to the model, i.e., analyzing affected sections of reality and changing this reality. Since this reality and the desired changes are very complex, several modeling concepts from the social sciences, business administration and computer science are brought together and combined in BPM.

In the following sections we outline an overall view of process management and then explain it in detail in the succeeding chapters. From the perspective of the participants on the world, the various facets of Business Process Management are presented, and a selection of models are introduced which have turned out useful in our practice. The design of such models supports the transition from a more or less unstructured or unsatisfactory way of working to a structured process handling that corresponds to the ideas of a company and its customers.

We develop the overall view step-by-step, starting from the individual perspectives of the participants on their work in a process, its structuring and harmonization, then moving forward with the specification in a model and its embedding in the organizational and IT environment of the company and finally culminating in the joint processing of process instances in the resulting socio-technical systems. A corresponding illustration which grows with this overall view ultimately shows our comprehensive understanding of Business Process Management.

1.2 View of the World, Structuring and Modeling

As already mentioned, it is important for a company to identify the business events of interest and to define the activities triggered by them. For this purpose, the corresponding extract of reality must be identified and examined more closely.

This extract is determined by the customers who demand a service and, for the group of employees of the company involved in the provision of the service, it represents the reality which directly affects and surrounds them. In order to provide the desired service, the parties involved must cooperate directly or indirectly.

Everyone makes their contribution in coordination with the others. Based on their personal background in terms of education, knowledge, motivation, experiences and preferences, each group member has his own perception of the process and its context. He develops his individual idea of what his contribution should be, how it is provided, which events with which activities need to be considered and by whom, in which order partial steps take place, which preliminary services are expected by whom and for whom preliminary services are provided.

As a result, all affected people possess their own mental "world model" of the extract of reality under consideration (cf., Fig. 1.1). For a successful reaction to business events, it is necessary to structure the different realities of the participants and to transform them into a consistent process model for joint, goal-oriented action. This means that the business process is "agreed upon" by harmonizing the individual, to a greater or lesser extent matching, mental models of the people involved.

This joining of the individual ideas of those affected by a business process and the mutual coordination of the different aspects of a business process (cf., Section 1.3) is itself a complex process and the central aspect of BPM.

1.3 Components of a Process Description

We split a business process description conceptually into three parts (see Fig. 1.2). The first part, called **process strategy,** makes statements about the purpose, triggers, inputs, end and outputs of the process. The trigger is the event that sets the service provision in motion on the basis of the initiator's expectations, i.e., generates a process instance. This impulse is accompanied by the fact that the initiator provides information or objects which are to be processed according to his expectations. These inputs must be transformed into the expected results and made available to the defined recipient. In this way, the business process creates a value for which a customer pays.

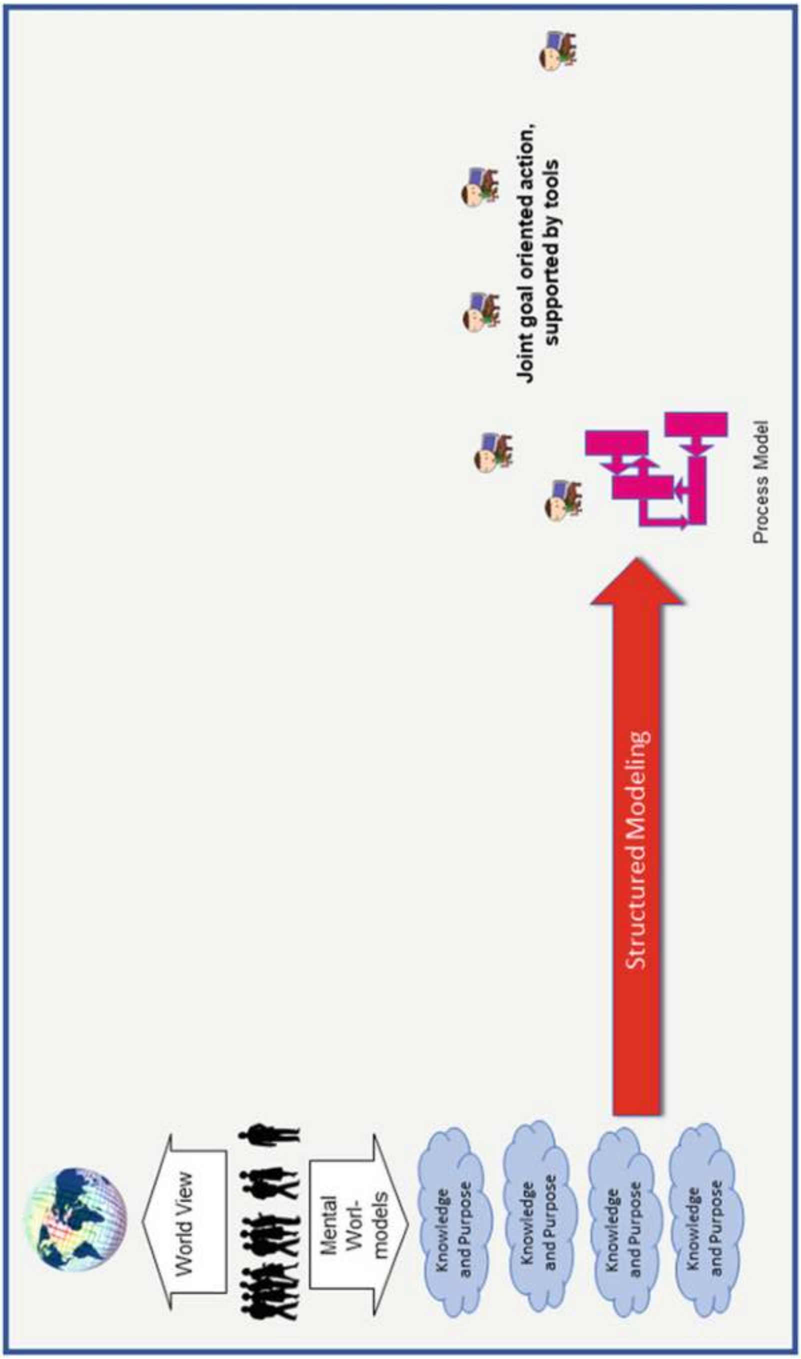


Figure 1.1: Individual mental models of the participants

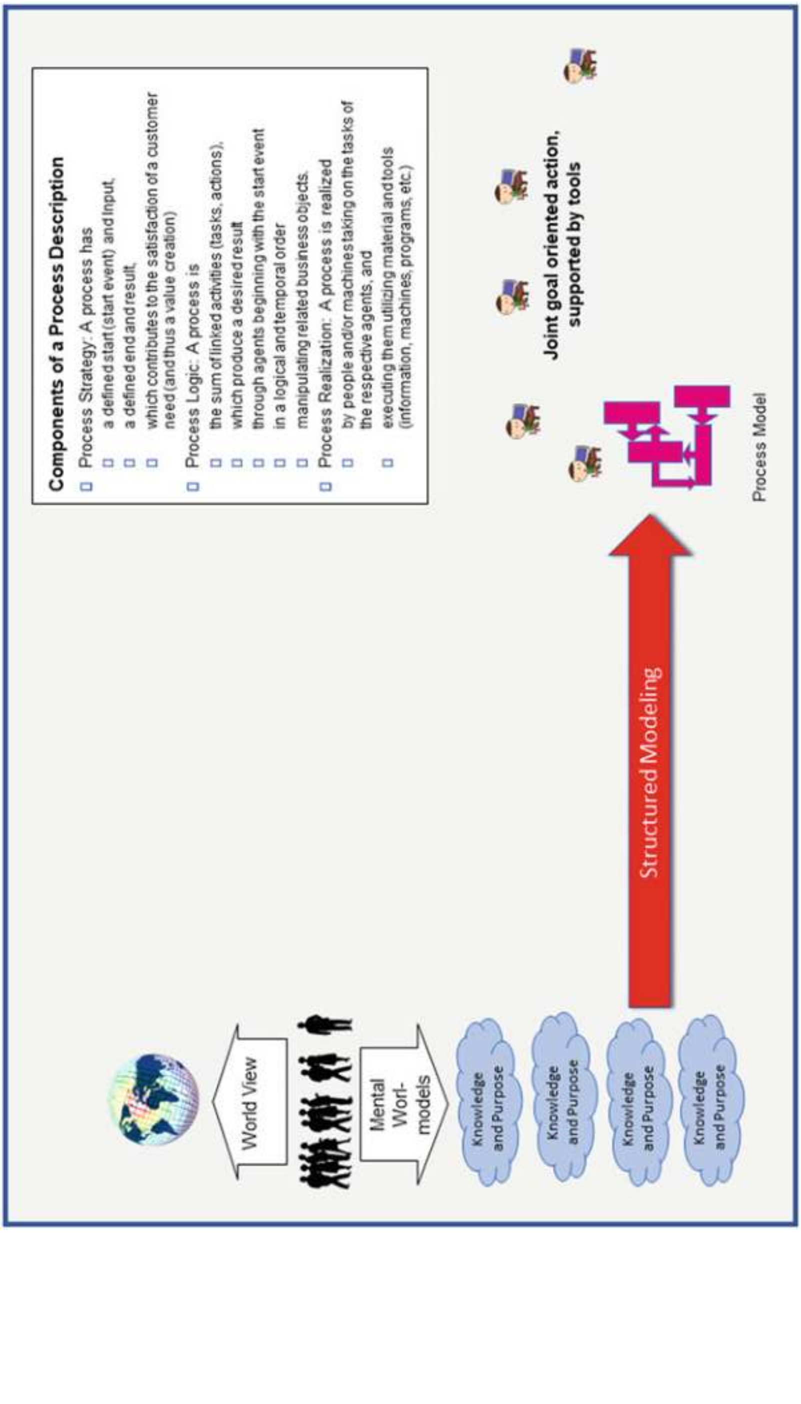


Figure 1.2: Components of a process description

This external view of a business process is supplemented by the **process logic**. This inner perspective describes the actors involved and their coordinated interaction. The actors carry out activities in a logical and timely meaningful order. They transfer the results of their actions to other actors for further processing, or to the intended recipient at the end.

Process implementation involves the provision of resources for the processing of process instances. These can be humans, machines and software systems, which take over the activities assigned to them as concrete realizations of the involved persons. In the age of digitalization, software systems (process or workflow engines) synchronize the actions of the actors by controlling the temporal and logically necessary sequence of the sub-steps according to the process model. For the handling of their individual tasks, the actors can use aids such as information, application programs or tools where required.

Throughout process realization, it must be ensured that several process instances can be executed in parallel and independently of each other on the basis of the defined exemplary model through appropriate resource allocation.

1.4 Determining Factors for Process Models and Process Instances

The business model essentially describes how a company affects the world and how it thereby generates revenues and profits. The customer promise as well as the resources and partners with whom this promise is fulfilled are essential.

The enterprise architecture describes a machinery with which the business model is to be brought to life. As a typical layer concept, it defines business and IT structures and links them together. The concept of Business Engineering, for [1] example, envisages the business architecture on a strategic level with the definition of goals and services that are interwoven with the business model. At the level of the processes, as implementation tools of the strategy, the process architecture follows with its organizational and operational structure. The transition to the IT structures for supporting the processes leads to the level of information systems with the application architecture and the IT architecture.

As a central component of an enterprise architecture, business processes are therefore in a kind of sandwich position that illustrates how they are influenced by other architectural elements. For example, a given organizational structure that is difficult to change can influence the procedures in processes and the way in which a company works together with external partners. The same applies to the availability of resources. But horizontal dependencies within the process organization must also be taken into account, e.g., whether a certain way of working in the ordering process has an effect on the design of payment processing.

The underlying technological infrastructure not only affects the content design of the process models, but also the level of detail and accuracy. For the development of IT solutions for process digitalization, rigorous requirements apply to the model definition. Process parts that are to be executed with IT support must be specified precisely.

In addition to the internal determining factors explained above by way of example and supplemented in Figure 1.3, external factors also have an impact on process design. Here one can see as an example test steps which have to be included in a process due to compliance regulations.

1.5 Process Metrics

The processes to be developed or changed have the general goal of supporting the implementation of the business model and the associated strategy. The relationship between the Key Performance Indicators (KPIs) from the business model and the processes is established using Process Performance Indicators (PPIs). These Process Performance Indicators are refinements of objectives from the business model (cf., Fig. 1.4).

Typical business Key Performance Indicators are derived from business models and strategies and measure business success at higher aggregation levels, e.g., revenues and costs at the overall company, division, product group level, etc. The focus here is on effectiveness ("Doing the right things"). The business processes are used to implement the strategy and bring together the elements of the enterprise architecture. The associated Process Performance Indicators aim at efficiency ("Doing things right"). They are therefore closely related to the Key Performance Indicators and are partly derived from them.

When deriving the performance indicators, it must already be checked whether they can be measured with sufficient precision and justifiable effort. Under certain circumstances, this may also place demands on the process to be developed in order to be able to measure the performance indicators directly or indirectly. If direct measurement is not possible, targets for alternative performance indicators can also be defined and values for the performance indicator actually desired can be derived from them.

Target values are defined for the Process Performance Indicators, which are to be achieved by a changed or redesigned process. Throughout the entire process, from the identification of the problem to the implementation of a modified or new process, it is important to constantly check whether the desired goals can be achieved with the resulting process.

1.6 Support Concepts

The path from individual knowledge and willingness, i.e., from the mental models of the participants, to a process model that can at least in part be digitalized, is complex and costly. In order to reduce complexity and effort, support concepts such as frameworks, process models and description languages were developed.

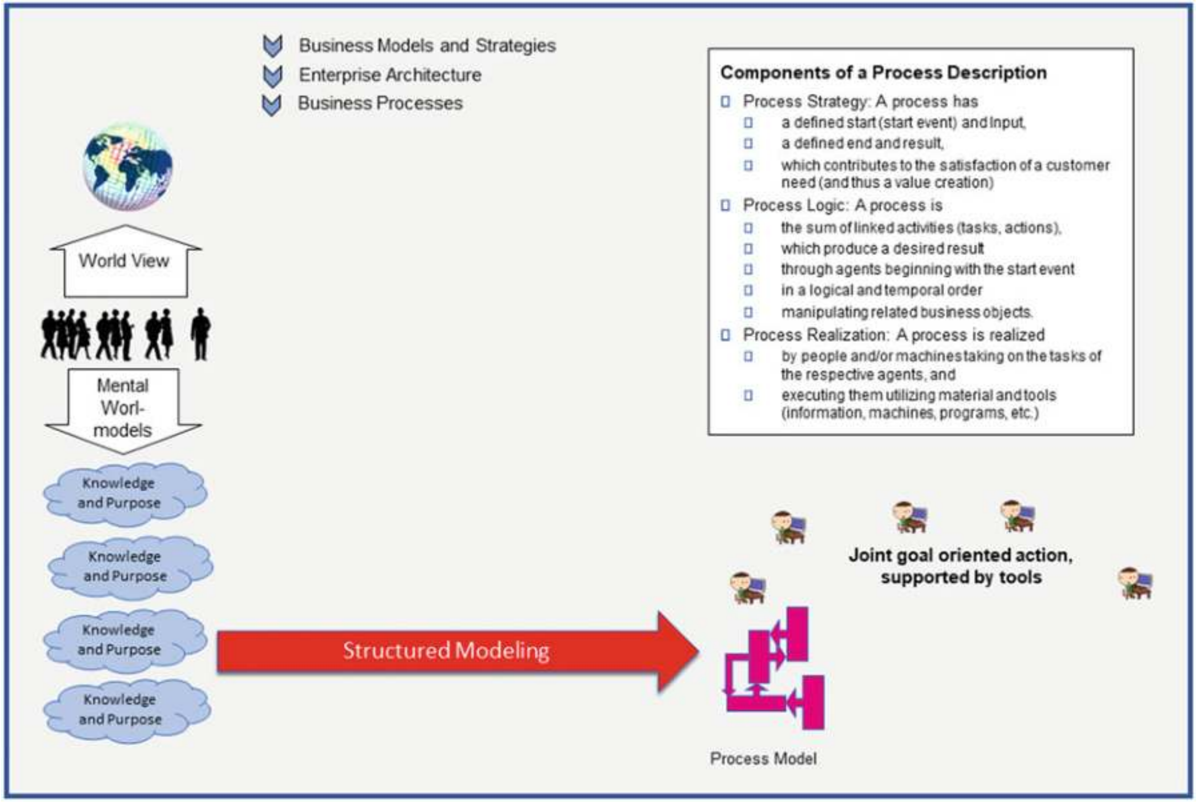


Figure 1.3: Addition of determining factors for process definition

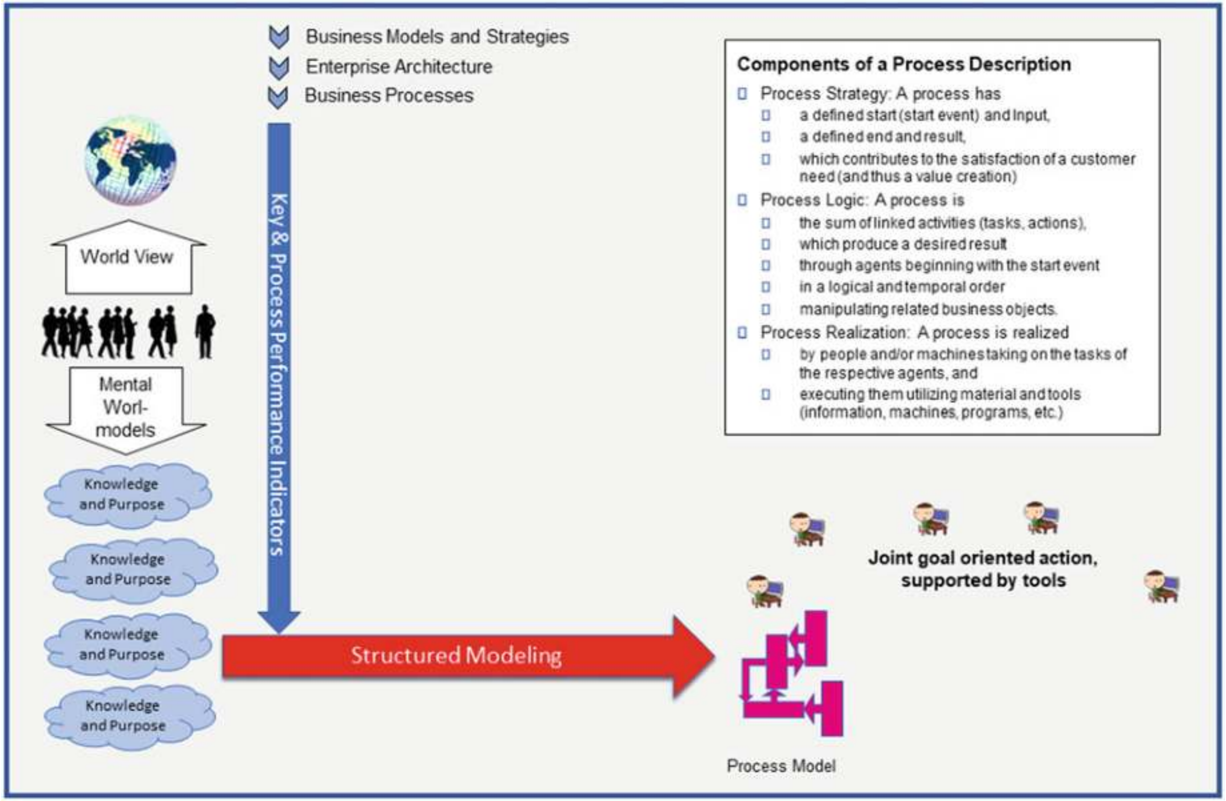


Figure 1.4: Definition of Process Performance Indicators

The following overview comprises a thematically grouped selection of such tools, which according to our experience are widely used in practice. They are inserted in Figure 1.5 and are discussed in more detail in the chapters on models (Chapter 2) and modeling languages (Chapter 3).

Frameworks for Quality Management:

- Total Quality Management (TQM) TQM/PDCA,
- Deming Cycle (PDCA, Plan-Do-Check-Act)
- EN ISO 9001
- European Foundation for Quality Management (EFQM)

Frameworks for Enterprise Architecture Management (EAM):

- Zachman Framework
- The Open Group Architecture Framework (TOGAF)
- Architecture-Animate (ArchiMate)

Frameworks for IT management and IT governance:

- IT Infrastructure Library (ITIL)
- Control Objectives for Information and Related Technology (COBIT)

Description languages for process logic:

- Flowcharts
- Event-controlled Process Chains and extended Event-controlled Process Chains (EPC, eEPC)
- Business Process Model and Notation (BPMN)
- Subject-oriented Business Process Management (S-BPM)

1.7 Digitalization

Today, digitalization is the key word in the transformation of value creation. Digitalization in the economy or in organizations in general means digitalization of business models, products and services as well as of whole processes or parts thereof. For processes, however, this does not necessarily mean full automation without any human intervention. For example, a program that controls a process may, if necessary, include actions executed by humans or by Cyber-Physical Systems. The latter consist of communicating devices with software as well as mechanical and electronic components. In the Industry 4.0 Initiative, the aim is to achieve this comprehensive consideration of processes, i.e., the communication between people, machines and workpieces. On the one hand, these aspects must

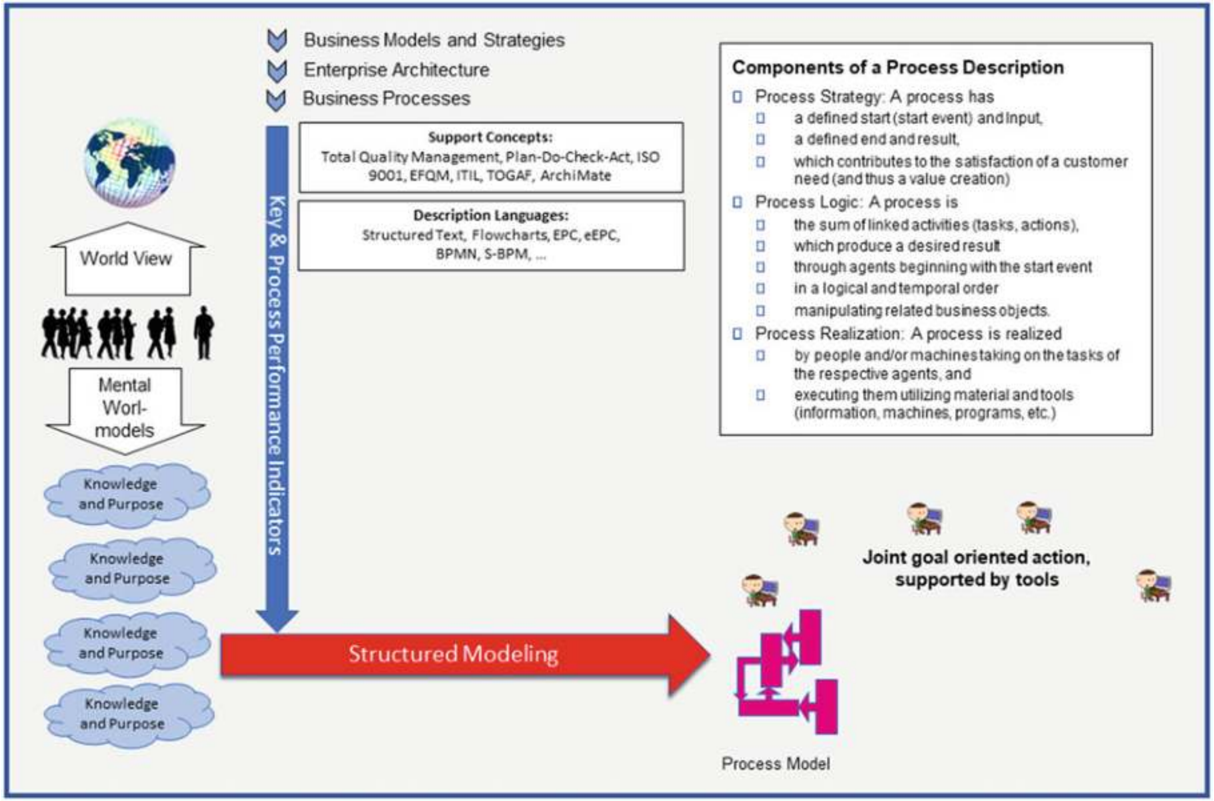


Figure 1.5: Addition of concepts to support the process definition

be expressed in the process models, and on the other hand, the transfer of a business process model into digital execution must be supported as far as possible. Particularly when aspects of quality management, i.e., the continuous improvement of processes, are taken into consideration, it must be possible to implement process changes that entail a change in digitalization quickly and with as little effort as possible.

The aspects described in the previous sections must already be included in the creation of the models in order to facilitate the technical implementation of processes, but without already anticipating implementation details (cf., Fig. 1.6). The more precisely the processes are described, the easier this task becomes. Process segments whose flow logic cannot yet be precisely described at the time of modeling must be marked accordingly. However, these parts of a process can be modeled with other suitable methods according to the desired or necessary candor. Such process segments can either be described with Adaptive Case Management methods or, if a communication-oriented description language is used, as a communication loop. The latter is terminated by one of the partners involved after a corresponding result has been achieved, before continuing the process.

Important in this context is the granularity, i.e., the level of detail of the process description. Activities should be broken down in such detail that one can clearly determine whether they can be digitalized, partially digitalized (human IT, physical IT), or are performed manually by humans. The tailoring should be based on the business requirements and not on the functionality of a potentially already existing IT system. If necessary, such a system must be adapted to meet the needs of the desired business specification during process implementation.

1.8 Process for Creating Processes

The definition of the business processes cannot be done schematically or algorithmically, i.e., there is no software that when fed with the business model, the enterprise architecture, and the Key Performance Indicators with associated target values and support concepts, delivers a suitable process description directly.

The definition of business processes is an intuitive and creative process. Therefore, creativity techniques and knowledge management methods such as Storytelling, World Café or Value Networks are also used, especially at the beginning of Business Process Management activities.

For example, one can use the Design Thinking approach. This is a concept in which interdisciplinary teams work together in an iterative process in an environment which fosters creativity to develop innovative solutions to a problem (see section 5.3). A key point thereby is to develop and consider an in depth understanding of the needs and motivations of people in the target group. Design Thinking offers a comprehensive collection of methods for use in the individual steps of the approach. With these characteristics, it can also be used for the revision or redefinition of a business process. Under certain circumstances, extraordinary solutions can be found that would not have been possible with the usual BPM approach.

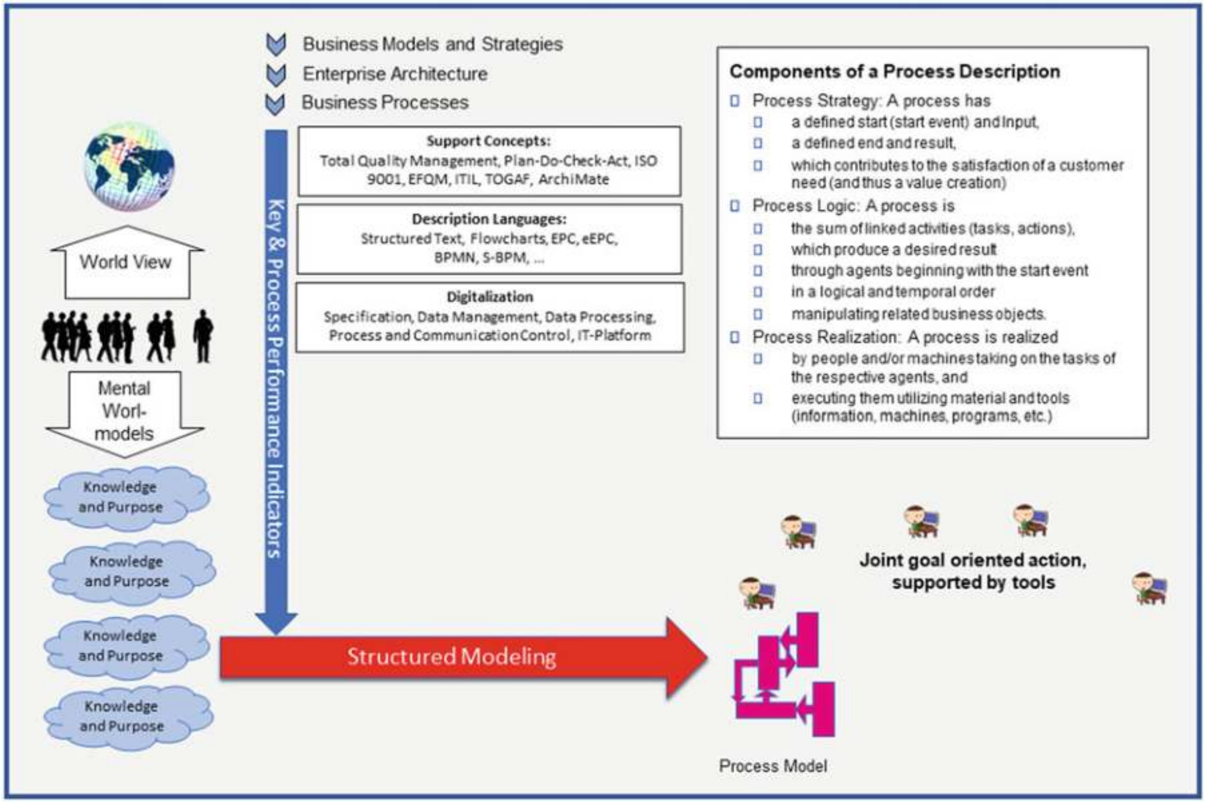


Figure 1.6: Consideration of digitalization aspects in the model

However, a creative, innovative process concept must also be devised and implemented in detail. Creative design is therefore embedded in a bundle of activities that ultimately makes the process part of the real world. As such activity bundles, we identify analysis and modeling, validation, optimization, organizational implementation, IT implementation as well as operation and monitoring. These activity bundles are a further development or refinement of the Plan-Do-Check-Act cycle. They are usually arranged in a circle, which implies a corresponding flow. This does not always correspond to reality, which is why we present the activity bundles in Figure 1.7 as loosely networked honeycombs (cf., Fig. 1.7). There, the phases of the Design Thinking process and the activity bundles are supplemented. Both concepts are presented in more detail in Chapter 4 and put into relation with each other.

Extensive and complex process changes usually require activities from several activity bundles and are carried out as a project. Such a project can thus be regarded as an iteration (process instance) of the process for creating business processes. For this, a detailed project plan must be created with the activities to be carried out, responsibilities and deadlines (cf., Fig. 1.7). The project plan should then be executed according to the methods of project management.

1.9 Organizational and Technical Implementation

Once the process model has been created, the model must be embedded in the organizational structure of a company. This determines which activity is performed by which person or organizational unit. This mapping does not have to be static, but can vary from instance to instance. For example, the purchasing process can have the same flow logic for parts A and B, but a different purchasing department is responsible for purchasing parts A than for parts B. Process instances for parts A thus affect other organizational units (and persons assigned to them) than for parts B. These rules must be mapped in such a way that a process is correctly linked to the organizational structure.

In addition to activities performed by people, there may also be activities in the process that execute application programs or IT services. For this purpose, such actions must be mapped in the process model to functions of software modules, which then execute them at runtime. If during the process modeling attention was already paid to the possible digitalization, this mapping is more or less unproblematic.

Software can also control the processing of process steps and assign the tasks specified in the model to the respective persons or IT services as actors. Software systems that support this are also referred to as workflow systems (process engine, workflow engine). Ideally, process descriptions can be transferred directly into workflow systems.

After being embedded into the organization and the IT environment, a process can be used for the handling of instances, i.e., real business cases - the goal is achieved. Figure 1.8 shows the now completed path from the individual mental

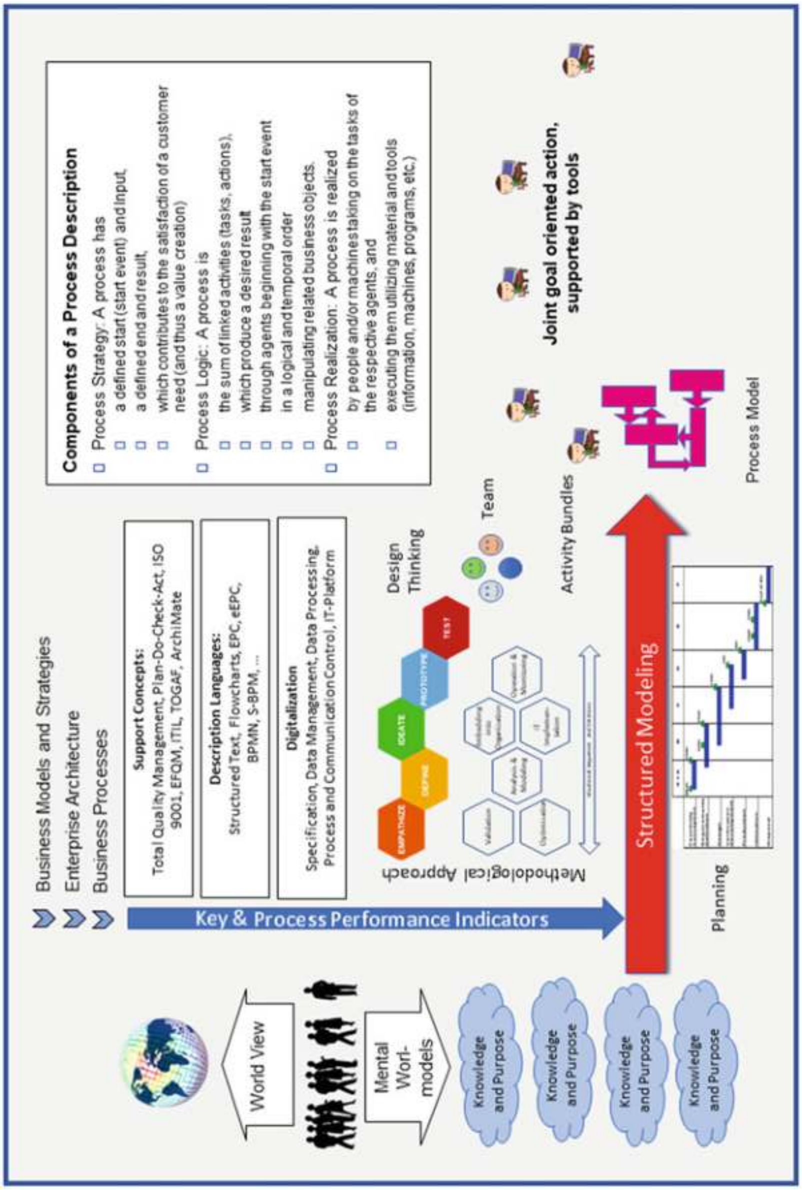


Figure 1.7: Supplementing with process structure and planning for process changes

models, including knowledge and intentions of participants, to the joint handling of process instances.

1.10 Success Measurement with Performance Indicators

When instances of a process are executed, one can check whether the target values defined for the Process Performance Indicators are reached. For this purpose, actual values for the defined Key and Process Performance Indicators (KPIs and PPIs) are measured, calculated, stored and compared with the target values. This comparison can be made in real time or over longer time intervals. Any real-time evaluation leads to the immediate initiation of suitable countermeasures in the event of a deviation from the target. An evaluation of measured values over a longer period of time, on the other hand, shows medium to long-term trends in performance indicators and can trigger corresponding changes. Evaluation results are visualized in process cockpits, among other things (cf., Figure 1.9).

1.11 Continuous Improvement

Processes are not static but are rather subject to changes in the internal and external determining factors described in Section 1.4. Developments such as business model modifications, new competitors, technical progress or deteriorations in measured PPIs, such as lead time, may require adjustments to a process. To this end, appropriate measures should be taken within the framework of the bundles of activities and procedures presented in Section 1.8.

The feedback arrow in Figure 1.10 indicates that the participants may again have diverging views of the selected part of reality. By harmonizing them in the way described, a new instance of the process for creating processes is started.

Continuous improvement is a very important aspect of process management. Ongoing adaptations bring one closer to the desired process. However, changes in the environment can influence this convergence. The state being pursued is therefore a 'moving target'.

1.12 Corporate Governance and Business Process Management

Corporate governance as an institution shapes a company. It has a decisive influence on the business model, corporate strategy, and organization. The business model and strategy are designed to open up future potential for success and thus, secure the sustainable existence of the company. The enterprise architecture creates the infrastructure to exploit the potential for success. The business processes and the business objects (data) processed by them link the business and technical levels of the enterprise architecture.

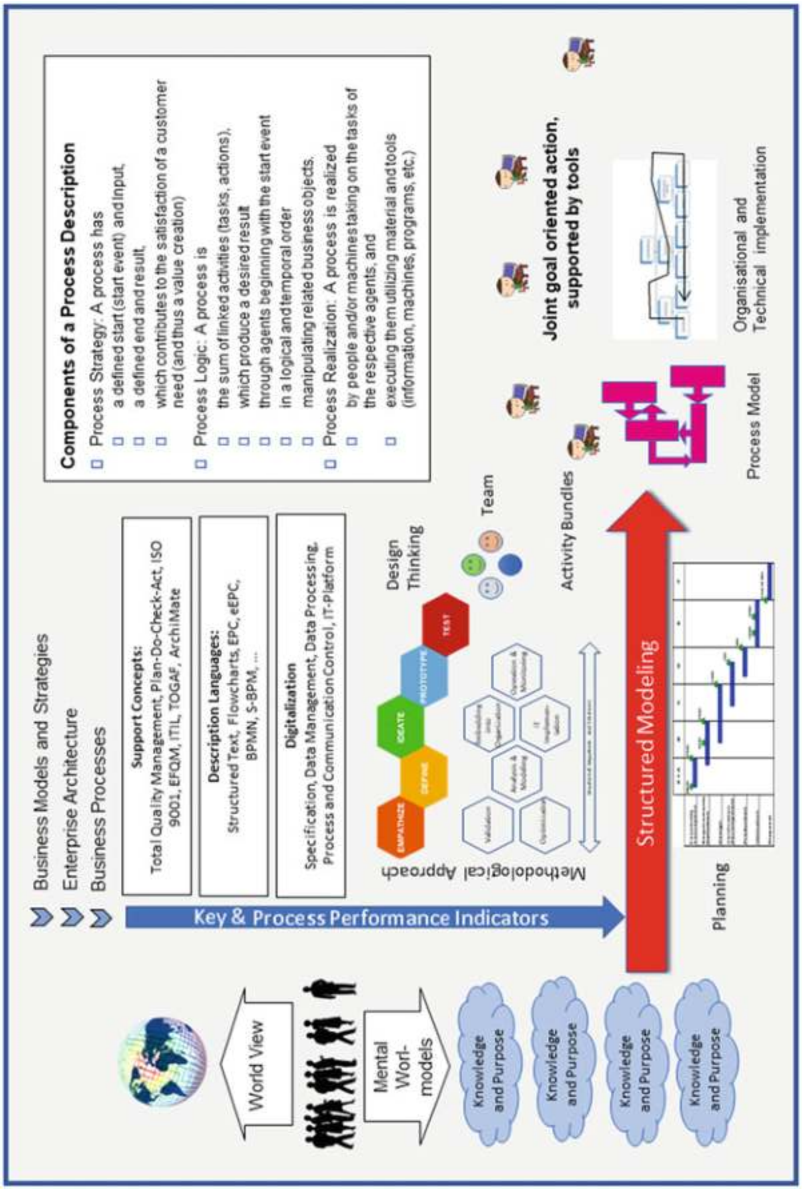


Figure 1.8: Addition of embedding in the organization and IT environment

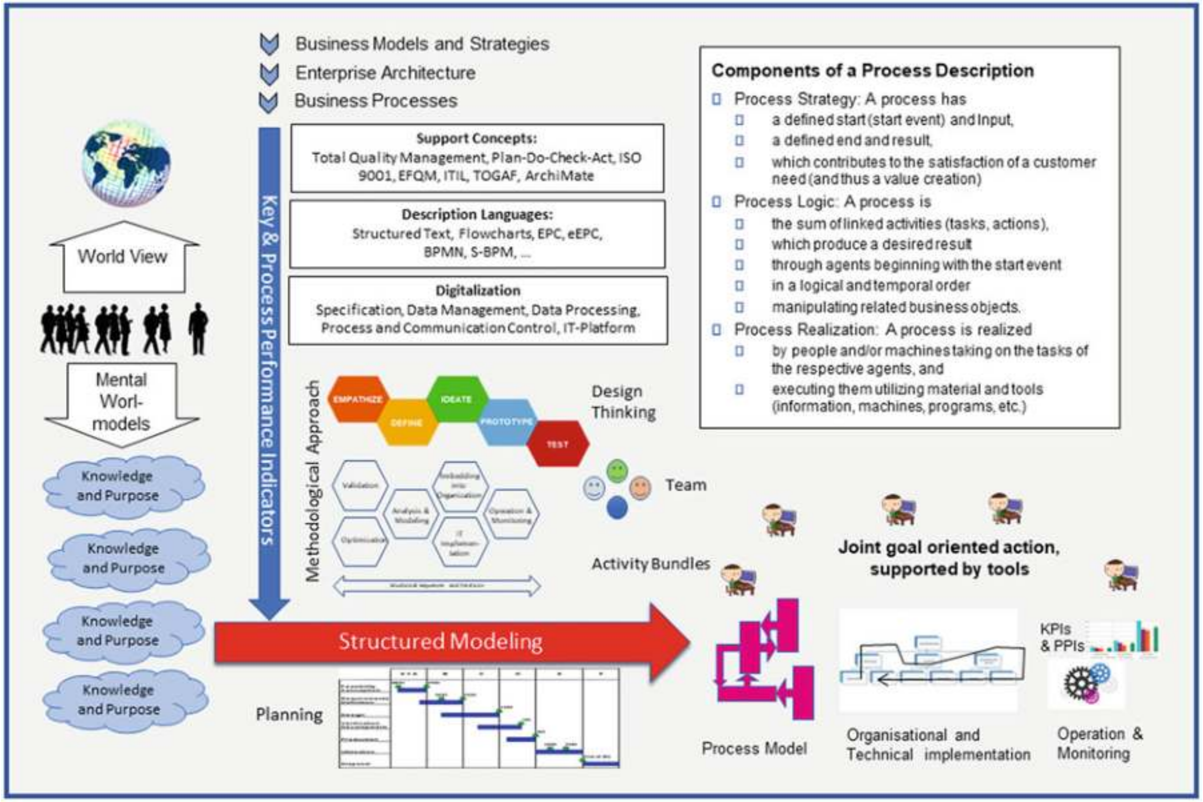


Figure 1.9: Operation, monitoring and performance indicators

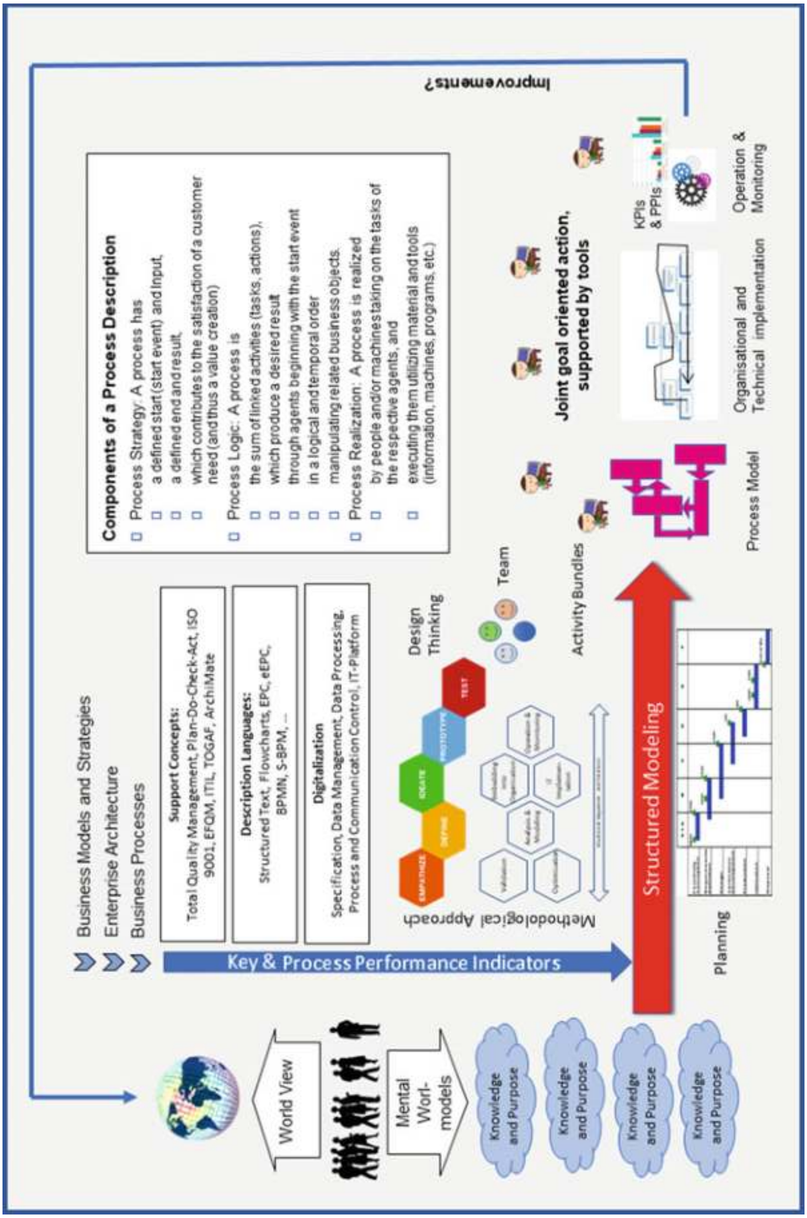


Figure 1.10: Supplementing with continuous improvement

The business processes are the subject of digitalization, i.e., the IT support of process execution by people and machines. In recent years, the associated requirements have increased significantly. In business processes, not only people and IT systems, but also "smart" machines and devices should be able to interact. This refers to highly integrated business processes in the context of Industry 4.0 and the Internet of Things, which integrate human actors as well as individual devices and machines into a common whole. The technical players are often referred to as "smart" or "intelligent".

Corporate governance as a process describes the management activities involved in creating and exploiting the potential for success. In the context of BPM, this means the management of socio-technical systems with people that are involved in processes, and machines that support people in their activities or autonomously carry out a chain of activities.

Despite the increasing importance of digitalization, the human being, as designer of socio-technical systems and user of supporting technology, is at the center of process management. Not least due to increasing agility requirements, the goal today is for employees to be able to design (model) the operative processes autonomously and independently as far as possible, and for these to then be directly supported by Information Technology without significant delays and additional effort. With a clear commitment to process orientation, management must create the conditions for this as such ("Tone from the top"). These include both the necessary infrastructure and an environment that encourages people to become actively involved in process management activities. The degree of employee involvement is determined by the image of humanity and the associated management philosophy of the company management ("Tone at the top"). In a classical, more hierarchical approach, people and their skills are seen as a resource that is the subject of managerial action, and these people ultimately execute instructions. Such a management philosophy is characterized by direct intervention of the company management and follows Theory X. According to this theory, any lack of motivation is countered by the threat of sanctioning by the company management.

In a more systemic, i.e., holistic approach, such as the one that the St. Gallen Management Model is based on, a system is to be created that works itself largely independently on the design of a Business Process Management System. All employees should be able to actively contribute. This management style follows the image of humanity according to Theory Y. According to this theory, the essential characteristics of humans are pleasure in demanding work, self-discipline, responsibility and intellectual power.

The image of humanity is supplemented by corresponding organizational theories. The purpose of these is to explain the creation, existence and functioning of organizations. Organizational theories implicitly assume a corresponding view of humanity. Thus, Taylorism is more based on an image of man that corresponds to Theory X. Luhmann's Systemic Organizational Theory, on the other hand, makes no ethical assumptions about the people in an organization; it only assumes that they communicate. Although the Theory of Communicative Action also focuses on communication, the world is to be changed through theory and rationality. It is assumed that man is by nature insightful and open to argumentation.

There are management philosophies and organizational theories to match the various concepts of humanity. The nature and use of methods, techniques and tools must be consistent with them. For example, it is not appropriate to propagate the involvement of employees if the company management then does not take their suggestions seriously or does not take notice of these suggestions at all. Before it starts designing processes, a company should therefore be aware of the image of humanity that shapes its leadership and corporate culture.

We think that especially for the challenges associated with digitalization, an orientation to Theory Y is necessary, which will often (and must) lead to cultural change in practice.

Reference

1. Österle, H., & Winter, R. (Eds.). (2003). *Business Engineering* (2nd ed.). Berlin: Springer.

2 Models

In the previous chapter we outlined the various aspects of Business Process Management. Since models are often used in practice to describe these aspects, the following sections will first look in more detail at the tasks and properties of models and modeling. We then present examples of models from various areas of expertise that in our experience have proven helpful in numerous Business Process Management projects. There is no claim to completeness for this exposition, but it can serve as an orientation for the reader when needing to choose description models for individual project needs.

2.1 Model and Reality

"You don't have to understand the world, you just have to find your way around in it." According to internet sources, this sentence is attributed to Albert Einstein. Who understands what is going on in the world? Who knows how it works? Therefore, we should take care of our world, namely the part of the world that is important to us at the moment. We should recognize that we create or construct our world on a daily basis. Any excerpt of reality is naturally determined by our subjective interests. We decide which part of the world we want to consider and which aspects seem important to us.

In doing so, we identify the artifacts and the relationships between them that are essential for us. Such an abstraction of a part of reality is called a model. It is also possible that the segment of reality considered is already a model itself. This allows parts of an already existing model to be examined more closely. This would then be a model of a model. This process can be repeated as many times as desired.

Every person has his or her own subjective view of the world or a part of it. Different people may consider the same part of reality and arrive at different models because they set different priorities (cf., Fig. 2.1).

Since models do not cover all aspects of the assigned reality, there are indeed incidents in reality that are not covered by a model and are incomprehensible therein.

A. Fleischmann et al., *Contextual Process Digitalization*,
https://doi.org/10.1007/978-3-030-38300-8_2

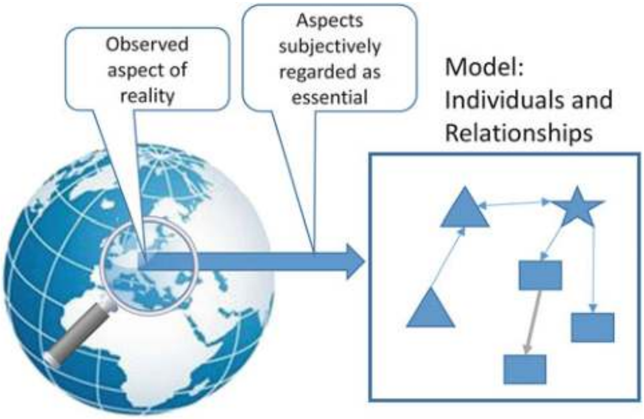


Figure 2.1: Modeling

These phenomena, which cannot be explained in a model itself, form its limits. The creator of a model may decide to adapt it accordingly when intending to include phenomena not yet covered or to omit these if they should no longer be taken into consideration. In any case, a model remains a simplification of reality, which is why there will always be phenomena that are not covered by a model. Each person needs to decide whether the model in use is sufficient for his purposes. If one wants to cover all aspects of reality with a single model, it reaches the complexity of reality. Then one doesn't understand the model to the same extent one didn't understand the reality before mapping it to a model. The intention of modeling, namely, to make reality more comprehensible and manageable, is then no longer achieved.

The creation of a model and its use are subjective activities, i.e., the creator chooses the characteristics of the representation of reality according to his ideas. However, it is common for groups of acting participants, subsequently referred to as subjects or actors, to agree on a model for observing reality. Many scientific schools of thought are based on such common models of the involved researchers.

Modeling is an essential activity in all sciences, be it philosophy, sociology, physics, chemistry, all engineering sciences, economics, etc. The respective models have different tasks: Either they depict the specific part of reality under consideration, as is traditionally done in the natural sciences, or they serve to try out certain necessary changes to said part (simulation model). This is particularly important to avoid endangering human life. Before going into series production, the properties of cars are checked in safety models in appropriate tests. This does not happen with people, but instead with models of people, so-called dummies. In doing so, two types of models are combined with each other: The car model, which implements the corresponding safety concepts, and the human model (dummy) to investigate the risks of injury.

The state of affairs captured in a model can be adapted as often as required until a desired result is achieved for the phenomena considered in the model. For instance, further safety concepts are added to the safety model of a car until the desired reduction of the risk of injury is achieved. Such models are then no longer images of reality, but rather represent a desired reality. The desired properties are then transferred back into reality, e.g., by incorporating the safety concepts tested in the model into the series production of cars.

The aim of modeling is always to find our way around in the world, or to try out safely how a corresponding change in reality would affect us. We will not likely succeed in understanding what holds the world together at its core in the foreseeable future. The corresponding model would then be the world that does not exist [1].

2.2 Properties of Models

Models serve both as the abstract representation of the observed reality in the sense of a cognitive function and as the design of the observed reality in the sense of a conclusion. As already mentioned, once a model is modified until it corresponds to a desired reality, it can be used as a blueprint for a corresponding transformation of reality. In natural sciences such as physics, models predominantly have a cognitive function, while in engineering sciences and business administration they are intended to support the shaping of reality [2].

In clarifying the model concept used thus far we follow the studies of Herbert Stachowiak. In his model theory he examined the characteristics of models and the properties derived from these more closely [3]. Accordingly, models are identified by at least three characteristics ([3] page 131):

1. Mapping

   A model is always a representation of something. It can be a depiction or representation of a natural or artificial original, whereby this original itself can again be a model. The originals can be created in a natural way, technically produced, or simply exist in some other way. Models can be described or represented in very different ways:

   - Mental models: Imagination in the human mind
   - Verbal models: Natural language description
   - Graphical models: Technical drawings or other pictures
   - Material models: Models of buildings
   - Formal models: Mathematical models, computer programs, etc.

A model and the original form a class of attributes. Attributes are characteristics and properties of individuals, relations between individuals, properties of properties, properties of relations, etc. Stachowiak leaves it up to the modeling subject to determine how to conceptualize individuals. He considers individuals as attributes on level 0. Sets of attributes can be combined to form classes, which then form attributes on level 1. These classes can be combined again to form a next attribute level, and so on.

2. Reduction

   In general, a model does not capture all attributes of the original, but only those that appear relevant to the creator or user of the model.

   Since not all attributes of the original are captured by a model, a pragmatic dimension has been introduced in the broader sense. In the "broader sense" here means that not yet specific pragmatic-operational aspects are considered, according to which the attribute classes that are to be included in a model are selected. This initial selection of attributes is intuitive and arbitrary. In the narrower sense, the reduction is pragmatic only when the intentions and operational objectives of the model creator or user influence the selection of model-relevant attributes. These adjustments to the intended practical use are made in the next step.

3. Pragmatism

   After the intuitive selection of the attributes, it is checked whether the intended purpose has been achieved. Models are not clearly assigned to their originals. They serve as a replacement function

   - for certain discerning or acting model-using subjects (for whom?).

     Models are not only models of something, they are also models for someone. This someone can be a human being or an artificial model user such as a computer program. For a modeler, models can serve as a possibility to find one's way around in the world, i.e., the modeler is also a user of the model. Modelers and model users can also be two different subjects.
   - within certain time intervals (when?)

     Models also perform a function over time, i.e., their use is related to a specific point in time or to a defined time interval. During this time, the observed reality or the ideas of the modeler or model user may have changed in such a way that additional attributes should become part of the model.
   - with restriction to certain mental or actual operations (why?).

     Models are created for a certain purpose, be it for better understanding a certain part of reality or for creating a blueprint for the transformation of reality.

When creating a model, modelers are always in a certain dilemma. On the one hand, the model should sufficiently reflect the desired aspects of reality, whereby it is not clearly defined what is sufficient. On the other hand, the model should not be too complex in order to remain manageable. This conflict of objectives leads to the fact that most models are developed iteratively until they reach the end of their life cycle due to increasing complexity - they are no longer manageable.

In the following sections, we present examples of models that consider aspects that are explicitly or implicitly incorporated into business process models. We have grouped the examples into models from the social sciences, business administration, business informatics, and computer science. The classification into these groups is not entirely free of overlaps, since especially business informatics as a cross-sectoral discipline considers issues from various perspectives.

2.3 Models of the Social Sciences

Business Process Management has to do with people and machines. It aims at organizing their interaction while taking into account additional requirements with regards to technical, economic and ecological feasibility. In particular, the interaction and coexistence of people has been a topic of philosophy for thousands of years. Philosophy, as the doctrine of the basic rules and structures of life, the world and knowledge, seeks to fathom, interpret and understand the world and human existence. The original meaning of philosophy was the teaching of good life.

In this sense philosophy is the attempt to create the comprehensive model of our world, but this has not yet been successful. Social philosophers reduce the view and try to create a model of society as some part of reality and thus better understand its meaning and essence. In particular, social philosophies illuminate the relationship between the individual and the community as well as the structures of living together. They are therefore also regarded as variants of philosophy that touch on sociology. They should help sociologists to analyze social processes and support organizational developers in their work and help people to find their way around in the world. There are numerous organizational theories that focus on different aspects in their models (see [4–7]). The question of which organizational theory fits best cannot be answered. Organizational theories are models and thus, according to Stachowiak, the justified but subjective view of the modeler. Organizational theories emphasize the analysis of organizations in highly different ways and pursue different objectives. There are empirical studies on organizational theories that provide results in favor of, or against, a theory. However, the analysis methods used are controversial (see [8] page 68).

Organizational theories are based on certain perspectives on people, and the design of Business Process Management is strongly influenced by the prevailing image of people in an organization. Hence, in the following sections we discuss Taylorism, Habermas's Theory of Communicative Action and Luhmann's Social Systems as distinct organizational theories to exemplify different perspectives on human and organizational behavior.

Taylorism and Fordism

Taylorism introduced the ‘experiment’ into management theory and practice and is one of the classics of organizational theory. With the so-called Scientific Management, organizations received an instrument to design themselves efficiently. One important characteristic is the separation between planning mental work and performing manual work. According to Taylor’s view of man, workers are dumb and lazy and must therefore be subject to strict rules. Even today, this view often implicitly influences leadership behavior.

For Frederik Winslow Taylor, the most important goals of running a business were the perfection of the means of production and work processes, tighter organization and temporal structuring of workflows in the company, as well as a reorganization of the remuneration system. A core element of Taylorism is the design of work processes on the basis of time and movement studies. A breakdown of the production process into the smallest work steps, an exoneration of workers from mental activities, as well as a change of the wage system should lead to an optimal use of existing performance potentials. The ultimate goal is to increase the productivity of human labor. This is done by dividing the work into its smallest units, which require only little or no cognitive effort to accomplish, and which can be repeated quickly and repetitively due to their small size or the content of the work.

Taylorism is based on the following core principles:

- Work scheduling
  The planning of the work is done by other persons than those who carry it out (separation of manual and mental work). In this way Taylor wanted to avoid the shirking that he accused the workers of. Through time and movement studies carried out by the mental workers, the least amount of movement and time required for a work step was to be determined.
- Incentivized wages
  These time and movement studies also revealed what the workers had to achieve in a certain period of time. A bonus ensured that the workers who were classified as dumb and lazy actually tried to achieve the specified performance.
- Selection of the most suitable workers
  One aim was to build up a first-class workforce by means of an appropriate selection mechanism. Tests were developed and used to identify particularly agile and nimble-fingered workers.
- Reconciliation between workers and management
  Taylor believed that the system he had developed could increase productivity to such an extent that the dispute over the distribution of profits would become a minor issue. This achievement should resolve the conflict between employers and employees.

Taylorism essentially considers the structuring of work steps but does not focus on their sequence. This aspect was addressed by Henry Ford, who coordinated the individual activities by introducing the assembly line. This step created the basis for the mass production that characterized the 20th century. The assembly line principle was also transferred to administration and strongly influenced Business Process Management. Flowcharts are the assembly line specifications for the execution of administrative tasks or the "production" of services.

Communicative Action According to Habermas

In contrast to Taylorism, Habermas's Theory of Communicative Action is based on insightful people who, through communication among themselves, come to a common rational action [9, 10]. Using his social model Habermas explains the processes in a society, such as the search for truth, for justice, etc. Consequently, it is a model that concerns everyone, because the issues of truth and justice affect all members of societies. The central aspect in Habermas's model of society is the so-called Communicative Action.

"Finally, the concept of Communicative Action refers to the interaction of at least two subjects capable of speaking and acting that enter into an interpersonal relationship (whether by verbal or nonverbal means). The actors seek an understanding of the action situation in order to coordinate their action plans, and thus their actions, amicably". (see [11], Volume 1, page 128).

According to Habermas, communication enables an individual human being, who is not gifted with rationality on his own initiative, to overcome this deficiency. Communication between people becomes intersubjective action and a possible source of rationality. Communicative Action means acting on the basis of mutual understanding between people.

Habermas wants to offer sociologists and politicians a model that they can use to analyze and shape society. The individual can use it to find his way around in today's societies, despite their complexity.

Social Systems According to Luhmann

Similar to Habermas's, Luhmann's Social Model is based on communication. The differences lie in the extent to which communication and action are combined. Luhmann only allows communication as a constituent aspect for organizations. Communication does not occur between people, but between at least two information-processing processors. Luhmann thus sees communication more abstractly. According to him, society does not consist of people or parts of people. Otherwise one would cut off something from society, when one cuts off something from a person. The body of a human being (i.e., as a biological system) with a conscious mind (a psychological system) is in many cases a prerequisite for the functioning of a social system, i.e., communication. However, a human being is not the social system itself. Luhmann makes no assertions about the nature of man in his organizational theory, thus leaving the perspective on humans open. People are only part of the organization insofar as they communicate with each other.

The communication between the information-processing processors consists of the so-called selections of information, message and understanding (see Fig. 2.2). The first two selections are for the sender and the third for the recipient. Communication as a piece with at least two actors in three acts is an indivisible unit, namely the smallest unit of a social system and the elementary operation of society

	Two information-processing processors, usually people or social systems	
	transmitter Luhmann's "Alter"	receiver Luhmann's "Ego"
Three selections:	1. Selection of information 2. Selection of message	3. Selection of acceptance/ understanding

Figure 2.2: Luhmann's understanding of communication

[12, 13]. This view can serve as a pattern for the definition of communication in business processes, completely independent of a specific perspective on humans.

Both Luhmann and Habermas put communication at the center of their organizational theory. The fact that these two important organizational theorists so strongly emphasize the communication aspect of the organization, and that their theories are widely accepted, can be seen as an indication for considering business processes as primarily communication oriented.

Organizations

Complex social systems can be divided into smaller social systems. This structure of a complex social system is called the organizational structure. The criteria according to which the division into smaller social systems takes place are subjective and depend on the respective intentions. According to Luhmann's definition of a social system, the individual social systems communicate within a more complex social system. Organizational structures are thus a model of a more complex social system.

In contrast to Luhmann's or Habermas's broad understanding of the term organization, a narrower understanding of organizations has also developed. In business administration, organization is the formal set of rules of a system based on the division of labor. In organizational sociology it refers to a special form of social entity that can be distinguished from other social entities such as families, groups, movements or networks. Essential characteristics of organizations are that people can join them or leave them. In addition, they have a purpose that they are geared to. Organizations have regulations on the division of labor, such as specialization according to performance, function, objects or space, or corresponding hybrid forms. This division of labor requires the coordination of individual activities. Hierarchy is the central instrument of coordination in organizational theory. The hierarchical coordination is supplemented by cadres, commissions, task forces, etc. For one-time problems or problems to be solved for the first time, the hierarchy is supplemented by a project organization.

2.4 Models of Business Administration

Business administration is the study of economic, organizational, technical and financial processes and structures in companies. Business Process Management is therefore also a part of business administration. Business processes serve to improve the economic efficiency of a company with all of the associated aspects such as customer satisfaction, employee motivation, the integration of partners, etc. For the structuring of all these aspects and for the analysis of their interaction, business administration has developed models which, when applied, have an effect on Business Process Management.

Business Model

A business model refers to the overall concept of a company. It represents the interrelationships as a model, how a company can generate added value for its

Figure 2.3: Schema of the Business Model Canvas

customers and thus achieve sustainable returns. In addition to the products and services offered, the focus is on the structure of the company, the definition of the target groups (customers) and how they are addressed, as well as the design of the business processes. Beyond this understanding, there are a number of other definitions of the term business model (cf., e.g., [14, 15]).

A business model therefore serves to understand the relationship between the company as a system of action and the creation of value. It reflects how a company works and what values it generates for specific target groups. Business models are created within the scope of a company foundation or a reorientation. They consist of several sub-models that describe which resources (materials, information, etc.) are (must be) available to a company as input variables, and how these resources are processed and transformed into marketable products or services, which are then transferred to the customer in order to generate corresponding revenues [16].

Business models can serve multiple stakeholders. The company management can thus better understand its own business and recognize existing strengths and weaknesses as well as opportunities for further development, transformation and improvement of its competitive position. For investors, the business model is often an important aspect of investment decisions.

A number of instruments have been developed to create business models. The best known is the Business Model Canvas by Alexander Osterwalder [17], which has found high acceptance in recent years. As the name suggests, the Business Model Canvas approach is based on a poster on which various aspects of the business model are visualized. The canvas provides a grid for nine business model aspects, which is filled with the concrete characteristics for a company at hand (see Fig. 2.3). The focus is on the value proposition (product or service).

When completing the form, a series of questions on each of the nine aspects must be answered. The following explanations briefly describe the aspects and provide a selection of associated questions.

1. Customer segments, target groups:

   All persons or organizations for whom the company in question wants to create values.

   Questions to be answered include:

   - Who benefits from the product or service?
   - Which customers are particularly important?

2. Value promise, customer benefit:

   Each customer segment has its own value proposition, the customer benefit. This is a combination of a product and service tailored to the needs of the respective segment.

   Questions to be answered include:

   - What benefit or value does the offer have for the customers?
   - Which customer problems are solved with the offered products and/or services?

3. Channels, sales channels:

   This factor represents the specific channels through which customers are addressed and promised values are communicated to them. Sales channels determine how interaction with customers takes place. Communication, distribution and points of sale form the interfaces between a company and its customers. The perception of the customer at these points of contact is central and determines the impression a customer has of a company.

   Questions to be answered include:

   - How do customers find out about the products and services offered?
   - How do the products/services reach the customer?

4. Customer relations:

   This section describes how dealings with customers are fostered.

   Every company should think about what types of customer relationships it wants to establish with different target groups. The design of customer relationships depends not only on the respective target group, but also on the associated objectives of the company (new customer acquisition, existing customer care, etc.).

   Questions to be answered are among others:

   - What kind of relationship do the individual customer groups expect?
   - How is the relationship with the customers organized?
   - How much does it cost to maintain customer contact and what is the value of this particular customer?

5. Revenue sources, revenue models:

   The company creates added value with its products and services. The central question is how much the customer is willing to pay for this. The company needs to decide on pricing models and pricing strategy (one-time payment, subscription, etc.).

   Questions to be answered include:

   - For what and how much are customers really willing to pay for the offer?
   - How much does each of the individual revenue sources contribute to total revenue?

- How would customers like to pay?

6. Key resources:

   Every company requires certain resources to prepare offers. These can be owned by the company itself, or also leased or provided by strategic partners.

   Questions to be answered include:

   - Which physical resources (facilities, production machines) are required to create and offer a product or service?
   - Which intellectual resources (knowledge, patents, partnerships, customer base) are needed?
   - Which personnel resources (teams) are required?
   - What financial resources (available capital, collateral) are required?
   - How can the necessary resources be procured and maintained?

7. Key activities:

   Key activities are the activities necessary for the creation and utilization of services, such as production, sales, and so on.

   Questions to be answered include:

   - Which key activities have to be carried out in order to offer a product or a service and thus realize the customer benefit?
   - Which activities are needed for which sales channels?
   - Which activities are required for which customer relationships?

8. Key partners:

   Key partners are business partners who provide important resources for the realization of the business model.

   Companies often enter into strategic alliances with these partners. Examples are suppliers, service providers, etc.

   Questions to be answered include:

   - Who are key partners and what do they do for the company?
   - Which key resources are provided by which partners?

9. Cost structure:

   The cost structure provides information on the most important cost factors of a business model.

   Questions to be answered include:

   - What are the largest and most important cost factors in the business model?
   - Which key resources/key activities are the most expensive?

Key Performance Indicators and associated target values for business processes can be derived from the individual parts of a business model. Conversely, the Key Performance Indicators and target values can influence the design of the processes. If the focus is on low prices, processes will look different compared to a business model focused on high quality.

Balanced Scorecard

The Balanced Scorecard (BSC) was introduced in the early 1990s by Kaplan and Norton [18]. It is a link between the business model, the development of a strategy, and its implementation. In the business world, strategy is classically understood as the (usually long-term) planned behavior of companies in order to achieve their goals.

A BSC starts with the vision and strategy of a company and defines the Critical Success Factors (CSF) on this basis with the help of Key Performance Indicators and associated target values. The vision of a company describes the long-term ambitious goal that an organization or company strives for. Typical visions are formulations such as "we want to become the market leader in our market segment," or "we want to become the most profitable company in our market segment".

The Key Performance Indicators promote goal setting and performance in critical areas of the strategy to achieve the vision. The BSC is therefore a management system that is derived from the vision as part of the business model and the strategy to implement said model. It reflects the key aspects of the company. The BSC concept supports strategic planning and implementation by bundling the measures taken by all entities of a company on the basis of a common understanding of its goals and by facilitating access to the evaluation and updating of the strategy.

Since traditional management based purely on financial indicators no longer meets the requirements of companies for effective planning tools in the information age, Kaplan and Norton have introduced four perspectives for the BSC which allow the activities of a company to be assessed comprehensively. For each perspective, objectives, performance indicators, targets, and measures to be taken are defined (cf., Fig. 2.4).

Total Quality Management and EFQM

The term Total Quality Management (TQM) denotes the optimization of the quality of a company's products and services in all functional areas and at all levels through the participation of all employees. Optimization of quality means neither reaching the highest quality level with the given effort, nor increasing the quality without consideration of costs. Rather, it is a matter of focusing on the interests of the customer and determining quality in terms of the fulfillment of customer requirements.

The management of a company decides which requirements the company places on itself and which positioning toward the customer promises the most sustainable business success. This positioning is not static. Knowledge about customer needs and about the procedures to meet these needs require a continuous adaptation of the company.

In order to establish TQM, the European Foundation for Quality Management (EFQM) offers organizations assistance in setting up and continuously developing a comprehensive management system. Figure 2.5 shows the structure of the EFQM approach. On the one hand, this structure serves as a tool to build up a TQM and, on the other hand, to identify improvement potentials through a comprehensive evaluation system as well as to increase business success.

Enablers in the EFQM model are the methods and concepts used to achieve the results shown in the right half of the figure. The percentages in the presentation indicate the extent to which the individual aspects are included in the overall evaluation of the company.

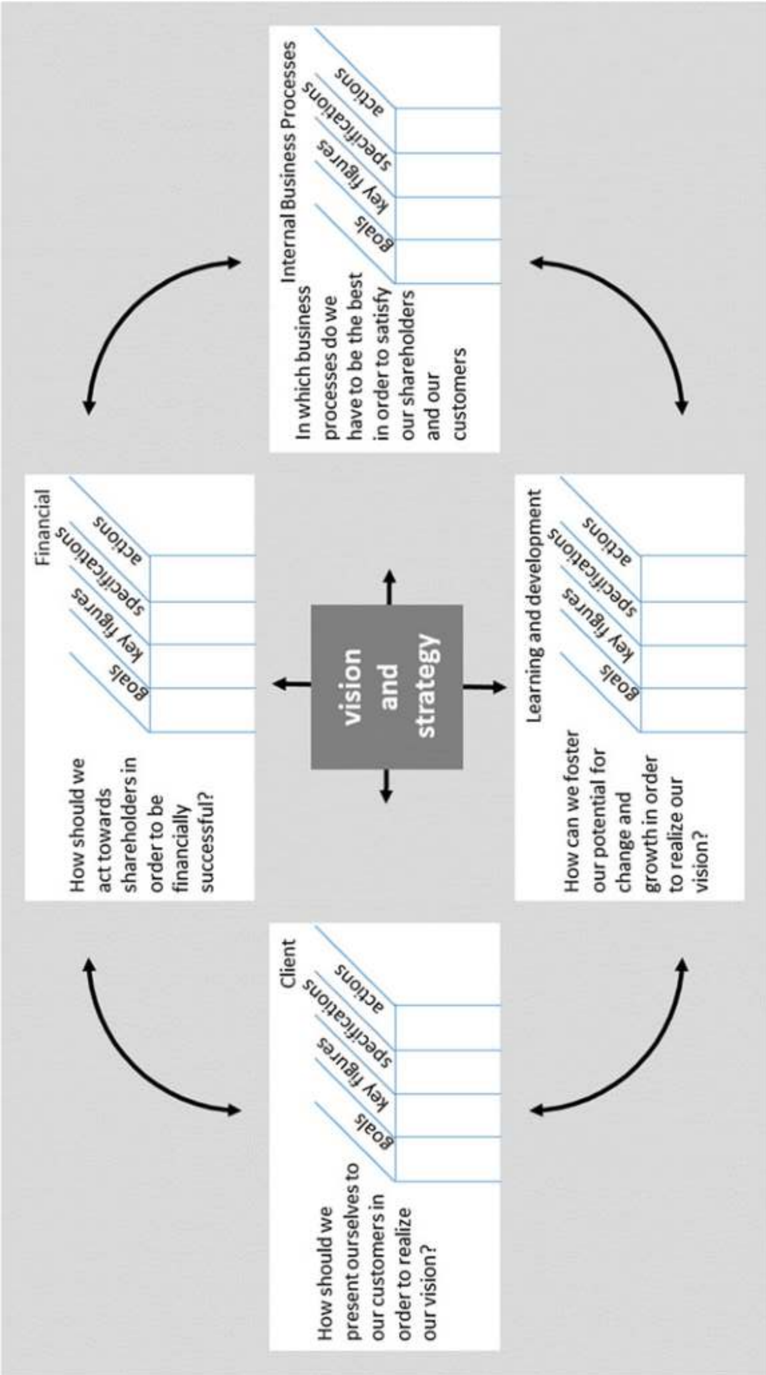


Figure 2.4: Perspectives of the Balanced Scorecard (adapted from Springer Gabler Verlag (publisher), Gabler Wirtschaftslexikon, keyword: Balanced Scorecard, online at: http://wirtschaftslexikon.gabler.de/Archiv/1856/balanced-scorecard-v8.html, last accessed 22.02.2018)

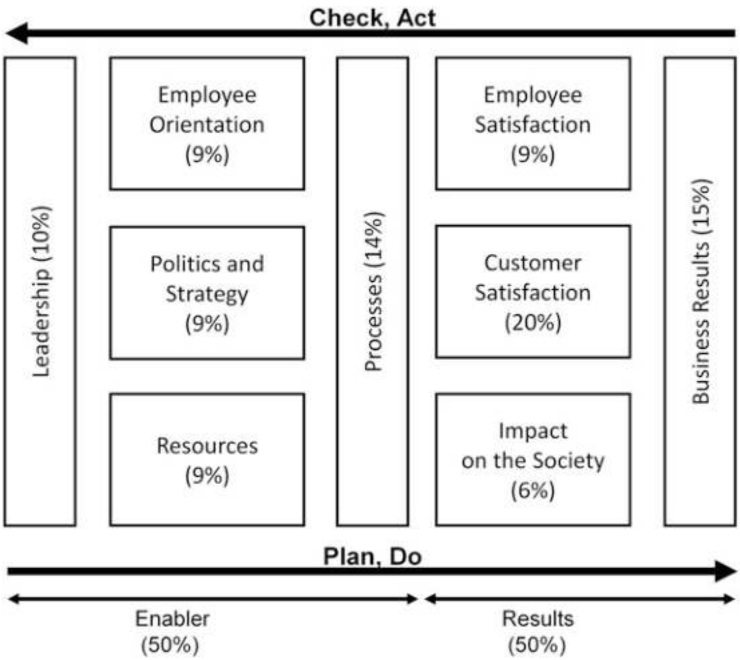


Figure 2.5: EFQM structure

The EFQM assumes that the enabling methods and concepts have the largest influence on the results (right side of Figure 2.5). The model thus provides good starting points for identifying Key Performance Indicators and their target values.

In addition to the possibility of setting up a management system, EFQM also offers a very sophisticated concept for evaluating its development status. A comprehensive catalogue of questions can be used to carry out an all-round evaluation of an organization. The evaluation can be carried out by employees of the organization itself or by external consultants. The best organizations in Europe score around 750 out of a maximum of 1,000 points in such evaluations.

EN ISO 9001

Compared to TQM, the EN ISO 9001 standard represents a weakened form of quality management. It describes minimum requirements for a quality management system. Figure 2.6 illustrates the basics of the standard.

Management's responsibility means that it defines which customer requirements are met and which quality policy is pursued. The implementation of the quality policy is planned, and the corresponding responsibilities and authorities are defined in the organization. Management is also responsible for evaluating the QM system at planned intervals and, in particular, taking customer feedback into account while doing so. Corporate management must also provide the necessary resources such as personnel, infrastructure, and an adequate work environment.

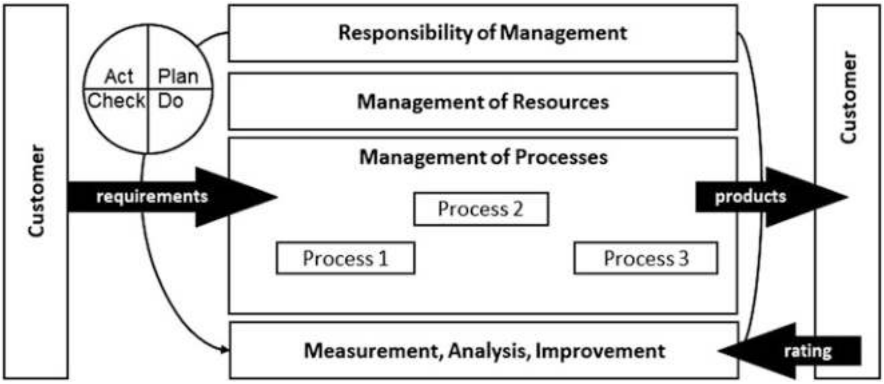


Figure 2.6: EN ISO 9001

The core of an EN ISO 9001-compliant QM system are the processes for realizing the products and the associated customer-related services. Tasks include the planning and definition of suitable processes for the development and manufacture of products, the procurement of inputs, etc. The tools used to monitor product manufacturing and quality must be regularly checked for their suitability. The execution of the processes must be continuously monitored through measurements and analyses of Key Performance Indicators, in order to be able to initiate appropriate improvement measures in the event of deviations.

EN ISO 9001 thus provides a framework for Business Process Management. The explanations show that, strictly speaking, there is no difference between Business Process Management and quality management. Without Business Process Management there is no quality management and vice versa.

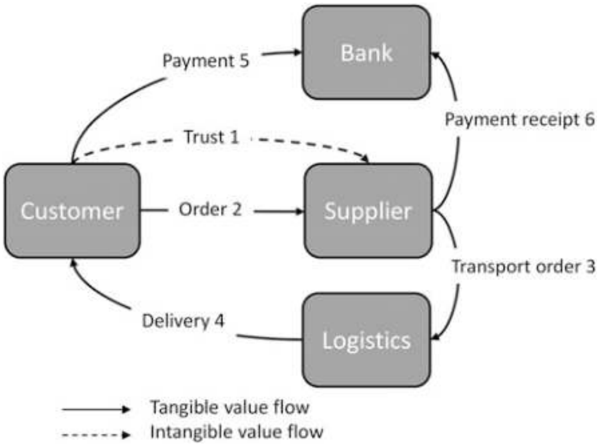
The comparatively lower requirements of EN ISO 9001 are expressed in the fact that a company with a (merely) EN ISO 9001 compliant quality management system can only achieve about 300 points in an EFQM assessment.

Value Networks

The Value Networks concept was introduced by Verna Allee [19]. A Value Network is understood as roles and persons who exchange so-called tangibles and intangibles with each other. Tangible value flows are material value flows between roles and persons, and correspond to the exchange of goods, services, revenues etc. Tangible value flows represent transactions based on contracts. Intangible value flows are an additional benefit through the flow of knowledge; they are not contractually fixed or subject to a charge. Intangible value flows are, e.g., strategic information, planning knowledge, as well as existing emotional components such as mutual trust, common interests, need for knowledge, security, etc.

Value Networks should enable participants and organizational developers to actively shape social and professional relationships of interaction in organizational systems by visualizing and creatively handling mutual tangible and intangible performance flows (transactions). Figure 2.7 shows a simple Value Network. The

Figure 2.7: Schema of a Value Network



customer sends a value purchase order as a tangible value flow to the supplier. This tangible value flow is accompanied by the intangible value flow of trust. The customer and the logistics company are connected with the tangible value flow of delivery, etc. The numbers on the individual transactions express the sequence in which they are executed.

Organizations provide services as a result of their activities that ultimately contribute to the added value of a company or institution. In order to record these services and make them visible, an exchange-oriented view of organizations is recommended. This usually results in a network-like structure in which the roles within an organization and their interaction and communication channels are in the foreground. This perspective enables the transition to a communication-oriented Business Process Management.

2.5 Models of Business Informatics

Models in business informatics combine aspects from the economic and social fields with computer science to derive requirements for information systems. The models are mainly used to describe socio-technical human-machine systems. The social component covers the aspects around employees and partners. The technical dimension concerns the circumstances of Information Technology. It is important that corresponding models consider the interaction between the two domains, especially the human-machine interaction. In contrast to purely technical systems, which are regarded as deterministic, socio-technical systems can also be non-deterministic, i.e., complex, due to the involvement of social components. The state of development and research on the subject of modeling in business informatics can be found in [20]. We limit ourselves here to the handling of frameworks for enterprise architectures and IT service management.

In order to simplify the creation of very similar process models, reference models have been defined over the years by consulting firms or standardization bodies. A well-known example of such a reference model is ITIL (IT Infrastructure Library, [21]). Due to its wide distribution, we describe it in more detail as an example of a reference model. A reference model that includes ITIL while being more focused on governance and compliance, is COBIT [22]. Due to space limitations we do not provide a representation here but refer to the extensive literature and the official website [23].

Enterprise Architectures

The understanding of architecture in the context of companies coincides with the original meaning of the term architecture. In many areas of expertise it describes the basic organization of a system with its components and their relationships to one another and to their environment.

As already mentioned in Chapter 1, an enterprise architecture specifically describes and links the business and technical elements of an enterprise. The latter include in particular the IT landscape. Both the overall architecture and its parts are described through models. The range of model types used for this purpose extends from a business model through organigrams, data models and process models at the business level, to database models, algorithms and programs in the technical layer.

As for business models, there are also numerous frameworks for the modeling of enterprise architectures that provide orientation for those responsible and are intended to facilitate work processes. Dirk Matthes [24] has identified more than 40 frameworks with varying foci, levels of detail and degrees of familiarity. Only four of the most relevant will be handled in the following sections.

The Zachman Framework and The Open Group Architecture Framework (TOGAF) with its extension Architecture-Animate (ArchiMate) were named as essential frameworks in surveys (see also [24] page 5). The Architecture of Integrated Information Systems (ARIS) is widely used in practice in German-speaking countries and is of significant importance in the context of process management.

Zachman Framework

The framework presented by John A. Zachman in 1992 [25] in its extended form represents a structure grid similar to the Business Model Canvas, which the user has to complete with the facts for the enterprise at hand. It consists of a matrix with different perspectives in the rows and abstractions to each perspective in the columns. Figure 2.8 shows a condensed representation, a detailed picture can be found on the Zachman International website (www.zachman.com).

The perspectives in the rows have the following meaning:

- Planner: Company objectives, external requirements and influences, business model
- Owner: Requirements for data, processes, structures, etc. for company management
- Designer: System design and system structure to implement the requirements

	What	How	Where	Who	When	Why
Objective/area -> Role:Planer	List of important factors in the business	List of core processes, process map	List of branch ofices	List of mportant organizations and stakeholders	List of business events	List of business goals, strategies and business models
Business model-> Role: Owner	Conceptional data model/object model	Business process model	Business logistics systems, distri-bution model	Workflow model, organizational structure	Schedule	Business plan
System model (logical) ->Role: Designer	Logical data model	Process implementation, system archi-tecture model	Distributed systems architecture	Human interface architecture role model	Process structure	Busines rules model
Technology model (Physically) -> Role: Builder	Physical data model, class model	Technology design model, implementation model	Technology architecture	Representation architecture, system usage	Control structure, system dynamics	System requirements
Detailed presentation (from context) -> Role: Programmer	Data definitions, database structure	Program, process configuration	Network architecture	Security architecture, directory services	Schedule	System monitoring
Operations ->Role: User	Concretly available equipment	Process instances	Business locations	Execution	Current schedules	Current moods
	Inventory, material and data	Processes and functions	Spatial distribution, geometry	People and machines	Scheduling	Goals, motivation

Figure 2.8: Zachman framework

- Builder: Implementation of the system design
- Programmer: Provision of the technical infrastructure
- User: responsible person for the operation to ensure the functionality

The columns contain the questions that the company needs to answer:

- What (inventory): What objects, equipment, data, information, etc. are required?
- How (functions and processes): How does the company work, for example, what do the business processes look like?
- Where (locations, network); Where are the company's locations?
- Who (people): Who are the people who keep the company running? Which business units are there and what is the organizational structure like?
- When (Time): When are business processes instantiated and executed? What are the time schedules for the business?
- Why (motivation): Why do we run the business the way we run it? What are the drivers of the business? Aspects of the business model are incorporated here.

Zachman envisages that a suitable model will be developed for each cell in the table. From this perspective, his framework is a model for a set of models that allow a closer look at different aspects of a company.

The users can deviate from the original structure in the rows and columns by changing the emphasis. This flexibility is a strength of the model frame. However, it does not contain any procedure or methodology for defining a concrete enterprise architecture. Processes for their development or transformation have to be exploited elsewhere by the users or have to be designed entirely by themselves.

The Open Group Architecture Framework (TOGAF)

TOGAF is the Open Group's framework for the development of enterprise architectures, including business processes [26]. While the Zachman framework emphasizes the object perspective and offers little support for the architecture development process, TOGAF focuses on the procedure for model creation. It provides methods and tools that help with the introduction, creation, use and further development of enterprise architectures.

TOGAF distinguishes four sub-architectures:

- Business Architecture: Business aspects of enterprise architecture
- Data Architecture: Logical and physical structures of data and resources for their management.
- Application Architecture: The application systems used and their relationships with each other as well as their relevance for the company's business.
- Technology Architecture: Software and hardware requirements for data management and application system execution. This includes, e.g., runtime environments, networks, middleware, and other operational infrastructures.

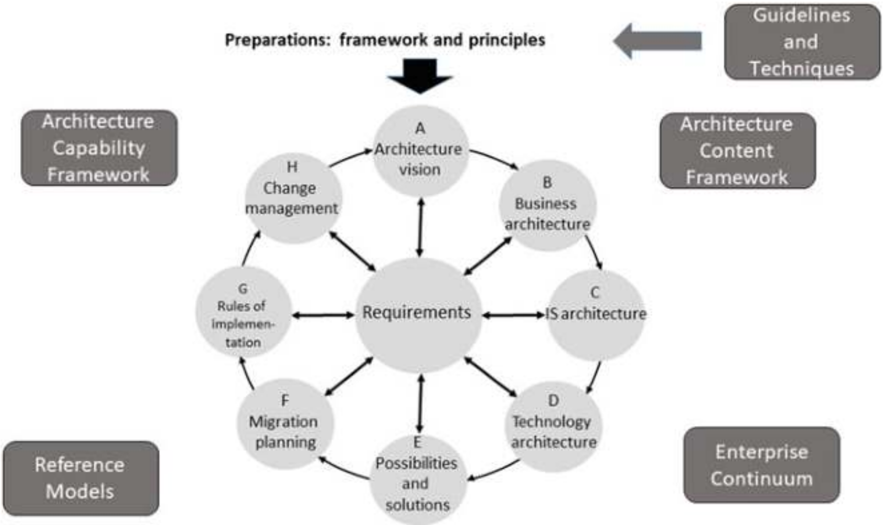


Figure 2.9: Architecture Development Method from TOGAF

The TOGAF framework consists of the following components:

- Architecture Development Method (ADM)
  Method and procedure for the development of an enterprise architecture.
- ADM Guidelines and Techniques
  Set of tools and guidelines that support the use of ADM (e.g., tools for iterative use of ADM).
- Architecture Content Framework
  Structural model to define, structure and display the results generated with ADM in a uniform and consistent way.
- Enterprise Continuum
  Model for structuring a possible repository that can contain the respective architectures and the possible solutions such as models, patterns, architecture descriptions, etc.
- Reference Models
  Basic models that can be used as a basis for specific models for a company. These are the Technical Reference Model (TRM) and the Integrated Information Infrastructure Model.
- Architecture Capability Framework

Various reference materials for the development of specific architectural models. The Architecture Development Method (ADM) forms the core of TOGAF as an iterative process model (cf., Fig. 2.9). This creates all of the architecture artifacts. ADM can be applied at multiple levels, allowing architects to define different levels of detail of the enterprise architecture. With the help of the other components, the results are then described, structured and stored.

The phases of the ADM are:

- Preliminary phase: Here, the organizational environment and the frameworks, methods, support tools and important principles used are defined.
- Phase A - Architecture Vision: Here, the goals and the parties involved in updating the enterprise architecture are defined and integrated.
- Phase B - Business Architecture: The current and desired state of the business architecture is described here. The decisive differences are worked out. The desired views are defined and the associated appropriate tools are selected.
- Phase C - IS architecture (Information System Architecture): The current and desired state of the application and information/data architecture is described here. The decisive differences are worked out. The concrete applications and data models are used for this purpose.
- Phase D - Technology Architecture: The current and desired state of the technology architecture is described here. The decisive differences are worked out. In addition, the concrete hardware systems are described.
- Phase E - Opportunities and Solutions: Here, the projects are defined which carry out the transformation from the current situation to the target state.
- Phase F - Migration Planning: The transfer from a current state to a target state is planned here.
- Phase G - Implementation Governance: The implementation into the target state is carried out and monitored here.
- Phase H - Architecture Change Management: Requirements and external influences are collected here, which then serve as the basis for the next ADM run.
- Requirements management: The requirements management drives the ADM process continuously and is therefore at the center.

Architecture-Animates [27] (ArchiMate)
Architecture-Animate (ArchiMate) is the name of an open and independent modeling language for enterprise architectures published by the Open Group. It provides tools that enable enterprise architects to describe, analyze, and visualize the relationships between business units and their development.

The ArchiMate language enables the description of the structure and flow of business processes, organizational structures, information flows, IT systems, and technical infrastructure. The descriptions help the participants to design changes in architectural elements and their relationships, to evaluate the consequences and to communicate them. Figure 2.10 shows the ArchiMate framework.

The first three columns correspond to the basic concepts of ArchiMate:

- Passive structure elements

  Passive structure elements are the objects on which the actions from the behavior (behavior elements) are executed. In general, these are information objects, but physical objects can also be modeled as passive structural elements.

	Passive structure	Behavior	Active structure	Motivation
Strategy	Resources		Resources	Stakeholders, drivers, goals, principles and requirements
Business	Business objects	Business services, functions and processes	Business actors and roles	
Application	Data objects	Application services, functions and processes	Application components and interfaces	
Technology	Artifacts	Technology services, functions and processes	Devices, system software, communication networks	
Physical	Material			
Implementation and migration	Deliverables	Work packages	Platforms	

Figure 2.10: Elements of the ArchiMate framework

- Active structure elements
  Active structure elements are elements that can perform actions. Examples are people, applications, computer nodes, etc. The actions can be triggered via interfaces, which also provide the results.
- Behavior elements
  Behavior elements represent the dynamic aspects of a company. A service is the externally visible behavior of the system that provides the service. The services are used via the corresponding interfaces. Interface events trigger the active structure elements, which then execute the corresponding service function.

These three model fragments correspond to the basic elements of natural languages: subject, predicate or verb and object. They are considered on a total of six layers:

- Strategy layer
  Motivation describes what a company wants to achieve. The strategy concepts describe at a high level of abstraction how a company wants to achieve its goals.
- Business layer
  The elements of the business layer can be used to describe products and services that a company makes available externally. The business layer shows how the company realizes these products and services and is intended to help with the analysis of the corporate structure.
- Application layer
  In the application layer, the support of the business layer is represented by applications and data.
- Technology layer
  In the technology layer, the infrastructure needed to implement applications is described. These are essentially the required hardware and software components.
- Physical layer
  This layer focuses on the interaction of IT and physical components such as machines, sensors and actors.

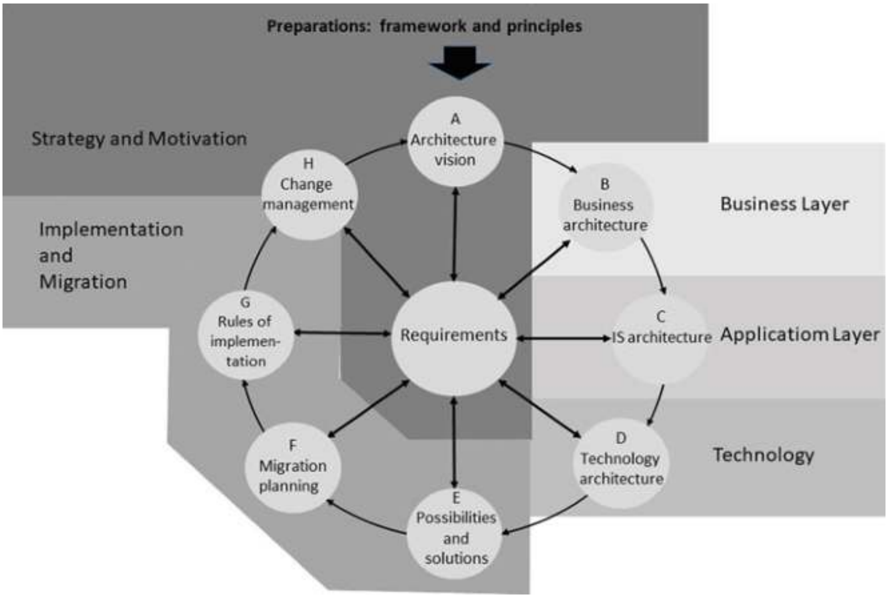


Figure 2.11: Relationship between ArchiMate layers and TOGAF ADM

- Implementation and migration layer
  The implementation and migration concept describes how a defined architecture is to be implemented. In particular, it describes the work packages for implementation.

The cells of the table therefore contain the core elements for the active and passive structures as well as for the behavior in the respective layer.

The motivation column describes the reasons for designing or changing an enterprise model. These influence the modeling and give it the appropriate direction.

ArchiMate is seen as a supplement and concretization of TOGAF. TOGAF describes the process for the definition and description of an enterprise architecture (enterprise model), but it does not contain any description languages for the respective sub-models. ArchiMate aims to fill that gap. In addition to the framework shown for the architecture to be developed, it also offers languages for expressing the aspects in the individual layers. Figure 2.11 shows how the individual steps in TOGAF's architecture development method are related to the ArchiMate layers.

Architecture of integrated information systems (ARIS)

The Architecture of Integrated Information Systems (ARIS) is a framework for the definition of enterprise models. It includes the data, function, organization, control, and performance views. For each view, ARIS provides a number of model types for documentation.

- Data view:
  The data view comprises the business-relevant business or information objects and their relationships to each other, that is, all data that is related to the activities of a company. Information objects include states such as article or customer status as well as events such as "Sales order has arrived" or "Production order has been triggered". The relevant model type is the Entity-Relationship Model (ERM).
- Function view:
  The function view describes the business-relevant activities (functions, activities) and their hierarchical relationships. Subordinate functions are sub-functions of the higher-level function. The functions perform operations on the objects described in the data view. In practice, functions are modeled with function trees.
- Organizational view:
  In the organizational view, the organizational structure, that is, the personnel resources of a company and their hierarchical relationships are modeled in an organizational plan. The organigram is the usual model type to represent the organizational view.
- Control view:
  The control view establishes the chronological and factual connection between the individual operational activities. It merges the data, functional and organizational views and thus plays a central integrating role. The control layer is therefore also called the process view. The main model type is the Event-driven Process Chain (EPC).
- Performance view:
  In the performance view, the entries and results of the business process at hand are usually described using product trees.

The views, typical model types and their context are shown in Figure 2.12. The picture also makes clear that ARIS, with its integrated control view, places business processes, especially the sequence of activities, at the center of the consideration.

Orthogonal to the structuring carried out by the views, ARIS differentiates the abstraction levels functional concept, data processing concept and implementation on the basis of software engineering. This shows the close relationship between the models developed with ARIS and Information Technology. To solve an operational problem, a functional model is created and transferred to a corresponding data processing concept (data processing model). This ultimately serves as the basis for the concrete technical implementation.

- Functional concept:
  The functional concept describes the facts of the operational problem. At this level, data models, functional models, organizational charts, value chains or Event-driven Process Chains (EPCs) and product models are used.
- Data processing concept:
  The data processing concept specifies how the functional concept is to be implemented in IT terms. At this level, database models (data view), structure

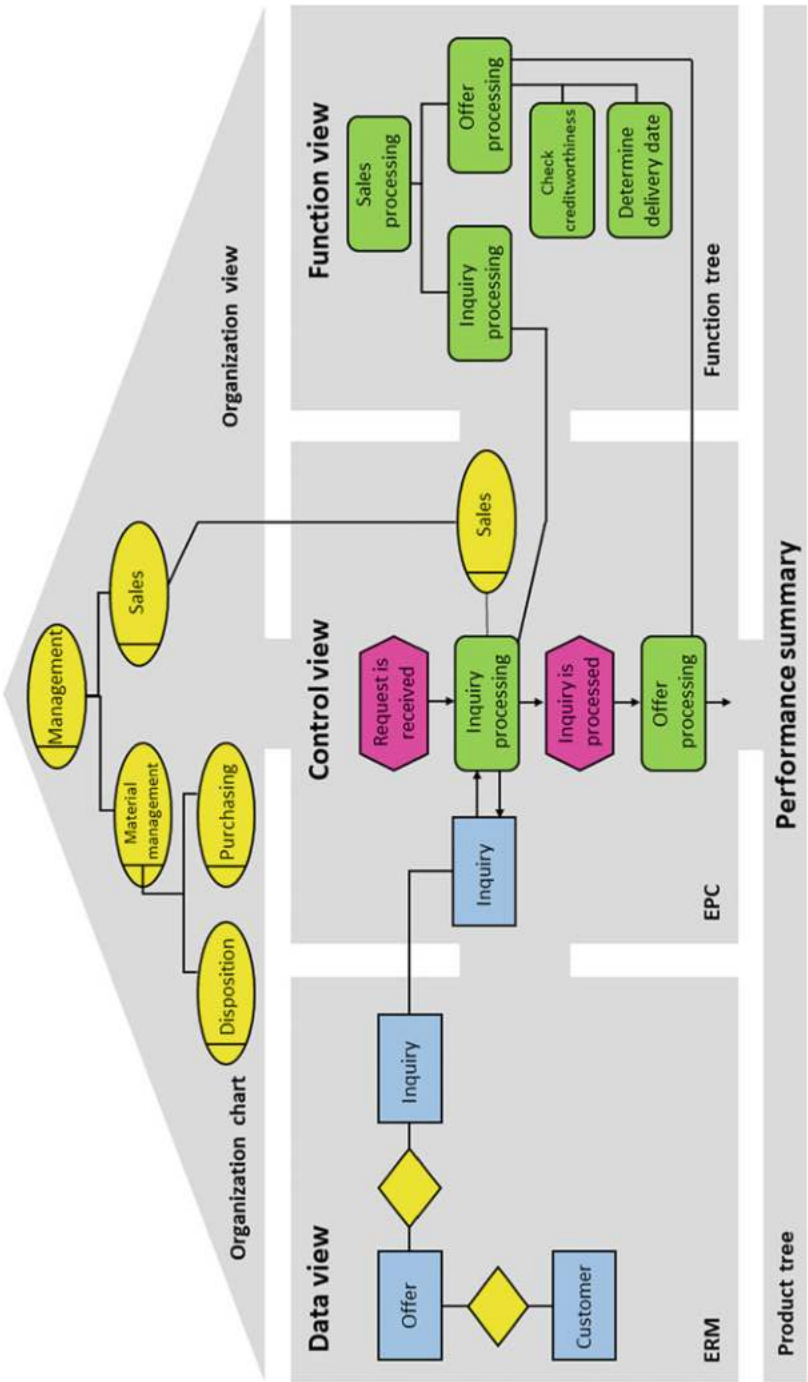


Figure 2.12: Views and relevant model types in ARIS

charts (functional view), network topologies (organizational view) and trigger mechanisms (control view) are considered. The purpose of the data processing concept is to adapt the functional concept to the requirements of Information Technology.

- Implementation:
  At this level, the data processing concept is converted into an executable software system. At this abstraction level, data description languages (data view), programs (function view), network protocols (organizational view), and program control (control view) are considered.

Framework for IT Service Management: ITIL

The IT Infrastructure Library (ITIL) is a collection of predefined processes, functions and roles that typically occur in every IT infrastructure of medium-sized and large companies [21]. The practical assignment of activities is based on roles and functions. These are best practice proposals that have to be adapted to the needs of the company. The collection has since been supplemented by ISO 20000:2005, an ITIL-based certification model for organizations.

The IT Infrastructure Library comprises five core volumes with a current total of 37 core processes. Figure 2.13 shows the structure of ITIL. The five core volumes are based on the service life cycle. Based on the service strategy containing the processes strategy development, financial management, service portfolio management and demand management, the service is finally provided via the process groups of service development and service commissioning with the processes of the service provision group. The procedures are subject to continuous improvement.

Further books such as "Software Asset Management", "Small-Scale Implementation" or "Building an ITIL based Service Management Department" supplement the core publications. ITIL thus offers comprehensive support in the development of a process system for the IT department of an organization.

2.6 Models in Computer Science

Models in computer science relate to data structure models and processes in computer systems as well as to various essential accompanying aspects such as security, robustness, etc.

On the one hand, these models serve to illustrate a considered part of reality in order to solve a task with the help of information processing. They refer to a defined problem area or certain application areas of computer systems. This includes models that focus on data and the operations running on it, as well as models that look more at the overall structure of complex programs, i.e., their architecture. These two model categories are also called models for programming on a small scale or models for programming on a large scale respectively.

In addition to these central model categories of computer science, there are other models that consider flanking aspects such as access models, security models, etc. These are not the subject of further discussion here.

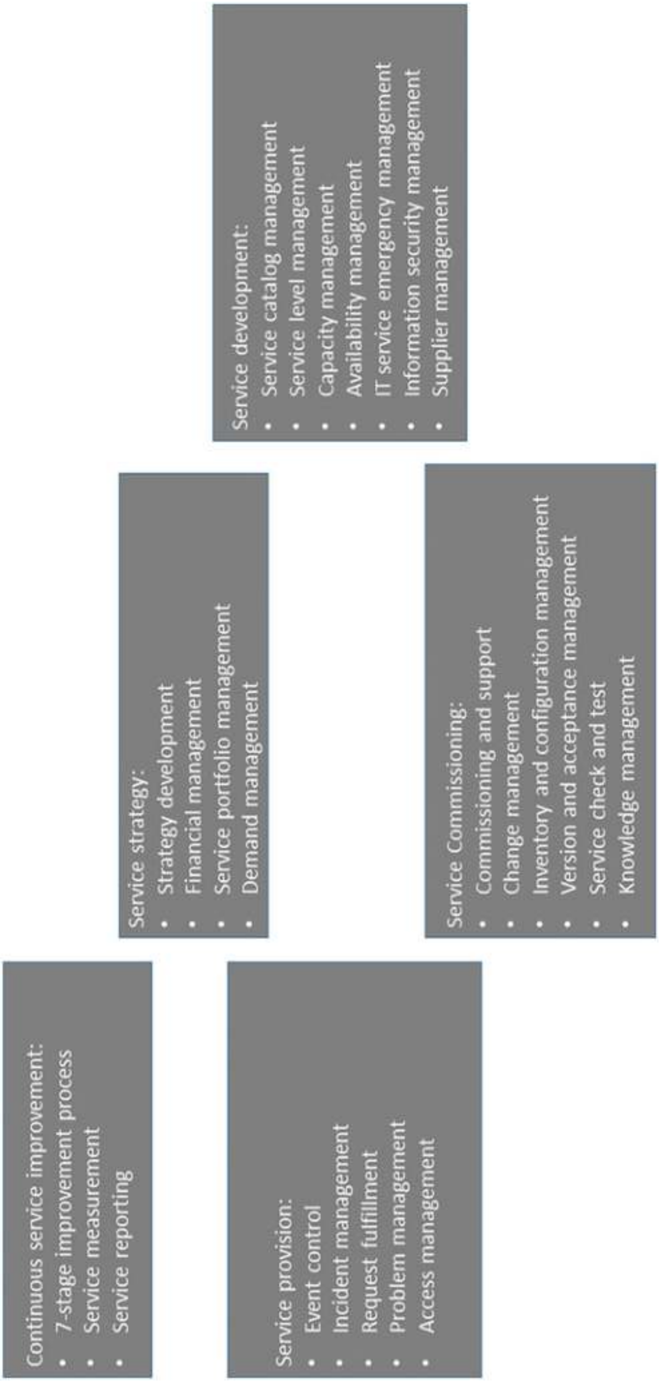


Figure 2.13: ITIL process groups

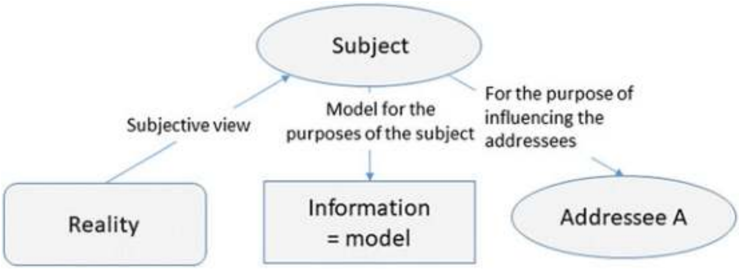


Figure 2.14: Information is “model - from what - for what - for whom" [11]

Instead, in the following sections we explain several model concepts considered essential for Business Process Management.

Information

Information is the more central aspect of data processing, which is why it is also called Information Technology or IT. Information and information systems are models that represent an object of the real world according to the ideas of one or more subjects. The subject's ideas are oriented toward the intended purpose. For example, items in a warehouse are described by their properties such as part number, dimensions, weight, etc. Different users use different parts of the model: The purchasing department looks at information such as purchase price and order limit, while the warehouse worker is more interested in dimensions and storage location.

The modeling of the object reality can therefore be understood as an interpretation by the user agents (subjects) such as purchasing or warehouse clerk. Information is then the result and the reason for an interpretation, but it can also itself be an object and thus an object of interpretation and modeling. This relationship between subject, information (model) and original is shown in Figure 2.14.

Information receives its value through the interpretation of the overall event by the observing subjects. This observation is partly conscious, but mostly unconscious. The amount of information is reduced and filtered according to the respective need for knowledge, or linked with other information.

Data is different from information. A data element is initially only a sequence of characters whose meaning is not unique. The characters can be numbers, letters or symbols. In the marketing department of an online shop, for example, the sequence of numbers 0815 may be found. Although this sequence of characters represents a date, its meaning is not known. The string itself has no meaning except for its individual elements.

From this data, however, information can arise, if it is known in which context it is to be interpreted. By combining it with other data, a relationship is created that can be interpreted, and information can be generated. If the data 0815 is in the context "Customer Max Sample, Article 0815", the marketing department can interpret that the customer Max Sample ordered an article with the number 0815. The supplementation of data with other data depends on a subject's interest in knowledge or his

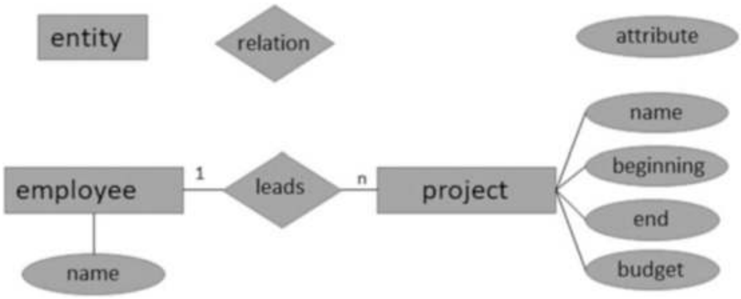


Figure 2.15: Example of an Entity-Relationship Diagram

intentions to use it. With the information produced in each case, a subject usually wants to influence an addressee, e.g., to perform an action. Information is thus to be understood as "statements that improve the degree of knowledge of a subject (information subject/user) about an object (information object) in a given situation and environment (information environment) in order to fulfil a task (information purpose)" [28] (cf., Fig. 2.14). Modeling is therefore a part of information management.

Information is important for politicians and business leaders, but also for every citizen of the world. It reflects a particular situation that applied at a particular point in time and usually allows an update into the future. It serves to make political, economic or personal decisions. When using information, the question always arises of who created it and what intentions that person has. The subjectivity of models thus plays an essential role here.

Entity-Relationship Model

In order to turn data into information, it must be combined with other data and the relevant relationships described. This creates a data model. The best-known method for describing data models is the Entity-Relationship Model (ERM). An Entity-Relationship Model consists of three main elements:

- Entities
  Entities are the object classes that are considered in the part of the real world that is of interest.
- Relationships or relations
  Relations describe the relationships between entities.
- Attributes
  Attributes are properties within the context of an entity.

Figure 2.15 shows an example of an ERM. It also indicates which symbols are generally used to express the main elements. However, one can also find ERM representations with notation elements of the Unified Modeling Language (UML).

Figure 2.15 describes the following situation:

An employee has a name. A project has a name, a start, an end and a budget. The so-called cardinality expresses the fact that an employee can lead several projects, but a project can only be led by exactly one employee.

In the modeling concept of classification, objects are combined to form object types (entity sets) and relationships are combined to form relationship types (relationship sets).

These types are differentiated according to:

- Entity type: Typecasting of similar entities (e.g., employee, project)
- Relationship type: Typecasting of similar relationships (e.g., employee leads project)
- Attribute type: Typecasting of similar properties (for example, name for the entity type employee). Attributes or combinations of attributes whose value(s) uniquely identify an entity are called identifying attribute(s) (e.g., the attribute project name identifies the entity type project).

Flowcharts

Flowcharts illustrate an execution sequence of activities or actions and are used in numerous application areas. They can describe the order in which actions are to be performed by people or other actors. Algorithms or computer programs are often documented in the form of flowcharts (e.g., program flowcharts). Due to the broad application of flowcharts, numerous variants have developed which take into account special circumstances of the respective field of application. For data processing, the symbolism for flowcharts was defined in the standards DIN 66001:1983-12 and ISO 5807:1985. Figure 2.16 shows an example of a flowchart.

In the explanation of ARIS in section 1.5, we addressed Event-driven Process Chains (EPC) as a model type for the control view. These EPCs are flowcharts of sequences of event nodes, function nodes (operations), and connectors. Arrows as edges represent control flows between the symbols. Functions and events (with the exception of start and end events) each have exactly one incoming and one outgoing edge. If functions are to create several events or if several events should trigger a function, connectors such as an exclusive or (XOR), must be used. The modeler can also express who executes a function with which IT support and what data is manipulated in doing so. For this purpose he assigns symbols for organizational units (for example, departments, jobs, roles), information objects (data), or application systems to the function nodes. These elements must be specified in the corresponding model types. This is referred to as extended Event-driven Process Chains (eEPC).

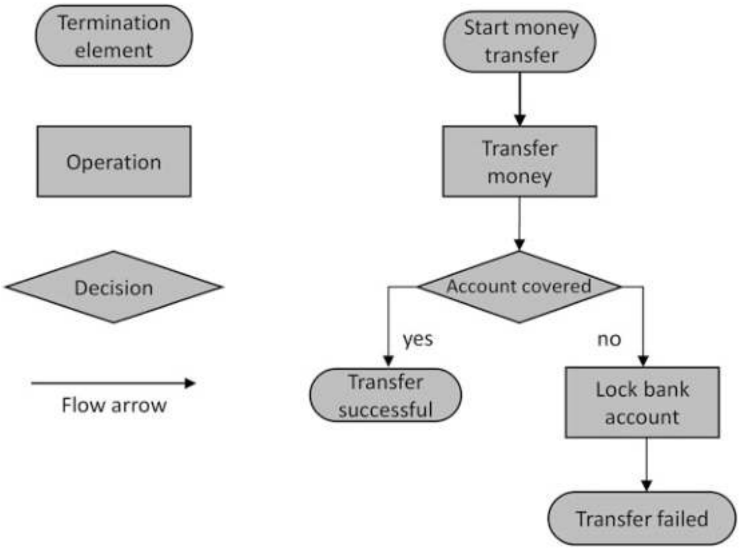


Figure 2.16: Flowchart for account transaction

We have already briefly indicated in section 1.5 that an eEPC, as an instrument to describe the control or process view; correlates the interaction between the elements of the other views and model types. Specifically, it expresses itself in the following way:

- Control view
  - An event is a state that occurs before or after a function. The symbol for an event is hexagonal.
  - A function (process) is an action or task that follows an event. Functions are symbolized by rectangles with rounded corners.
  - Connectors are used to split or join the control flow. The three connectors AND, OR and XOR are available, each represented in a small circle with the corresponding symbol. The decision as to which path is followed after a connector is made by the function preceding the connector.
- Functional view
  - The function nodes in the control view are linked with nodes from the function tree of the function view and thus specify the activity to be executed.
- Data view
  - Information objects are entities from the data model that are bound to carriers such as documents or other data stores. They represent inputs or outputs of the function to which they are connected by a directed edge. The symbol for an information object is a rectangle, the character as input or output is determined by the arrow direction of the connection edge.
- Organizational view
  - Organizational units show which elements from the organizational chart, modeled according to the organizational view, execute the activities (functions) in the process. Organizational units are connected to functions by undirected edges.

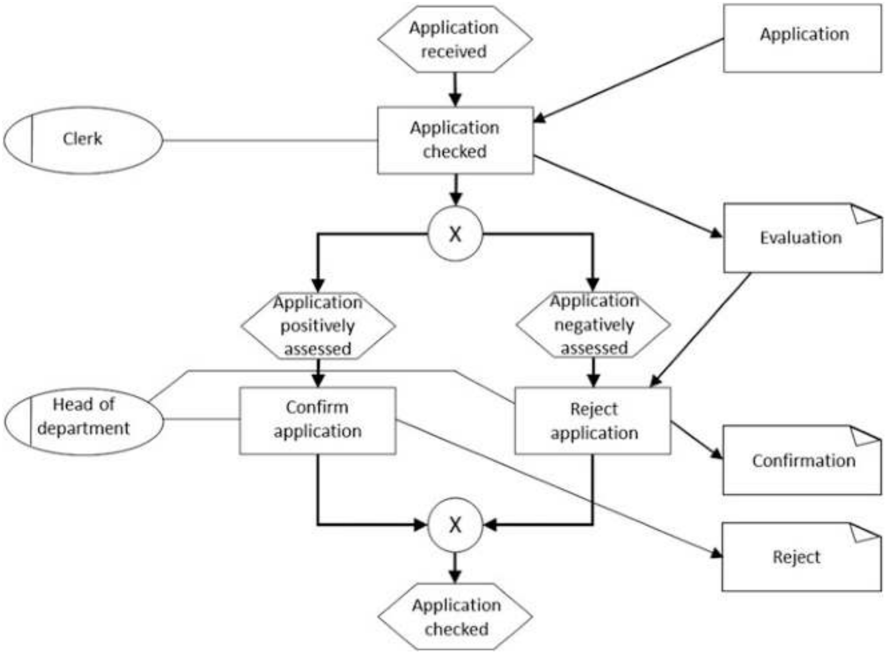


Figure 2.17: Extended Event-driven Process Chain for an account transaction

Figure 2.17 shows an example for an extended Event-driven Process Chain (eEPC).

Petri Nets [18]

One of the first and most widespread theoretical approaches to the description of parallelism are Petri nets.

Petri nets are used for the logical modeling of behavior. They consider the behavior of systems, usually information systems, under the following aspects:

- Activities to be carried out
- Preconditions and postconditions of an activity
- States of all conditions (possible values for each individual precondition or postcondition): The state of a condition is the distribution of so-called tokens as a state display to the pre- or post-range.
- Initial state (initial token)
- Procedures (possible consequences of activities)

A Petri net is a structure that is formally and mathematically precisely described as a directed graph with nodes consisting of two disjoint subsets marked with tokens. In the representation as quadruple the following applies: A Petri net is a quadruple: PN = (S, T, K, M) with

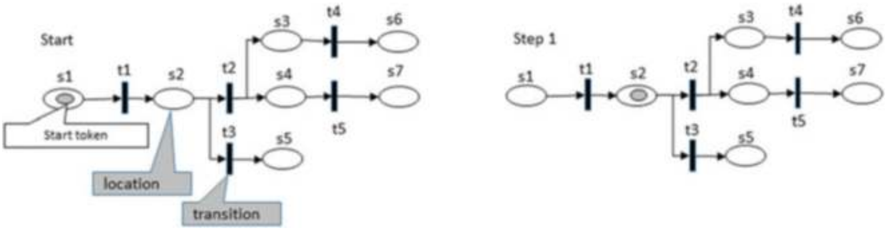


Figure 2.18: Petri net I

- s ε S: Places (i.e., conditions), for the description of states and/or conditions, buffers, memories or storage spaces. They are circular in the graph and are used to store information or tokens.
- t ε T: Transitions, describe state transitions, events, actions or activities and are shown in the graph as line, bar or cuboid forms. Their purpose is the processing of information.
- k ε K: Arcs are possibly weighted (i.e., numbered) connections between places and transitions, shown in the graph as arrows. They indicate the course of the transitions.
- m ε M: Tokens that represent the current state of the Petri net. Every network has an initial state, i.e., there is an initial token.

According to the above definition, arcs only go from place to transition or from transition to place. Input places of a transition t are places from which arcs run to the transition t. Output places of a transition t are places to which arcs of the transition t lead.

A transition can switch if there is at least one token in each input place. Switching means that a token is removed from each input place of the switching transition and a token is added to each output place.

The following figure 2.19 show a Petri net with its respective states after performing the switching operations. In the start state in Figure 2.18, only the place s1 has a token, the initial token or the start token. The place s1 is the only input place of transition t1, which can therefore switch. The token is removed from s1 and a token is added to the single exit place s2. The right half of the following figure 2.19 shows the state after switching t1.

The place s2 is the input place of the two transitions t2 and t3 (see left side of Fig. 2.19). So it could switch the transition t2 or t3, but not both. Which transition switches is random. This gives us a so-called non-deterministic state. If t3 switches, a token is added to the place s5 and a final state is reached, since no other arc leads out of s5. If the transition t2 switches, a token is added to the places s3 and s4, which are starting points of t2. The corresponding state is shown in the right half of Figure 2.19.

Now either the transition t4 or t5 can switch. Here one could conclude that these two transitions switch simultaneously. But this is not allowed in Petri nets. Only one

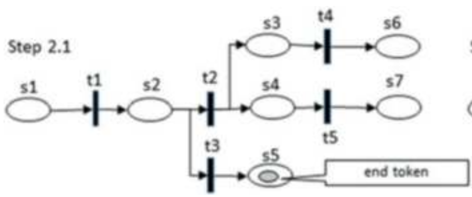


Figure 2.19: Petri net II

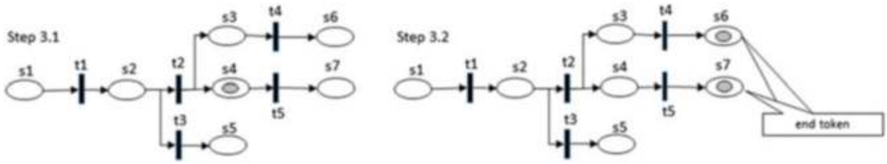


Figure 2.20: Petri net III

transition may switch at a time. Consequently, parallelism as observed in reality cannot be represented. However, it is arbitrary whether t4 or t5 switches first. In our example, t4 switches first and then t5. When both transitions have switched, a final state is reached again. Figure 2.20 shows the corresponding network states.

Petri net models allow the analysis and simulation of dynamic systems with concurrent and non-deterministic processes.

The type of Petri nets presented here is the basic version. It cannot be used to describe certain situations. For example, it is not possible to assign priorities for transitions, which is why, for example, the right-before-left rule at traffic junctions cannot be modeled.

In order to remedy such deficits, extensions have been introduced to map further aspects of reality to the model or to describe certain situations more compactly. Examples are multi-valued, colored or prioritized Petri nets. Details of such extensions can be found in the available literature, e.g., [29].

Calculus of Communicating Systems

The Calculus of Communicating Systems (CCS) was published by Robin Milner in 1980 (see [30]). This calculus enables the formal modeling of parallel communicating systems. This permits networked systems with a static topology to be described. CCS can be used to formally investigate the properties of programs such as deadlocks, bisimilarity, etc. CCS enables the description of the following aspects:

- Communication between actors via channels
- Interaction with the environment, i.e., reactivity
- Parallel composition
- Hiding actions from the environment (information hiding)
- Non-deterministic branches

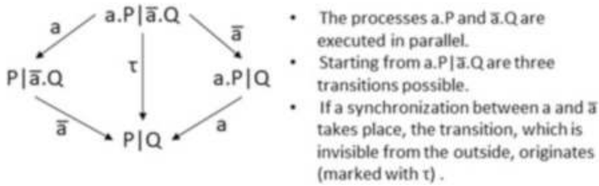


Figure 2.21: Process interaction with CCS

The course of a process is described as a tree, i.e., there is a root that represents the initial state and from where the individual branches originate. Each of the branches is marked. These markers represent the actions performed to move from one state to the next. A distinction is made between observable and unobservable actions. Unobservable actions can be executed at any time within a process without affecting other processes. Processes have no common variables.

Recursive expressions are used to describe the behavior of a process. Within behavior expressions, variables can be used to reference other behavior expressions.

The behavior expressions are described according to the following syntax, in which uppercase letters denote process names and lowercase letters denote actions.

- Empty process:
  Ø
- action
  Process a.P1 executes action a and then behaves like P1.
- process name
  With the expression A := P1 the process P1 gets the name A. Since recursive definitions are allowed, the expression P1 can contain the name A again.
- choice
  Process P1+P2 can be continued with either process P1 or process P2.
- parallel composition
  P1|P2 means that processes P1 and P2 are executed in parallel.
- renaming
  P1[b/a] describes the process P1, in which all actions with the denotation a are renamed to b.
- restriction
  P1\a denotes the process P1 without the action a.

Matching input and output actions in two different processes can synchronize and become an internal action τ. In general, the coaction for an action is marked with a line above the action name. These complementary actions are the send and receive actions. Figure 2.21 shows an example of the interaction between two processes.

The example in Figure 2.22 shows how a simple holiday request process can be described with CCS.

In the employee process, the vacation request send action is executed (action name with overline). The system then waits for the messages 'rejected' or 'approved'. The manager receives the vacation request message, which he replies to either with

$$\text{Employee} := \overline{\text{vacation request}}.\ (\text{Rejected} + \text{Approved}).\text{NIL}$$

$$\text{Manager} := \text{vacation request}.\ (\overline{\text{Rejected}} + \overline{\text{Approved}}.\overline{\text{approved vacation request}}).\text{NIL}$$

$$\text{Travel departement} := \text{approved vacation request}.\text{NIL}$$

$$\text{Vacation application process} := \text{Employee} \mid \text{Manager} \mid \text{Travel departement}$$

Figure 2.22 Example process in CCS

the message 'approved' or 'rejected'. If he sends the message 'approved', the message 'approved vacation request' is also sent afterwards. This message is received by the travel department. Messages are exchanged synchronously in CCS, that is, a sender waits until the receiver executes the corresponding receive action.

There are also formal rules of derivation for the informal interpretation. This makes process definitions accessible for formal evaluations.

π Calculus

CCS only allows static process structures. Communication relationships cannot be changed dynamically. The π Calculus (see [31]), also developed by Robin Milner, allows the representation of processes with changing structures. Any connections between components can be displayed and these connections can also change, or new ones can be created. Thus, the π Calculus is an extension of CCS to include concurrency. The notation in the π Calculus is largely based on the CCS notation. The following example explains the modeling possibilities of the π Calculus (see Figure 2.23).

The agent (process) P wants to send the value 7 to Q via a link a. However, the value is to be transmitted indirectly via another agent R.

In Figure 2.24 the individual steps for the execution of system O are shown.

The processes P, R and Q are executed in parallel. Process P sends via channel b the name a and then the name 7. Process P receives via channel b the two names. This means that each x is replaced by a and each z by 7. In Figure 2.24 this is the result after step 2. Now the name 7 can be sent via channel a, which is then accepted by process Q. Thus, the value 7 was sent to process Q via process R. The following graphic shows how the structure of System O is changed by its process flow.

Since the channel name a is transmitted from P to R, the processes R and Q are linked via channel a after the message has been accepted. This property shows that the π Calculus, in contrast to CCS, permits the modeling of structural changes (cf., Figure 2.25).

Communicating Sequential Processes

Communicating Sequential Processes (CSP) is a methodology for describing interaction between communicating processes. The idea was first introduced by Tony Hoare in 1978 as an imperative language (see [32]). It was then developed into a formal algebra and made famous in 1985 with the publication of the book Communicating Sequential Processes (see [33]).

- P:= $\bar{b}a.\ \bar{b}7.P'$ Process P transmits the value a via channel b and subsequently the value 7 and then behaves like P'.
- R:=b(x).b(z).$\overline{xz}$.0 Process R receives any value via channel b. This means in R, x and z are replaced by the received values.
- Q:=a(x).Q Process Q receives via channel a any value for x. The received value is used everywhere where x is located.
- O:=P I R I Q Entire system O

Figure 2.23: Process description with π Calculus

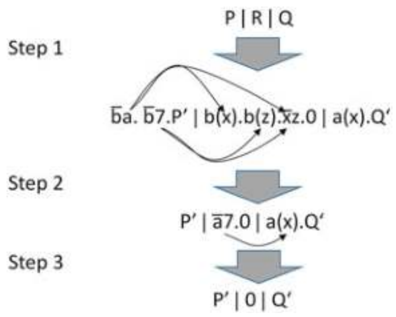


Figure 2.24: Execution steps of system O

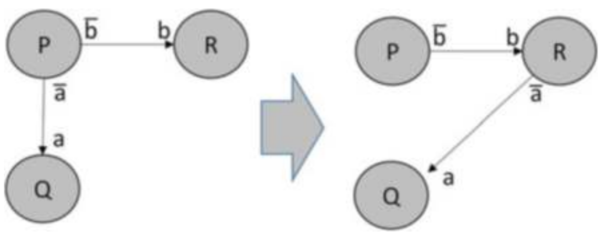


Figure 2.25: Structural change

In CSP, as in CCS, the number of processes is static. It cannot be changed during runtime and there are no common variables between processes. Instead, the processes 'know' each other and communicate with each other by sending and receiving messages. For sending, the send process P executes the output command Q! (expr) and the receiver process Q receives the input command P? (vars). Output and input commands are called corresponding if the sequence of expressions (expr) and the sequence of variables (vars) are of the same type in relation to their numbers and components. Analogous to CCS and the π Calculus, CSP is based on an unbuffered message exchange in which the send and receive processes must be explicitly named.

With Q!() and P?() a message without content is sent. Such messages are called signals and only serve to synchronize processes. If different signals are required, the distinction is made by means of type designators of the form Q!(Signal1) and P?(Signal1).

In addition to the unconditional message exchange described above, there is the receive instruction within a so-called guarded command. A guarded command is only executed if the preceding Boolean condition is true. The formula set

$$x > y; P?(z) - > x{:=}x + y; y{:=}z$$

is only executed if x is greater than y. Then the message is received by P if P is ready to send the message.

To be able to wait for messages from different senders, several guarded commands are combined to form an alternative instruction.

$$[\, x > y; P?(z) - > x{:=}x + y; y{:=}z \,||\, x < y; Q?(z) - > y{:=}x + y; y{:=}z \,]$$

In case $x > y$ the message P?(z) is expected, in case $x < y$ the message Q?(z). Alternative instructions can also be executed repeatedly. Syntactically, this is expressed by a $*$ in front of the alternative instruction.

$$*[\, x > y; P?(z) - > x{:=}x + y; y{:=}z \,||\, x < y; Q?(z) - > y{:=}x + y; y{:=}z \,]$$

The instruction is executed until none of the conditions are true, then the repetition is terminated.

The concepts regarding the concurrency of CSP serve as a design basis for the programming language Go.

Abstract State Machines

In computer science, an Abstract State Machine (ASM) is a model for the formal, operational description of algorithms. The states of an Abstract State Machine are general mathematical structures. The inventor of the model is Yuri Gurevich. Egon Börger has further developed the ASM for practical application [34].

Abstract State Machines (ASM) are finite sets of transition rules of the form

If condition then action

with which the states of an ASM are changed. Condition is any logical expression and action any action. As a rule, action is a value assignment of the form f(t1,. . .. . . tn) := s. The meaning of the rule is to execute the specified rule in the current state if the specified condition is met in that state. ASM states are generally defined as arbitrary sets of arbitrary elements with arbitrary functions (operations) and predicates defined on them. In the case of business objects, the elements are placeholders for values of any type and operations such as creating, duplicating, deleting, or algebraically manipulating objects.

A calculation step of an ASM in a certain state means that all actions for which the condition is true are executed simultaneously. Simultaneous execution can abstract from irrelevant sequences.

Several ASMs can run simultaneously and be linked via so-called controlled or monitored functions. A given ASM M can update controlled functions, but it cannot be modified by other ASMs in its environment. Monitored functions of a given ASM M can only be updated by its environment. By using controlled and monitored functions in pairs, a network of parallel coordinating ASMs can be set up.

Object-Oriented Models

The computer science models considered so far focus either on data or functions. The representation of data in an ERM and functions in a flowchart complement each other only in decoupled representations. The distinction between data and function views in ARIS makes this clear. Object-oriented modeling no longer reduces these individual entities into their separate parts but considers them as an integrated whole in which the individual components are interconnected and interdependent.

An object-oriented model is a view of a complex system in which the system is described by the interaction of objects. This type of modeling is intended to reduce the complexity of the description of situations to be mapped in software. The object orientation considers the entities occurring in the real world as objects. A telephone is just as much an object as a bicycle, a person or an employee. Such objects in turn consist of other objects such as screws, rods, arms, feet, head, etc. As is usual in modeling, the objects are reduced to their properties that are significant in the respective situation. For example, an employee in a payroll accounting system is reduced to name, address, employee number, agreed income, tax class, and so on.

The objects considered in a model are not designed individually. A rough blueprint with similar properties is created for similar objects.

For example, one models the properties of books for a library application that are identical for all books. Such general descriptions of objects are called classes in object orientation. These classes are then used to create the required concrete objects (instances) within the model. The representation of reality in a model is therefore a two-stage process. First, similar objects of reality are identified and described as classes. The individual objects are then created as instances of a class. A class thus describes the structure of a set of similar objects. Figure 2.26 shows a class-object relationship according to which the book "Subject-oriented Process Management" is an instance of the class Book. The notation used comes from the Unified Modeling Language (UML). UML is a language standardized by the Object Management Group for the description of object-oriented models.

The properties of a class are

- their components and the data and information they contain, also called attributes,
- the operations defined on the components and their parameters (methods) with which an object can be manipulated or its status queried, and
- conditions, prerequisites, and rules that the objects must fulfill (constraints).

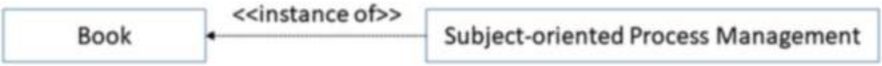


Figure 2.26: Relationship class-object

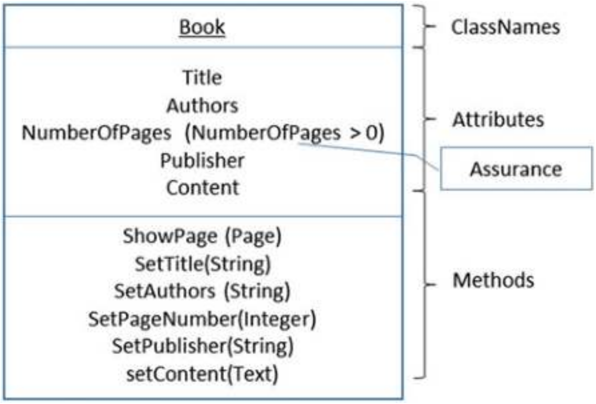


Figure 2.27: Description of the Book class

Figure 2.27 shows an example of the description of the Book class. This class has five attributes. The Page Number attribute has a constraint that the page number must be positive. The class allows the access and manipulation of its data with six methods. Thus, the title, the authors, the page number, the publisher and the content can be set (Set). The operation (method) DisplayPage(Page) can be used to "read" the contents of the specified page from the book.

Objects can now be instantiated from such a class definition. The operations can then be used to set the corresponding attributes for each of these instances.

An object-oriented model describes not only the definitions of the classes and the associated objects, but also the relationships between the classes. The types of relationships are:

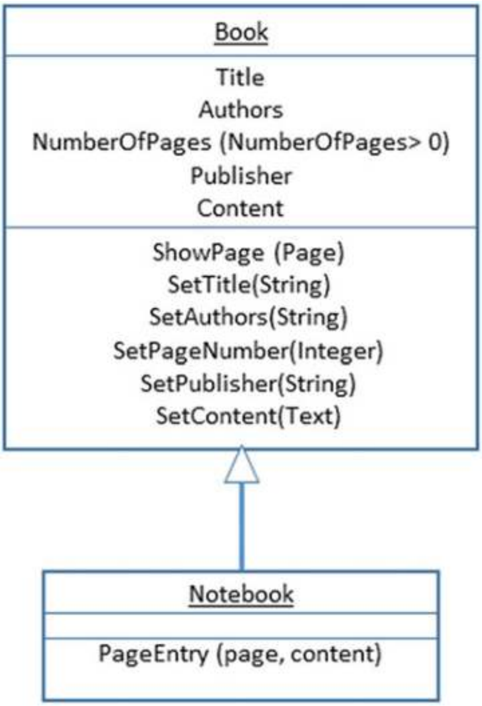
- Inheritance

  Properties can be passed from one class to the next. This is called inheritance. The class that is used as the basis for inheritance is called the superclass, and the class that inherits is called the subclass. Thus, a class Notebook is a subclass of the class Book. The method "PageEntry(page, content)" is added to the Notebook class. This allows a text to be entered on the specified page; otherwise, the Notebook subclass inherits all attributes and methods from the Book class. Figure 2.28 shows the inheritance relationship between Book and Notebook. The triangular arrow points from the subclass to the superclass.

- Associations

  An association is a relationship between different objects of one or more classes. Associations are represented as a simple line between two classes. The line can be provided with a name (identifier) and number specification. The

Figure 2.28: Inheritance of properties



associated classes can also receive names regarding the relationship. Figure 2.29 illustrates an example of an association. A person (employee) can be employed by no company or by just one company (employer) and a company can employ one or more persons. Associations are very similar to entity-relationship diagrams.

- Aggregations

  Aggregations are a variant of associations. This is also a relationship between two classes, but with the peculiarity that the two classes are related to each other like one part of a whole. An aggregation is made up of a quantity of individual parts. Figure 2.30 shows an aggregation example in which a company consists of departments and a department of employees.

- Composition

  A special form of aggregation is composition. Here the whole depends on the existence of its individual parts. Figure 2.31 shows a change in the above aggregation. A department does not exist anymore if no one belongs to it.

- Objects

  Communicate with each other, i.e., one object sends messages to another object. The messages then trigger the associated operations. An object therefore only understands the messages for which it contains the corresponding operations. Figure 2.32 shows a person's communication with the notebook to enter a new note.

Figure 2.29: Associations between objects

Figure 2.30: Multilevel aggregation

Figure 2.31: Composition

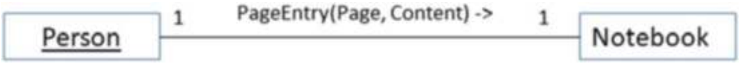


Figure 2.32: Message between objects

The described constructs form the nucleus of the object-oriented modeling approach. In addition, there are numerous enhancements for depicting certain circumstances, such as the so-called container classes.

2.7 Agent/Actor-Oriented Models

So far, there is no standardized and generally accepted definition of the agent concept. The term is defined in detail somewhat differently depending on the application domain [34, 35]. What all these definitions have in common, however, is that an agent is a scoped unit that is able to pursue the tasks assigned to it flexibly, interactively and autonomously. The term actor is often used as a synonym for the term agent. Thus [36] define a business actor as "... an entity that is capable of performing behavior". Multi-agent systems consist of several agents that exchange messages synchronously or asynchronously. Multi-agent systems can map the structures of software systems or serve as models for social systems. Depending on the application situation, a distinction can then be made between software agents and human agents.

In a way, the processes in CCS, the π Calculus and CSP can be seen as multi-agent systems. The term process is used there analogously to the terms agent and actor.

An agent-oriented model therefore contains the agents, the communication paths and the messages that are exchanged. A collection of agent-oriented modeling languages and the corresponding procedures can be found in [37, 38].

2.8 Conclusion: Models for Business Processes

Business process models depict the part of reality of business processes under consideration. The subjective understanding of the business process concept influences which process aspects are considered essential and therefore placed in the foreground in the models. The intentions and interests of the person creating the model are reflected here. Consequences are numerous interpretations of the business process concept, each of which is neither right nor wrong, but merely sets different focal points.

The following definitions of the term business process are examples of this:

1. "Sequence of value creation activities (value creation) with one or more inputs and a customer benefit generating output"[39].
2. "A process is the closed, temporal and logical sequence of activities that are necessary for the processing of a business-relevant object."[40]

Both definitions focus on the necessary activities and their consequences. The first example additionally mentions the input and the output with the customer benefit, while in the second definition the processing of the business objects is included instead.

On the other hand, neither definition includes the actors and necessary resources. They do not consider by whom and with what the activities are carried out. There is no relation to the organization in which a business process is embedded, or to which IT applications or other resources are required to execute it.

We therefore follow an understanding of the term based on that of Gerhard Schewe [41] that also takes these missing aspects into account [41]:

1. A process is the sum of linked activities (tasks)
2. carried out by actors (people, systems as task bearers)
3. in logical and chronological order
4. with aids (equipment, information)
5. for processing a business object
6. to satisfy a customer need (and thus contribute to value creation)
7. which includes a defined beginning and input
8. as well as a defined end and a result.

As already explained in Chapter 1, we will reorganize the components somewhat and group them as follows:

1. Process strategy: A process has

   a. a defined start and input (start event)
   b. and a defined end with a result
   c. that contributes to the satisfaction of a customer's needs (and thus to the creation of value)

2. Process logic: A process

    a. is the sum of linked activities (tasks)
    b. which, after the start event, are used by actors
    c. in logical and chronological order
    d. for processing a business object in order to
    e. generate the desired result.

3. Process realization: A process is realized

    a. with people and/or machines, that take over the tasks of the respective actor
    b. and carry these tasks out with tools (equipment, information, application programs, etc.).

With this understanding of business processes, the relationship between the various models from different domains described in this chapter and Business Process Management becomes clear. Figure 2.33 shows the associated integrative character of business process models.

Habermas's and Luhmann's models deal with aspects of social systems and organizations. They describe which components and relationships make up an organization and how people are positioned in it. Complex organizations can, for example, be structured into sub-organizations on the basis of operational functions, range of services, geographical aspects or combinations of these. The result is the organization chart.

Among other things, business models consider the aspects of customers, suppliers, partners and added value and thus look at the external service relationships on the one hand. On the other hand, they also establish the connection to the more

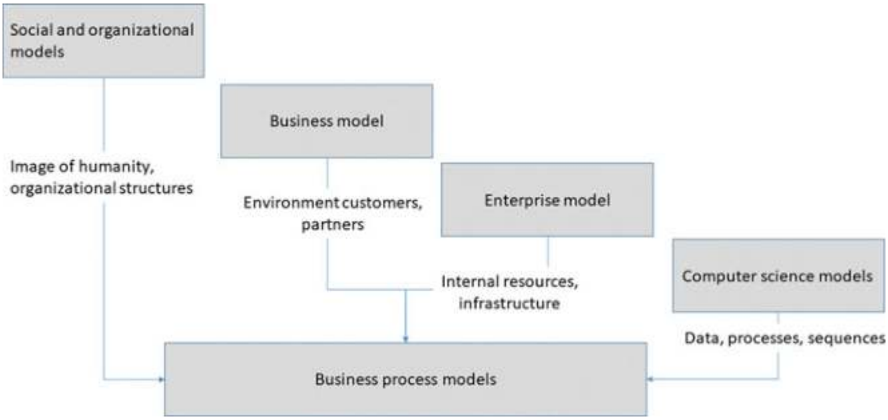


Figure 2.33: Integration of different models through business process models

inwardly oriented enterprise architecture, especially with value promises (products and services), activities and resources. At the business level, the organizational structure is modeled within this enterprise architecture with the personnel resources, the processes, and the logical business objects. The linkage with the technical layer of the enterprise architecture leads to the models from computer science. These describe, for example, data structures, control flows and algorithms for programs as well as the design and interaction of other information and communication technology components necessary for the execution of desired actions within the framework of process support and automation.

Models of business informatics generally try to unite computer science with models of social systems. These converge in business process models.

References

1. Gabriel, M. (2015). Why the world doesn't exist, Ullstein Taschenbuch.
2. Alle, V. (2015). *Value networks and true nature of collaboration*. Tampa, FL: Meghan Kiffer Press.
3. Stachowiak, H. (1973). *General model theory*. New York: Springer.
4. Kieser, A., & Walgenbach, P. (2007). *Organization*. Stuttgart: Schäffer-Pöschl.
5. Crowther, D., & Green, M. (2004). *Organizational theory*. London: Chartered Institute of Personnel Development.
6. Schreyögg, G. (2008). *Organization* (5th ed.). Wiesbaden: Gabler.
7. Kieser, A., & Ebers, M. (Eds.). (2006). *Organizational theories*. Stuttgart: Kohlhammer.
8. Szyperski, N. (1980). Information needs. In E. Grochla (Ed.), *Handwörterbuch der Organization* (2nd ed., p. 904). Stuttgart: C.E. Poeschel.
9. Krcmer, H. (2017). Information, encyclopedia of business informatics. *Online Lexicon.* Last accessed July 31, 2017, from http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/daten-wissen/Informationsmanagement/Information-/index.html
10. Berghaus, M. (2004). *Luhmann made easy* (2nd ed.). Cologne: Böhlau.
11. Luhmann, N. (1984). *Social systems*. Frankfurt: Suhrkamp.
12. Scheer, C., Deelmann, T., & Loos, P. (2003). *Business models and internet-based business models – Definition and participant model*. Working Papers of the Research Group Information Systems & Management, University of Mainz
13. Wirtz, B. W., Schilke, O., & Ullrich, S. (2010). Strategic development of business models implications of the web 2.0 for creating value on the internet. *Long Range Planning, 43*, 272e290. Elsevier 2010.
14. Osterwalder, A., & Pigneur, Y. (2010). *Business model generation*. Hoboken: Wiley.
15. Siemoneit, O. (2010). A theory of science in business administration, Dissertation. University of Stuttgart.
16. Kieser, A., & Ebers, M. (2006). *Organizational theories*. Stuttgart: Kohlhammer.
17. Wieß, G., & Jakob, R. (2005). *Agent-oriented software development*. Berlin: Springer.
18. Brushwood, W. (2010, July 15). *Petri nets: Modeling techniques, analysis methods, case studies* (2010th ed.). Vieweg+Teubner Publishers.
19. ITIL. (2018). *Best practice*. Last Access February 2018, from https://www.axelos.com/best-practice-solutions/itil
20. Matthes, D. (2011). *Enterprises architecture compendium*. Heidelberg: Springer.
21. Lehner, F. (1995) Models and modeling in business informatics - Attempting to determine the current situation. In H. Wächter (Ed.), *Selbstverständnis betriebswirtschaftlicher Forschung und Lehre*. Wiesbaden: Gabler Publishing House.

22. Kaplan, R. S., & Norten, D. P. (1997). *Balanced scorecard*. Stuttgart: Schaeffer-Poeschl Publishers.
23. Robin Milner, A. (1980). *Calculus of communicating systems*, Lecture Notes in Computer Science. Springer.
24. Milner, R. (1999). *Communicating and mobile systems: The π-calculus*. New York: Cambridge University Press.
25. Hoare, C. A. R. (1978). Communicating sequential processes. *Communications of the ACM, 21* (8), 666–677.
26. Hoare, C. A. R. (1985). *Communicating sequential processes*. Englewood Cliffs, NJ: Prentice Hall.
27. Open Group. (2019). ArchiMate® 3.1 specification. https://pubs.opengroup.org/architecture/archimate3-doc/
28. Börger, E., & Stärk, R. (2003). *Abstract state machines*. Springer.
29. Lankhorst, M., et al. (2017). *Enterprise architecture at work* (4th ed.). Berlin: Springer.
30. Wooldridge, M. (2002). *An introduction to multi agent systems*. Chichester: Wiley.
31. Cossetino, M., et al. (2014). *Handbook on agent-oriented design processes*. Berlin: Springer.
32. Schewe, G. (2017). *Business process, Gabler business encyclopedia*. Last accessed July 31, 2017 from http://wirtschaftslexikon.gabler.de/Definition/geschaeftsprozess.html#definition
33. Becker, J. (2012, October). Business process modeling, encyclopedia of business informatics. *Online encyclopedia.* Last accessed July 31, 2017, from http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/daten-wissen/Informationsmanagement/Information-/index.html
34. Schewe, G. (2017). *Business process, Gabler business encyclopedia.* Last accessed July 31, 2017 from http://wirtschaftslexikon.gabler.de/Definition/geschaeftsprozess.html#definition
35. Stähler, P. (2002, September). Business models as an unit of analysis for Strategizing. In P. Steels (Ed.), *Business models in the digital economy: Characteristics, strategies and effects*. Cologne: Josef Eul.
36. Habermas, J. (1981). *Theory of communicative action* (Vol. 1, 2014th ed.). Frankfurt am Main: Suhrkamp Paperback Science.
37. Römpp, G. (2015). *Habermas made easy*. Cologne: Böhlau.
38. Sowa, J. F., & Zachman, J. A. (1992). Extending and formalizing the framework for information systems architecture. *IBM Systems Journal, 31*(3), 590–616.
39. TOGAF. (2010, October). *TOGAF version 9*. Van Haren Publishing, Zaltbommel.
40. Franklin, S., & Graesser, A. (2017). *Is it an agent, or just a program? A taxonomy for autonomous agents.* Institute For Intelligent Systems, University of Memphis, Memphis, TN. Last accessed September 2017, from https://pdfs.semanticscholar.org/288d/7952b6648749fcbdcedabedf8f43cf7fda52.pdf
41. COBIT 5. (2018). Last Access February 2018, from http://www.isaca.org/COBIT/Pages/COBIT-5-german.aspx

3 Modeling Languages

Modeling languages determine the concepts that can be used to describe an extract from perceived reality and how these concepts can be put into mutual relationship. Modeling languages thus provide the vocabulary and the grammar needed to represent real-life situations in models. They will subsequently be considered in a structured way.

The selection of a specific modeling language creates a well-defined point of reference for modeling, to which all actors involved - whether they are actively involved in the creation or passively affected by the model as consumers - can refer [1]. The term “actors” here is not necessarily restricted to humans, who are provided with a common, defined vocabulary by the modeling language in order to form a common understanding of the situation being modeled. It also refers to computer systems that are given the opportunity to further process models with automation support (e.g., in order to use them as a basis for workflow support), as the semantics of model elements are exactly specified in the modeling language (ibid.).

The choice of the modeling language therefore determines what can be represented in a model, and what cannot be made visible because the modeling language does not offer any appropriate concepts. The choice of modeling language also influences the usability of the model for different target groups [2, 3]. Some modeling languages are designed for supporting communication among human actors and therefore may allow a "vague" representation of model semantics [4]. Other modeling languages require a more exact specification of semantics. They are preferably used to prepare models for use in IT systems. The choice of a modeling language therefore is dependent on the respective objective of the modeling process. It thus is a major step toward successful support of the activities the modeling process is embedded in.

In this chapter we provide the foundation for the appropriate selection of a suitable modeling language to fulfill the requirements of both human actors as well as machines and present different languages with their respective objectives and language elements. The focus here is placed on the mapping of business processes. All selected languages therefore allow the behavior of actors in

A. Fleischmann et al., *Contextual Process Digitalization*,
https://doi.org/10.1007/978-3-030-38300-8_3

organizations to be mapped in a broader sense. The languages, however, fundamentally differ in what is considered to be an "actor" and how the behavior of such can be described. The reasons for these differences can be found in the historical context of the languages and their respective objectives. The arrangement of the following sections therefore follows a historical perspective in order to clearly highlight the relationships between the languages and their origins. Finally, we discuss for each of the languages to which extent they allow the mapping of processes according to the definition used in this book, and where they have their respective focal points or gaps in representation capabilities.

3.1 Overview

Flowcharts, which are still used today, are one of the oldest modeling languages to describe processes. They were originally designed to represent the control flow in a computer program. Due to the generality of their language elements, they are also used for modeling processes in organizations and are therefore presented here as an introduction to graphical process modeling. They introduce the concept of branching in the control flow to represent alternative paths.

For a long time *eEPCs* ("extended Event-driven Process Chains") [5] were a de facto industry standard for Business Process Modeling in Europe. In addition to their capability to map process flows, they also include elements to include responsibilities, data or services in models. This enables the modeling of business processes in their organizational context. In addition, they allow explicit modeling of parallel activities and thus go beyond the expressiveness of flowcharts in the representation of the workflow.

Historically, Unified Modeling Language (*UML*) activity diagrams *can be seen* as a further development of flowcharts to model software processes [6]. As part of the UML, they today represent the de facto standard for representing control flows in software. Like eEPCs, they also allow the mapping of parallel process flows. Using activity diagrams, we introduce the structuring of models through partitioning and nesting and show how models can be structured based on responsibilities and not just on the workflow itself.

BPMN (Business Process Modeling Notation) is the predominant standard for representing business processes today [7]. Originally derived from several different modeling languages, including activity diagrams, BPMN was explicitly designed for representing business processes. In this context, the idea behind BPMN was that it should enable modeling with different objectives - from communication support to execution in Workflow Management Systems. From a conceptual language point of view, BPMN is particularly interesting with regard to its possibilities for compact representation of complex process flows (such as exception handling).

S-BPM (Subject-oriented Business Process Modeling) is a modeling approach that puts the actors involved in a business process and their interactions at the center of the modeling process [8]. The resulting modeling language is characterized by a small number of language elements and comprehensive expressiveness for mapping business processes. On the one hand, it is presented here as an example of a not primarily flow-oriented approach to Business Process Modeling, and on the other hand, it represents, in terms of language conception, an alternative to BPMN with its very extensive set of modeling elements.

3.2 Flowcharts

Flowcharts allow the depiction of simple, sequential processes. A sequential process is characterized by the fact that no more than one activity is performed at any one time – thus parallel processes cannot be represented. Flowcharts were first described in the context of industrial production planning in the 1920s. At the end of the 1940s, they were adapted for the description of processes in the emerging Information Technology sector. Since the mid-1960s, they have been used as a standardized form of representation for computer program sequences. To this day, they are used to represent flows in computer programs or processes in organizations, as long as their complexity does not exceed the expressiveness of a flowchart.

The limitation of flowcharts' expressiveness is due to their historical development. In both industrial production planning and in early computer systems, it was not necessary to be able to represent parallel processes. Due to the limited computing resources available (only one CPU or only one processor core), there was no need to provide language elements to design parallel software processes.

3.2.1 Notation Elements

Flowcharts today exist in many different variants, which mainly differ in the notation used (i.e., the graphical representation). The semantics of the language elements, however, is common to all (cf., Figure 3.1). Basically, any number of *operations* (i.e., activities, represented as rectangles) is defined. These operations are put into an execution sequence by means of *directed connections* (represented as arrows). *R*ounded *rectangles* indicate the start and end of a process.

An essential means of describing processes, both in a computer program and in organizations, is the representation of alternative operations. The selection of an alternative usually depends on a condition that can be checked - in a computer program, this could be exceeding a limit for the value of a variable, in an organization, the existence of a particular document or the decision of a person responsible for executing the process could constitute the criterion for selecting an alternative. Alternatives are represented in flowcharts by *branches* (represented as diamonds),

Figure 3.1: Notation of flowcharts

which are connected to the previous operation via an incoming arrow and to the alternatively executed subsequent operations by two outgoing arrows. The restriction to exactly two (and not several) subsequent operations is also due to the origin of flowcharts from the representation of computer programs, since these usually are only capable of evaluating a condition in a binary way (i.e., as true or false). If a condition is to be checked for more than two values, several branches must be cascaded in flowcharts. The outgoing connections are labeled with the respective characteristic value at which the program is continued after the condition has been checked. A repeated execution of operations (for example, as long as there are still documents that need to be processed) is represented by jumping back to an earlier operation via an outgoing connection from a branch. The other outgoing connection then progresses to the operation that is executed when the repeated execution is complete.

In addition to these basic elements, flowcharts in some variants also offer special language elements that are used mainly to represent specific operations in the area of application (such as input or output operations in computer systems). However, these are not relevant for the conceptual understanding of flowcharts and especially not for their application in Business Process Modeling.

In the following, we present the use of the notation elements using examples from Business Process Modeling. The notation used is based on the symbols defined by ANSI and DIN in the 1960s.

3.2.2 Examples

The example in Figure 3.2 shows a process with a single operation in which an application (for which we have no detailed information here) is processed. The process ends after the processing of the application has been completed.

In the example in Figure 3.3, the process is extended by a decision. The application is checked, and the result of this check allows a decision to be taken on its confirmation or rejection. The confirmation or rejection itself are, in turn, operations. The outgoing connections are merged after the alternative branches are completed.

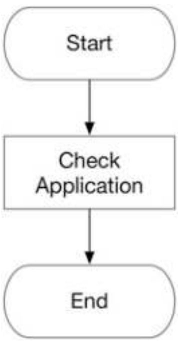
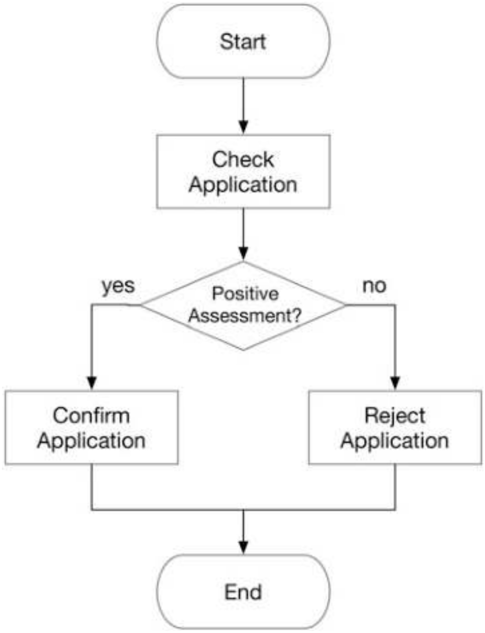


Figure 3.2: Simple flowchart

Figure 3.3: Flowchart with binary decision



Decisions with more than two possible outputs must be represented in flowcharts using cascaded decision elements. This can be seen in the example in Figure 3.4. Obviously, our applications are investment requests. The first decision now examines whether the sum of investment costs is less than EUR 1,000. If this is the case, the request is confirmed directly. If this is not the case, a second decision is taken. It checks whether the application sum is less than EUR 10,000. If this is the case, the application sum is between EUR 1,000 and EUR 9,999, which leads to an examination of the attachments to the application. If the application amount is not less than EUR 10,000.-, i.e., EUR 10,000.- or more, the application will be forwarded. We don't learn anything about the destination of the forwarding here, because flowcharts don't offer notation elements for depicting this information.

As already mentioned, branches can also be used to repeat parts of a process (cf., Figure 3.5). To do this, the decision is inserted at the end of the part to be repeated and an outgoing branch leads back to the point prior to the first operation to be repeated. The other outgoing branch continues the process after the repetition is completed. In the above example, applications are processed as long as there are more applications available. It is important to understand that at least one application must be processed before a decision can be reviewed. If the process should be able to finish if there are no applications at all, an additional decision would have to be inserted at the beginning of the process ("Applications present?"), from which an outgoing connection ("no") leads directly to the end of the process. The other outgoing connection ("yes") would lead to the process flow previously described.

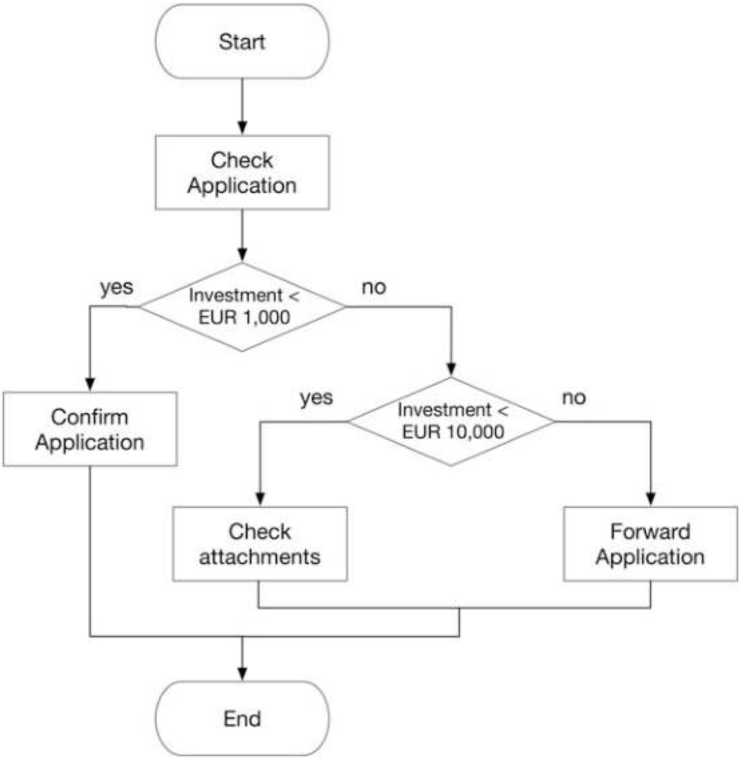


Figure 3.4: Flowchart with multiple-outcome decision

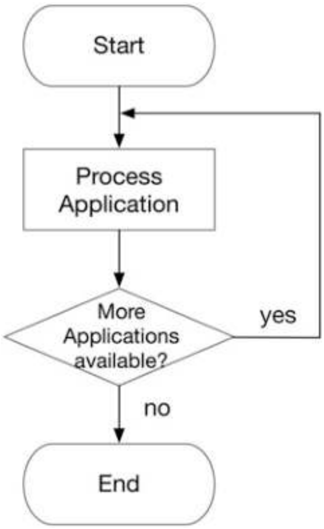


Figure 3.5: Flowchart with repeated execution of process parts

3.2.3 Classification

Flowcharts are a simple way to represent business processes in terms of the logical sequence of activities they contain. Other aspects of a business process, such as data or responsibilities, are not accounted for in the language and cannot therefore be represented.

It is also not possible to represent parallel process flows. This is a major reason why flowcharts are partly being replaced by more recent languages such as UML activity diagrams or BPMN, which offer constructs for executing operations in parallel. Since flowcharts do not offer any element to depict responsibilities, it is not possible to represent communication processes - this restriction also is addressed in more modern languages, which we introduce in the following sections.

3.3 Event-Driven Process Chains

For a long time Event-driven Process Chains (EPCs) were the de-facto industry standard in Europe for representing business process models. They were developed as part of the ARIS concept already presented in Section 1.5 and there serve as a means of representing models of the control view of an organization - i.e., the view that deals with the processes in an organization and the associated links between its resources. Resources here can be both acting members and/or departments of an organization from an active perspective, as well as the required and/or manipulated data or goods from a passive perspective.

EPCs links the functions that an organization is capable of performing based on upcoming events during business operation. The basic principle of representation is that a function is always triggered by an event - that is, a function must always be preceded by an event in an EPC model in order to determine whether the execution of a function can be started. In the more comprehensive variant "extended EPC" (eEPC), the functions are linked to the elements of the other ARIS views relevant for their execution. In particular, the responsible actors, roles, or organizational units can be assigned from the organizational view, and the relevant documents or data objects from the data view. If a function leads to a billable service, this can be represented by elements taken from the service view.

In addition to the representation of decisions, EPCs also enable the mapping of parallel process flows of a business process. For this purpose, it provides additional notation elements. By way of AND connectors process branches run in parallel can be displayed. The XOR connector is used to represent decisions for which exactly one alternative has to be selected (this corresponds to the decision element in flowcharts). The OR connector can be used to represent processes in which one or more alternatives can be chosen.

3.3.1 Notation Elements of EPCs

The basic element for describing business processes in EPCs is the *function* (similar to the *operation* in flowcharts, cf., Figure 3.6). However, the sequence of functions in a process is not determined exclusively by connection arrows. By using *events the* sequence is specified here in more detail. Every function is triggered by an event and subsequently creates one or more events itself. A process is thus represented by a sequence of events and functions, whereby events and functions always alternate. When naming these, functions should always describe a performable activity (e.g., "Check request"), while events should always describe a state (e.g., "Confirmed request" or "Rejected request").

An EPC always starts and ends with an event. Events that trigger a process are referred to as start events. Events that describe the completion of a process are referred to as end events. Subsequent processes can be triggered by end events of a previous process, that is, an end event can be a triggering start event in another process.

By using different connectors in combination with events, it is possible to represent different control flow variants of a process within a model. This includes the possibility of executing functions in parallel (if these are not dependent on each other). The AND, OR and XOR connectors can be used for this purpose:

If an *AND connector* with several outgoing connections is used, all outgoing paths are traversed in parallel. These paths are then usually joined at a later point in time with another AND connector. The function after the joining AND connector is not executed until all the incoming paths have been completed.

An *OR connector* with multiple outgoing connections indicates that one or more of the following paths are traversed in parallel. These paths are usually joined again at a later point with another OR connector, whereby the subsequent function is only carried out when exactly those paths chosen at the original OR branch have been completed. It is important here that the paths to be activated must be selected at the time of reaching the OR connector – it cannot be used to trigger further paths at a later point in time, should this become necessary during execution. Each OR connector must be followed by an event in each of the paths that could be triggered by the function preceding the OR connector. The paths whose first events actually occur are activated during runtime.

Finally, an *XOR connector* is used for representing "exclusive OR" or "either, or" decisions. Using an XOR connector with multiple outgoing connections means that exactly one of the following paths is selected when the process is executed. It is therefore suitable for representing mutually exclusive alternatives in processes. The

Figure 3.6: Notation of EPCs

paths are joined again with an XOR connector, and the subsequent function is performed when the selected path has been completed. As for OR connectors, XORs must also be followed by an event on each path that could be triggered by the preceding function. These events must be mutually exclusive. The path with the first event that actually occurs is activated during runtime. In contrast to flowcharts, it is also possible to describe more than two alternatives here, as long as the events used are mutually exclusive.

3.3.2 Examples of EPCs

We here use the same examples as shown for flowcharts to visualize the differences in the notation.

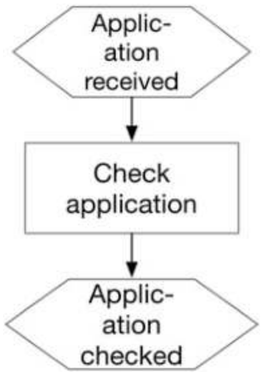
The example in Figure 3.7 shows a process with a single function in which an application (for which we have no further information here) is processed. The process ends after the application has been checked.

In the example in Figure 3.8, the process is extended by a decision. The application is checked, and the result of this check allows making a decision on its positive or negative assessment. The confirmation or rejection itself are, in turn, functions. The outgoing connections are joined with an XOR connector after the alternate branches are completed.

The representation of decisions that have more than two possible options is easier here than with flowcharts. In Figure 3.9, the XOR connector is followed by three events that all refer to the investment amount and are mutually exclusive.

The XOR connector can also be used to repeat parts of a process (cf., Figure 3.10). To do this, the connector is inserted at the end of the part to be repeated and an outgoing connection (line) links back to the event that triggers the part to be repeated. The other outgoing branch continues the part of the process to be executed after the repetition has been completed. It must lead to an event that triggers the termination of the repetition. In the above example, applications are processed as long as further applications are available.

Figure 3.7: Simple EPC



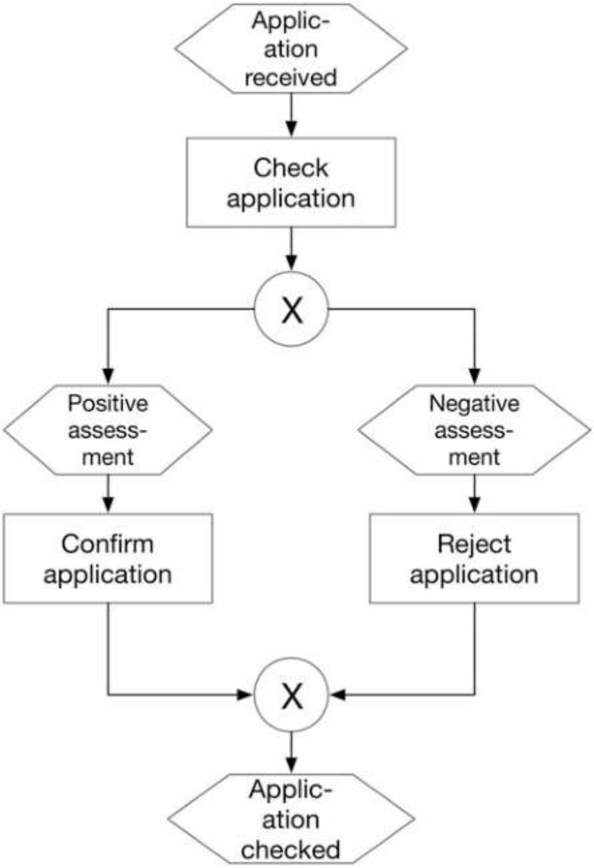


Figure 3.8: EPC with binary decision

The AND connector can be used if two functions can be executed independently of each other (cf., Figure 3.11). The process modeled in the above example is therefore only correct if the attachments can actually be checked independently of the application. If this is not the case, the two functions would have to be arranged sequentially. From the point of view of modeling, the AND connector, unlike the other connectors, does not have to be immediately followed by events, since no decision is made. All outgoing branches are activated in any case.

Figure 3.12 shows an example of the use of the OR connector. Here we assume that an application may contain offers or comments, or both (but must at least contain offers or comments). If offers and comments were completely optional (i.e., if both could be missing), their fundamental necessity would have to be examined with an upstream XOR connector and corresponding events. Alternatively, an additional branch could be added to the OR connector with an event "Application alone is sufficient". In both cases, the condition that events and functions must alternate on all paths through the process must not be violated. If not otherwise possible, this must be ensured by a "dummy" function that does not lead to any actual activity.

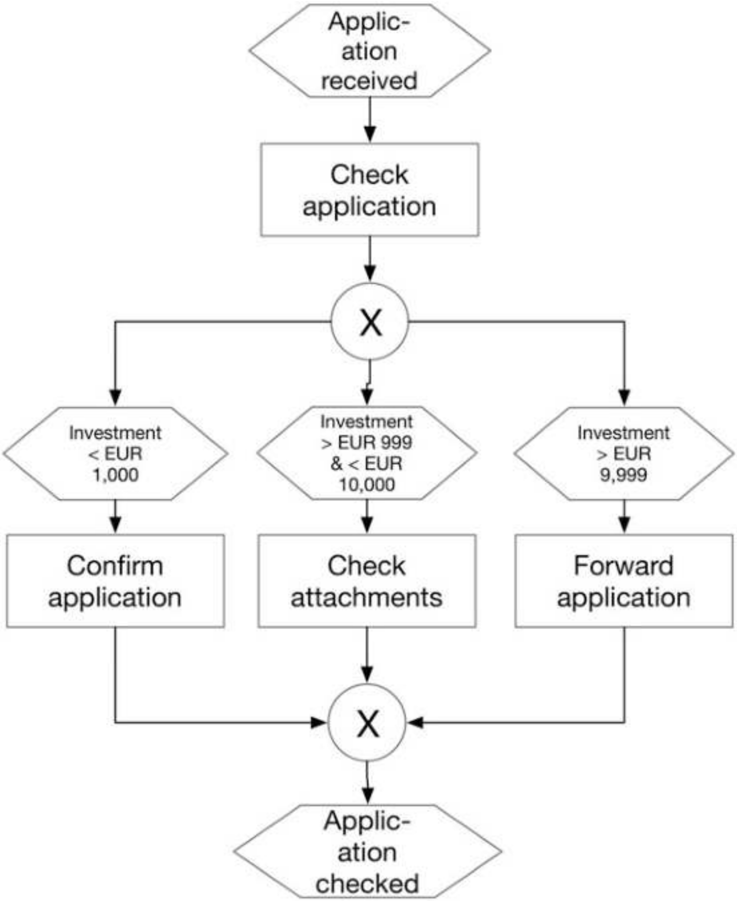


Figure 3.9: EPC with multi-outcome decision

3.3.3 Supplementary Notation Elements in eEPCs

The eEPC supplements the business process depicted in an EPC with information about its execution context. In particular, responsibilities and resource requirements are assigned to the functions here. The basic rules of an EPC remain unchanged. The additional elements can only be assigned to functions - events are not affected. At this point we do not provide a complete description of the possible elements of EPCs but focus on the most common ones (cf., Figure 3.13).

Responsibilities are represented by *organizational units.* Such units do not usually represent concrete persons but abstractly identify the name of a role (e.g., managing director) or department (e.g., financial accounting). This ensures that the specification of a process is independent of the availability of concrete staff resources. Concrete persons do not have to be assigned until the time of execution.

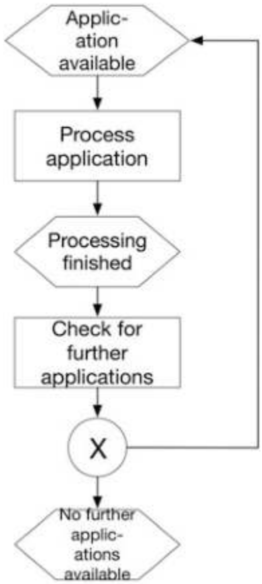


Figure 3.10: EPC with repeated execution of process parts

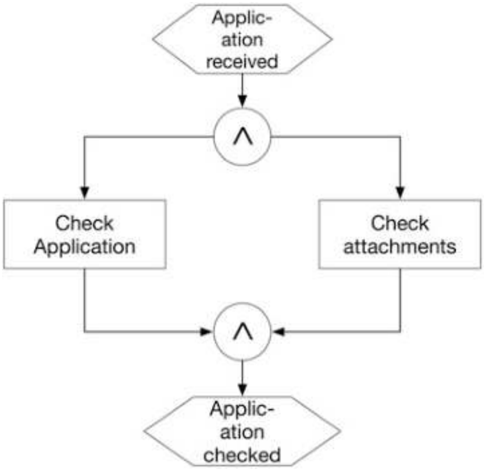


Figure 3.11: EPC with parallel execution of process parts

Undirected connections (lines) are used to show the assignment to a function. In this way, an organizational unit can be assigned to several functions. It is also possible to list organizational units more than once, if the model layout can be made clearer in this way.

IT systems are modeled in a similar way. They indicate the need to use a particular IT system (for example, an ERP system or a database) when executing a function. They are also assigned to functions with undirected connections (lines).

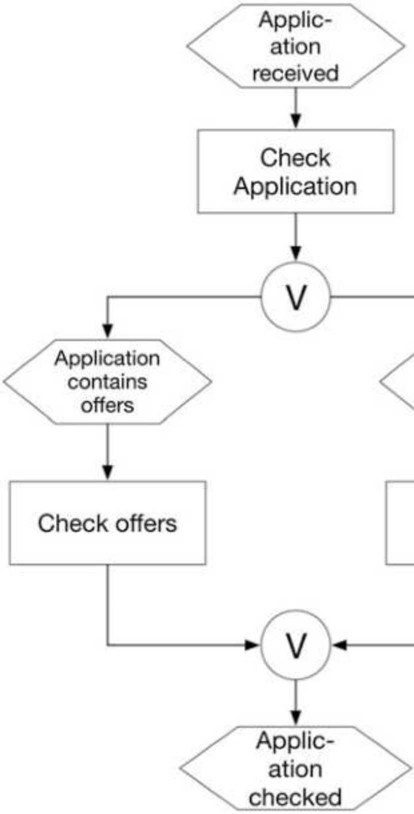


Figure 3.12: EPC with optional parallel execution of process parts

Figure 3.13: Additional notation elements for eEPCs

Information objects are used to represent data processing in a business process. An information object can be arbitrarily comprehensive (i.e., a single value, as well as a complete document) and is always assigned to a function by means of a directional connection (arrow) that describes the data flow. If the arrow ends at the function, this means that the information object is required to execute the function. If the arrow ends at the information object, this means that it is created or changed by the function. An information object can have several inbound and outbound connections, which can describe both its origin and its use in a business process.

3.3.4 Example of an eEPC

To illustrate eEPCs, we use one of the examples described above and add the data flow and the human resources required (cf., Figure 3.14). Application systems would be modeled analogously to the use in organizational units.

With regard to the data flow, we can now see that the actual application must be present in order to check the application. This check not only leads to a positive or negative assessment of the application in the process flow, but also to an information object in which the assessment is stored. In the case of a negative assessment, this information object is required to create the rejection (we can therefore assume that the rejection contains a substantive reason). If the application is confirmed, the "Assessment" data object is no longer required - so we can assume that no further justification will be given in this case.

With regard to responsibilities, we now recognize that several organizational units are involved in the process. While the application is checked by a clerk, the head of department is responsible for the final confirmation or rejection. It is important here that the events on which the decision is based are triggered by the "Check application" function, for which the clerk is responsible. In the present process model, the head of department has no possibility to revise the decision once made.

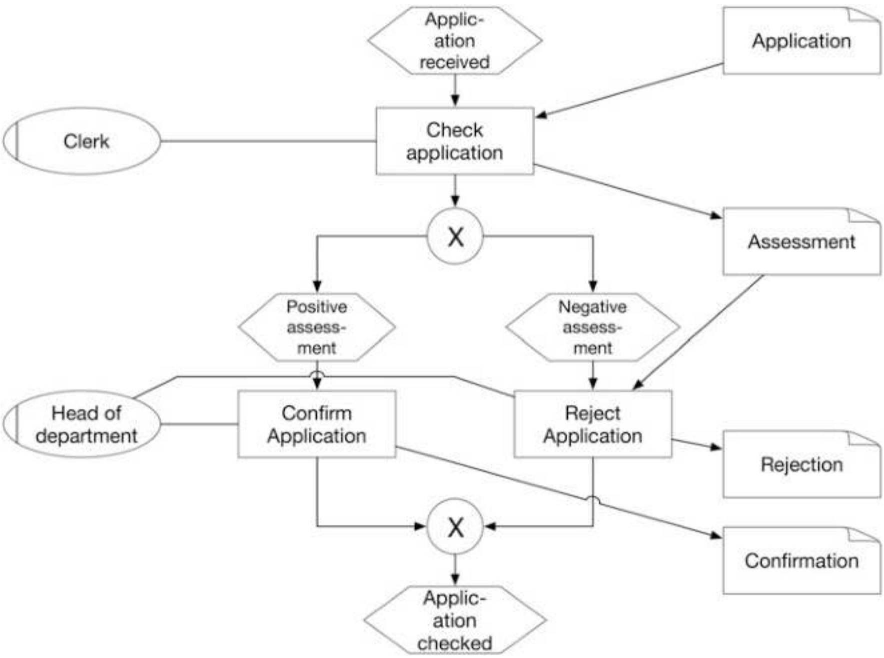


Figure 3.14: Example of an eEPC

3.3.5 Discussion

The EPC offers more comprehensive possibilities for representing business processes than the flowchart. What they have in common is their orientation toward using the tasks and activities within an organization as the primary structuring characteristic of the business process (i.e., all information depicted in the model is anchored in the description of the process flow). This is an obvious choice when describing organizational processes, but - as we will see later - not necessarily the only possibility. Other modeling languages use actors or data as their primary structuring elements on which all other information is anchored. This makes aspects of the process visible that can only be implicitly represented in (e)EPCs (such as the transition between responsibilities in the process and the necessary communication between the actors).

The requirement to alternate functions and events in the process flow in EPCs often leads to very extensive models that are sometimes difficult to understand. It also bears the risk of tempting modelers to formulate trivial events that do not add any information to the model (e.g., function: "Execute task", event: "Task executed"). When used correctly, however, the EPC approach offers advantages: On the one hand, processes can be described and delimited more precisely than with flowcharts; on the other hand, EPCs explicitly allow the view on the capabilities of an organization (its functions) to be linked to the view on how it uses these capabilities to react to external stimuli or events within the organization itself. Thus, organizational capabilities can be described generically and used multiple times in processes, avoiding inefficiency through replication. From a pragmatic point of view, however, practical experience has shown that the specification of generic functions, as well as process-specific events, in the necessary detail is not always feasible. More modern approaches, such as the activity diagrams discussed in the following or BPMN, therefore still use the concept of events, but only deploy these if an external stimulus (such as an incoming message, an error, or a deadline) actually needs to be addressed.

In contrast to the modeling languages with a technically oriented history (such as flowcharts or the activity diagrams described in the next section), the eEPC and the surrounding ARIS framework are concepts originally derived from business administration. They thus pursue a more comprehensive approach to the description of business processes than the technology-centric approaches. The consideration of data, responsibilities, but also goals or services (which were not discussed here), enables a comprehensive modeling of business processes, which still influences the design of contemporary modeling languages for the representation of organizational phenomena (such as business processes or enterprise architectures).

3.4 UML Activity Diagrams

The activity diagram is defined as part of UML (Unified Modeling Language), which contains a collection of modeling diagrams suitable for specifying software systems. The activity diagram there takes a role similar to that of the original flowchart and is used to illustrate the behavior of a software system. Because of its more recent historical context, it also provides elements for the illustration of distributed and parallel process flows. Like the flowchart, the activity diagram is also suitable for representing organizational processes (i.e., business processes). While it is still used for this purpose today, the focus in the area of Business Process Modeling has shifted strongly toward BPMN (Business Process Modeling Notation). BPMN was specified by the same standardization body as UML and has adopted many elements of the activity diagram. However, BPMN focuses more explicitly on the requirements of Business Process Modeling and the organizational aspects to be represented there, which we have already discussed for EPCs.

3.4.1 Notation Elements

By definition, an activity diagram always describes an activity that consists of individual *actions* ("activity" is thus used here analogously to "process"). An action corresponds to an operation for flowcharts, or a function for EPCs, cf., Figure 3.15).

An activity usually begins with a *start node* and finishes with an *end node* (similar to the associated elements for flowcharts). Between these nodes, the actions contained are specified and brought into sequence by control flow arrows. To influence the process, it is possible to insert *decision elements*. Decisions can have any number of outgoing branches, for which the activation conditions must be mutually exclusive. The conditions are listed at the outgoing connections. The semantics of the decision symbol corresponds to the XOR in the EPC - there is no equivalent for the OR connector in activity diagrams.

The activity diagram provides the *split/join element* to represent process parts that can be executed independently of each other and in parallel. When used to split the process, it can have any number of outgoing connections that are all activated at the same time. The branches created in this way should be merged again by the join element. The process is not continued until all branches have been finished.

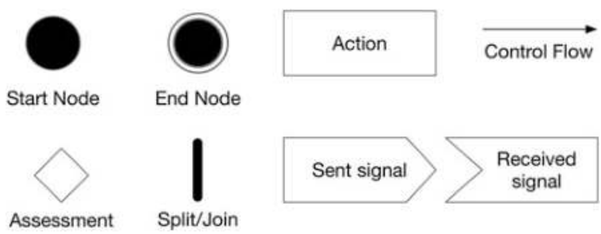


Figure 3.15: Notation elements of activity diagrams

Figure 3.16: Notation elements to represent responsibilities and data flows in activity diagrams

Signals are used for communication between process parts in different activities (i.e., in different diagrams), or within an activity when information is to be provided for subsequent process parts. They are incorporated into the control flow like actions. Models do not always have to contain complete signal pairs (i.e., send and receive signals). They can also send signals for processes that are not shown in the diagram (i.e., contain only a send signal) or receive signals from a process that is not shown in the diagram (i.e., contain only a receive signal). Received signals can also trigger an activity and thus replace the start node in a diagram.

The activity diagram also provides elements to represent responsibilities and data flows (cf., Figure 3.16). Responsibilities are represented using *partitions.* Partitions are elements that enclose parts of an activity diagram and thus determine that all elements, in particular the actions they enclose, fall under the responsibility of the stated organizational unit (or, in the case of software systems, system component). If partitions are used (which is not mandatory), all elements of the activity diagram should be enclosed by exactly one of the partitions shown. Overlaps are not allowed, actions outside a partition should be avoided.

Data objects in activity diagrams are directly incorporated into the control flow between actions. They can therefore (only) be used to represent the flow of information between two consecutive actions. If a data object is only needed again later in the process flow, it has to be forwarded in the control flow via all intermediate actions, or passed on by means of a signal.

3.4.2 Examples

In order to work out the differences and similarities to the previously discussed modeling languages, we again make use of the examples already used above.

The example in Figure 3.17 shows an activity with a single action in which an application (for which we have no further information here) is processed. The activity ends after the processing of the application has been completed.

In the example in Figure 3.18, the process is extended by a decision with three possible outcomes that are mutually exclusive. The application is checked, and the result of this check allows the decision to be taken with regards to the further processing of the application.

As with the other modeling languages, we can also use branches here to repeat parts of a process (cf., Figure 3.19). For this purpose, the decision is inserted at the

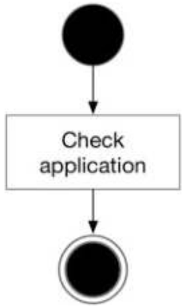


Figure 3.17: Simple acitivity diagram

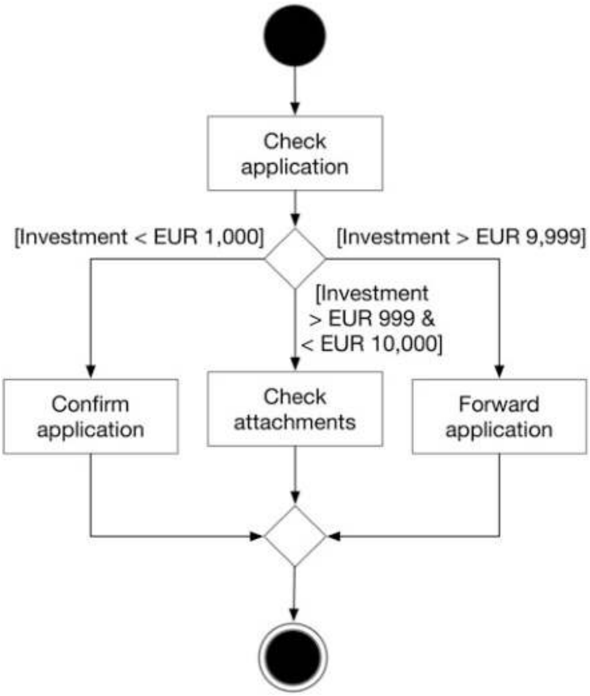


Figure 3.18: Activity diagram with a multiple-outcome decision

end of the part to be repeated and an outgoing branch is linked back to a closing decision before the first operation of the part to be repeated (a closing decision is used for joining alternative branches and has several incoming and only one outgoing connection). The other outgoing branch of the opening decision element continues the process after the repetition is completed.

The split/join element can be used if two action sequences can be executed independently of each other (cf., Figure 3.20). The process modeled in the above example is therefore only correct if the attachments can actually be checked independently of the application. If this is not the case, the two actions would have to be

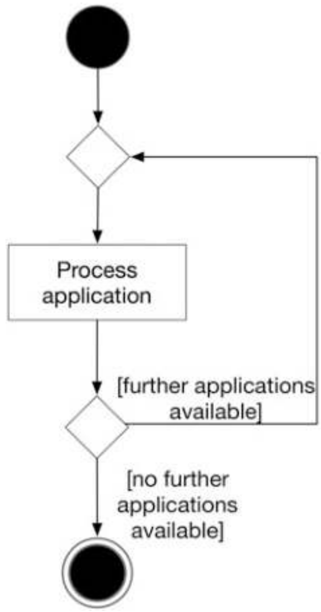


Figure 3.19: Activity diagram with repeated execution of process parts (loop)

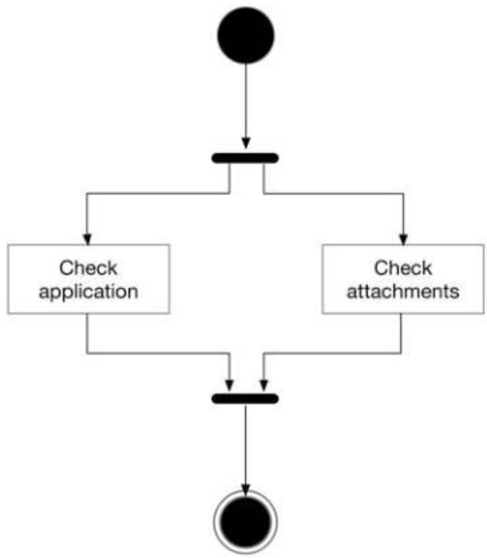


Figure 3.20: Activity diagram with parallel execution of process parts

arranged sequentially. The merging join element waits for both incoming branches to be completed before continuing the process.

The example in Figure 3.21 demonstrates the use of partitions, data objects and signals. Partitions are used to represent the responsibilities in the process. The process is triggered by a received signal, which explicitly shows that the execution

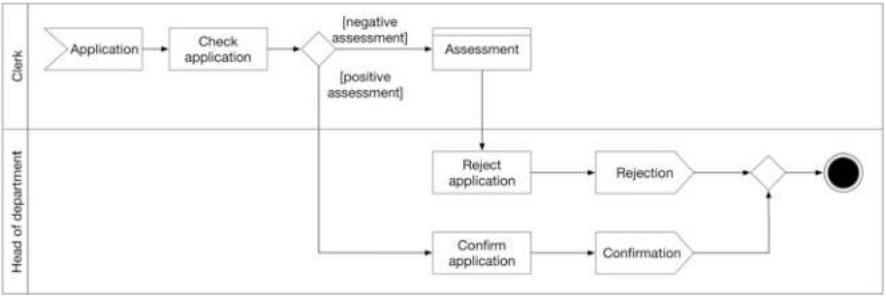


Figure 3.21: Example of an acitivity diagram with partitions, data objects and signals

only starts when an application is received (we had previously always implicitly assumed this for the flowchart and the other examples given for the activity diagram, but, unlike EPCs, were never able to represent it in the model). Send signals are used to transmit the confirmation or rejection to the recipient (who is not shown here). The assessment of the application is only transferred in the case of a negative assessment as a data object for the action "Reject application" (we can therefore assume that the rejection contains a substantive reason). If the application is confirmed, the "Assessment" data object is not required, so we can assume that no further justification will be given in this case.

3.4.3 Classification

Activity diagrams largely combine the simplicity of flowchart notation with the expressiveness of EPCs (with certain limitations). They allow the handling of data in the process to be represented and introduce a means of clearly representing responsibilities with the partitions. In contrast to the previously discussed languages, the availability of signals allows the depiction of communication processes between participants or with the surrounding environment of the displayed process.

The absence of an element corresponding to the OR connector in the EPC does indeed represent a limitation, however this is rarely relevant in practice, since we are usually confronted with mutually exclusive alternatives or completely independent branches of execution in the real world. Activity diagrams are therefore a suitable tool for representing business processes on the whole, especially if the target group using the model has an Information Technology background and is already familiar with the notation. For other target groups, BPMN, which we will discuss in the next section, is preferable due to its more flexible applicability and higher expressiveness for Business Process Modeling.

3.5 BPMN

BPMN – the modeling language referred to as Business Process Modeling (and) Notation - was developed by IBM in 2002 and subsequently published by the BPMI (Business Process Management Initiative). The aim was to create a universally applicable standard to counter the multitude of process modeling languages used in academia and industry. This language should adopt the essential characteristics of the most common languages and make it possible, in addition to the documentation of business processes, to create models that allow for immediate IT-supported execution. BPMI in turn was merged with the OMG (Object Management Group) in 2005. Thus, BPMN became an OMG standard and complements the already mentioned UML (Unified Modeling Language), which is maintained by the same group.

The BPMN 2.0 standard was published in 2010. This standard incorporates several new diagram types: the choreography diagram, the conversation diagram, and the collaboration diagram. In the following, we consider the basic elements of BPMN 2.0 that enable business processes to be represented at the business level.

BPMN focusses on business processes, which it presents as a temporally logical sequence of activities (tasks) which are structured in accordance with organizational responsibilities. The representation of data is not as comprehensive as in other modeling languages and only regarded in the context of process flows.

3.5.1 Notation Elements for Modeling Process Flows

Process diagrams created with BPMN are called Business Process Diagrams (BPD). In its core BPD follows the principles of activity diagrams, which are subsequently supplemented by elements that allow the representation of the potentially more complex control flow in business processes (cf., Figure 3.22).

As a basic principle of representation, certain things have to be done in a process (*tasks*), but possibly only under certain conditions (*gateways*), and things can happen

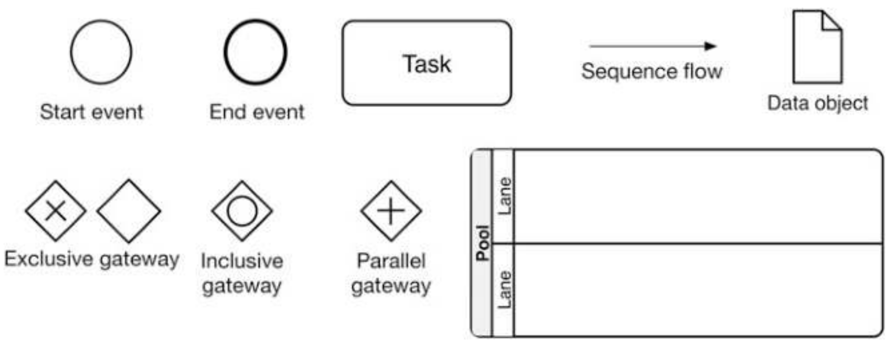


Figure 3.22: Core notation elements of BPMN

(*events*). These three objects are connected to each other via *sequence flows, however only within* the confinements of *pools* or *lanes*. Pools and lanes are constructs used to represent responsibilities in distributed business processes. They are discussed in more detail below. If a connection is made across pool boundaries, it is modeled using message flows, which we will expand on later.

A process consists of tasks. After starting a process (by means of an event), one task follows the other until the process ends (with an event). Tasks can be atomic (i.e., not refined further) or can contain sub-processes. In such cases, tasks are refined by an additional embedded BPD, which represents its detailed sequence of sub-tasks. This detailed sequence can be "hidden" and is represented by a "+" symbol at the bottom of the task.

A process begins with a *start event* and ends with an *end event*. BPMN offers a multitude of possibilities to define events that can trigger, complete or influence the course of a process. These will be discussed later.

At this point it is important to emphasize that a process can start with one or more start events and can end on any path through the process (see sequence flow and gateways below) with one or more end events. There must be a continuous sequence flow from each start event to at least one end event. Tasks, gateways or intermediate events must not be endpoints in the process and therefore always require at least one outgoing sequence flow.

A gateway represents a branch in the control flow. The *exclusive (XOR) gateway* requires a condition for each outgoing control flow, which according to the standard must always refer to the result of an immediately preceding task.

The *parallel (AND) gateway* tracks all outgoing control flows independently and in parallel. The branched control flows can be terminated separately with end events or explicitly merged again with another parallel gateway. After this merger, the control flow only continues once all incoming control flows have been completed (as with the split/join concept for activity diagrams).

The *inclusive (OR) gateway* can follow one or more paths, whereby a condition must be specified for path selection (as with the exclusive gateway). This condition must already be testable at the time of the decision, so the necessary data must have been generated in one of the previous tasks.

Decisions, which cannot be made on the basis of previously existing data, can be represented using the *event-based gateway*. This requires an event in each outgoing branch immediately after the gateway (e.g., an incoming message event or a timer event). When one of these events occurs, its respective branch (and only this branch) is activated. We will discuss this in more detail when discussing the use of events.

3.5.2 Examples for Modeling Process Flows

In order to work out the differences and similarities to the previously discussed modeling languages, we again make use of the examples already used above.

Figure 3.23: Simple BPMN diagram

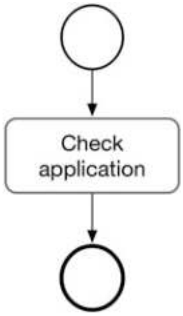
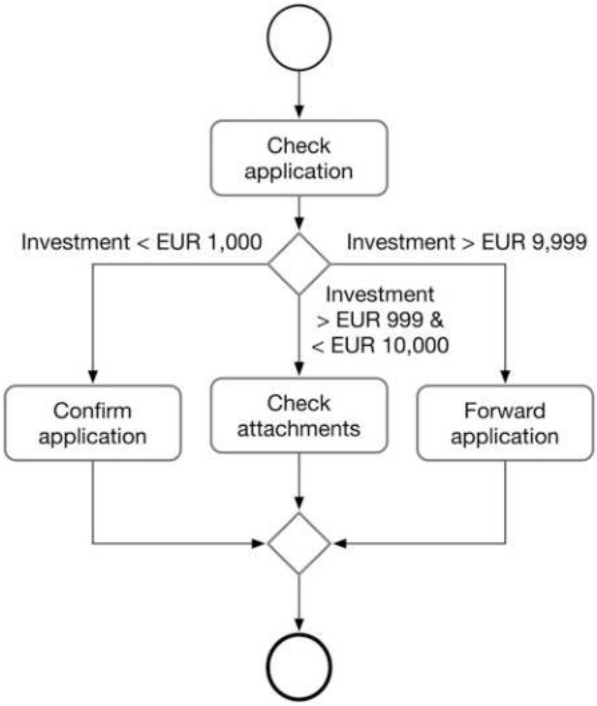


Figure 3.24: BPMN diagram with multiple-outcome decision



The example in Figure 3.23 shows a process with a single task in which an application (for which we have no detailed information here) is checked. The process ends after the checking of the application has been completed.

In the example in Figure 3.24, the process is extended by a decision with three possible results that are mutually exclusive. The application is checked, and the result of this check allows the making of a decision on how to further process the application. In BPMN, it is important that the data used as the basis for a decision is explicitly generated or received before the gateway.

Here, too, we can use branches to repeat parts of a process (cf., Figure 3.25). For this purpose, a gateway is inserted at the end of the part to be repeated and an

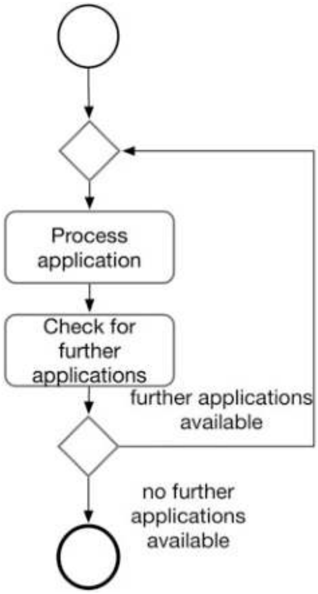


Figure 3.25: BPMN diagram with repeated execution of process parts (loop)

outgoing branch is returned to a closing gateway before the first task of the part to be repeated (a closing gateway is used for merging branches and has several incoming sequence flows and only one outgoing). The other outgoing branch of the opening gateway continues the process after the repetition is completed. The requirement to explicitly create the basis for decision-making prior to the gateway is illustrated here by the additional task of checking the existence of further applications.

The parallel gateway is used when two task sequences can be executed independently of each other (cf., Figure 3.26). The process modeled in the example therefore is only correct if the attachments can actually be checked independently of the application. If this is not the case, the two functions would have to be arranged sequentially. The merging gateway waits for both incoming branches to complete before continuing the process.

The example in Figure 3.27 shows the use of pools, lanes and data objects. The lanes are used to represent responsibilities in the process. With the BPMN elements introduced so far, we cannot map communication processes. The signals available in activity diagrams can therefore not be represented for the time being, which is why the example here is less specific than the version represented as an activity diagram. The use of message events, which we will introduce in a later chapter, will, however, remedy this deficiency. The assessment of the application is only transferred to the task "Reject application" in the case of a negative assessment as a data object (we can therefore assume that the rejection contains a substantive reason). In case the application is confirmed, the data object "Assessment" is no longer needed, so we can assume that in this case there will be no further justification.

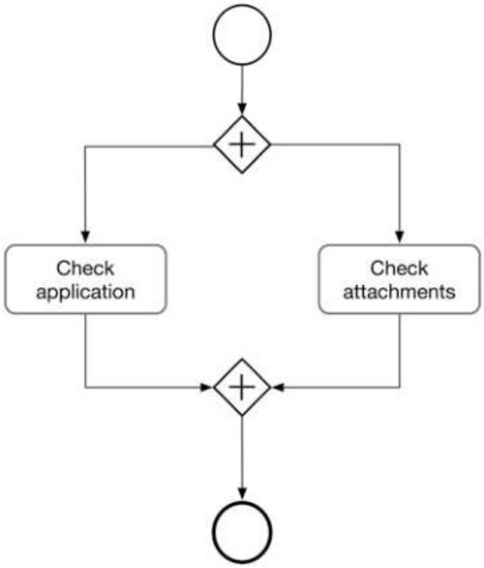


Figure 3.26: BPMN diagram with parallel execution of process parts

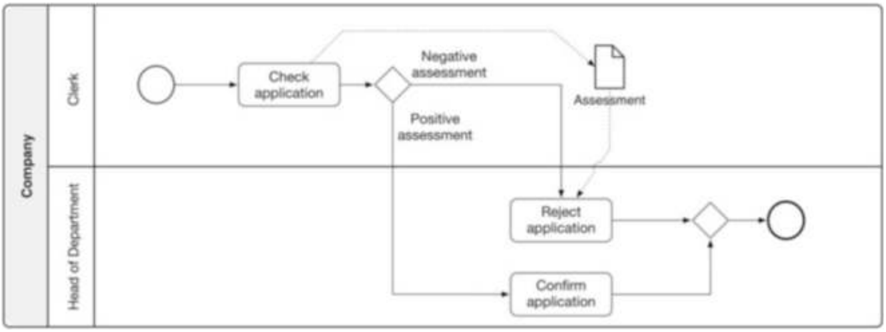


Figure 3.27: Example of BPMN diagram with pools, lanes and data objects

3.5.3 Notation Elements for Controlling Sequence Flow with Events

A distinguishing feature of BPMN is the very detailed and comprehensive set of event constructs, which enables exact control of the process flow.

Events indicate that something has happened and therefore represent points in times, as opposed to tasks that take a certain amount of time and effort to be completed. So far, we have only introduced start and end events. In the following we will describe start, intermediate and end events in detail once again (cf., Figure 3.28).

Events are always represented by a circle and usually an enclosed symbol. Simple circles indicate start events, double circle borders indicate intermediate events and thick circle outlines indicate end events. If no enclosed symbol is specified, the event

Figure 3.28: Examples of different event types

Figure 3.29: Basic BPMN triggers of start events

Figure 3.30: Examples of combined or complex BPMN triggers for start events

is of no particular type (blank) and can usually only be found at the start or end of a process or as a triggering intermediate event.

Start Events

Start events are used to trigger a process or subprocess. Generic (blank) events are often used if the trigger is either clear from the context, or when it is not yet known.

If a process is triggered on the basis of a particular point in time or a period of time or by a periodic event, the timer symbol (clock) is additionally used. A message symbol (letter) is used when an incoming message triggers the process (cf., Figure 3.29).

A process can also be triggered by a condition start event, if specific conditions need to be met in the process context in order to start it. Signal start events are used if some observable event from outside or inside the process should lead to the execution of a particular process. The pentagon as a symbol embedded in a start event indicates a combination of several potential start events, whereby only one of the events must occur to trigger the process (cf., Figure 3.30).

A process does not have to be limited to a single start event; processes can also have several alternative start events.

End Events

Processes always finish with end events, whereas the same symbols are used here as for the start events, with the exception of the timer symbol, the condition and the combination of parallel triggers. More specific end events include sending a message or issuing a signal to notify other processes about the end event at hand (for details, cf., section “Event Types”).

One notable end event type is the termination end event (rightmost symbol in Figure 3.31), which terminates the entire process immediately, regardless of whether

Figure 3.31: Examples of BPMN end events

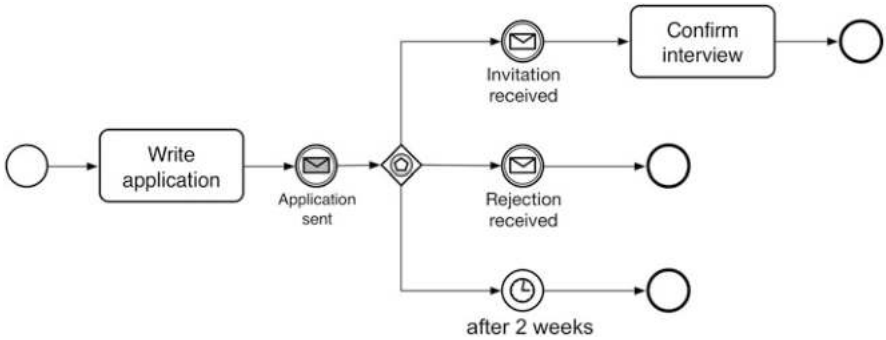


Figure 3.32: Example BPMN diagram showing the use of the event-based gateway

other sequence flows within the process are still running or not; that is, it terminates the entire process instance. A standard end event on the other hand always only terminates the process branch it concludes. Any further process branches that are still running will continue to be executed.

Processes can have several end events. A process without an end event is incomplete.

Intermediate Events & the Event-Based Gateway

Intermediate events can be used anywhere in a process and are represented by a circle with a double border. They are used to represent intermediate results relevant for other processes, or if certain (external) events have an impact on the execution of the process at hand (for example, an incoming message or the expiration of a certain time period).

Gateways can also be event-based if the execution of a process is dependent on the occurrence of different events, which require different reactions and therefore follow different paths. Such dependencies can be modeled with an event-based gateway.

Figure 3.32 shows a process of applying for a job and waiting for different potential reactions. Depending on whether an invitation or a rejection is received, or whether a deadline of 2 weeks expires, different paths are taken in the process. The event-based gateway is the only gateway where the necessary information required to make a decision does not need to be available at the time the gateway is checked. An event-based gateway blocks the sequence flow until one of the downstream events is triggered, and then continues execution exclusively along the respective branch.

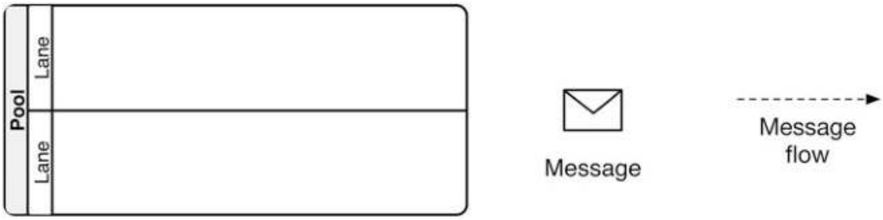


Figure 3.33: BPMN notation elements for modeling communication

3.5.4 Notation Elements for Modeling Communication

BPMN also enables the modeling of distributed business processes. Although BPMN clearly focuses on the process flow during modeling (similar to flowcharts, EPCs or activity diagrams), it also enables the structuring of the process in accordance with the participants involved and their associated responsibilities. The modeling elements available for this purpose are described in this section (cf., Figure 3.33).

A *pool* represents a company or an organizational unit in a company, such as a department. Each (swim) *lane* in a pool represents a person or role involved in the process that is assigned to this pool.

BPMN allows the representation of the interaction of two or more processes. The aforementioned pools and lanes are necessary for the representation of collaborations. Separate lanes are required for all persons or groups involved in a process, and separate pools are necessary for each process or organizational unit that is responsible for this process. Each pool thus contains its distinct processes with separate start and end events. Nevertheless, these individual processes can strongly influence each other, in which case they are coupled via message flows.

Message flows indicate that data is exchanged between different processes. Therefore, no message flow can take place within a process (pool). Consequently, there are no message flows within a lane or between different lanes of a single pool. Sequence flows show which activities are executed in which order and do not explicitly constitute an exchange of data. In contrast to message flows, they may only be used within a pool and not between different processes (pools).

Message flows can be augmented with message elements, which are used to explicitly represent the exchanged data and contain a more precise specification of the transmitted information.

Message flows can be used in different ways: they may either originate from pools and activities, and also end there, or they can be explicitly sent by send message events and received by receive message events. The first case is useful for the descriptive modeling of business processes in which a communication process is to be represented that does not necessarily have to be described exactly. A message originating from an activity or pool is sent at some point during task or process execution - the exact time remains unclear. A message ending at a pool only states that the represented organization receives this message, but not which activity

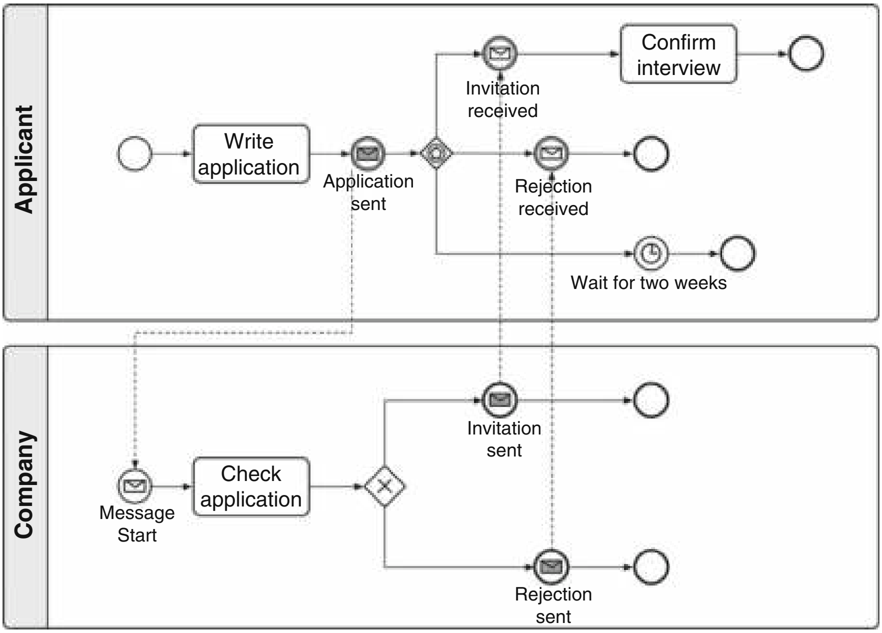


Figure 3.34: Example BPMN diagram showing communication-oriented processes

it triggers or how it is handled within a process. This can be useful when modeling external organizational units, whose detailed behavior is unknown. An exact specification of communication processes, however, is only possible by using explicit send and receive events.

3.5.5 Examples for Modeling Communication-Oriented Processes

In the following, we extend the example that we used to demonstrate the use of events with the communication partner not depicted originally, i.e., we now also model the process of the company to which an application is addressed.

The example in Figure 3.34 shows two processes (one per pool) that are linked by message flows. The company's process is triggered by an incoming application, which is represented here in the first message flow. After checking the application, the decision can be made whether to send an invitation or reject the application. In the upper pool, the applicant waits for an answer for a maximum of 2 weeks (as represented by the intermediate timer event). The event-based gateway activates the process branch whose event occurs first. The related send and receive events are linked via message flows. It is important to note here that message flows always represent 1:1 relationships – that is, a sent message can be received exactly once and a receive event can react to exactly one message.

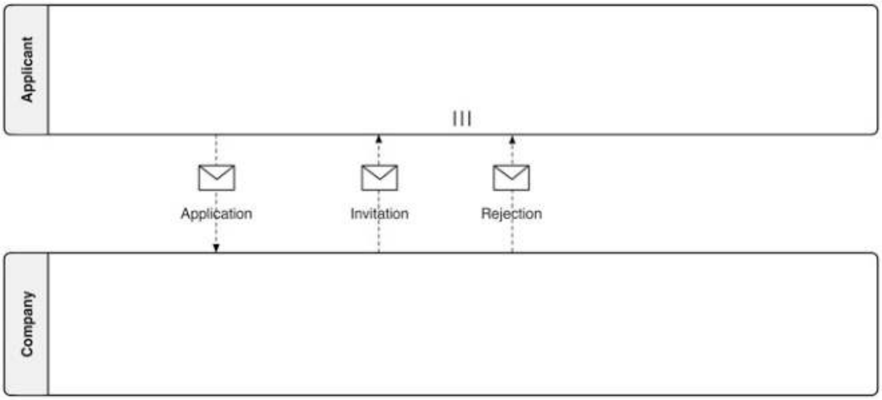


Figure 3.35: Example BPMN diagram showing the exclusive use of communication interactions

The example in Figure 3.35 shows how BPMN can be used to only represent communication acts between the actors in a distributed process. The pools are used here as black boxes, i.e., the behavior contained in them is not shown and remains unknown. All we see is that messages are exchanged, but the order of the messages is not defined. Since the specified events for sending and receiving are missing here, we augment the model with message elements attached to the message flows in order to be able to comprehensibly describe the nature of communication. Another extension of the original process model is constituted by the modifier in the upper pool, which indicates parallel multiple execution of the process contained in the upper pool. This means that the process in the lower pool could or even must be able to handle several applications arriving in parallel and independently of each other.

Empty and filled pools can also be combined as required. If, for example, we wanted to represent the process of handling an incoming application, we could leave the pool "Applicant" unspecific, since we do not know the behavior of applicants (nor is it relevant), but need to know that we can receive an application from them and that we will direct our responses to them again eventually.

3.5.6 Notation Elements for Modeling Complex Business Situations

The notation elements of BPMN introduced so far enable the representation of business processes from the point of view of the participating organizational units. BPMN allows keeping process models vague or leaving parts of them unspecified if they do not seem relevant for the objective of modeling. In some cases, however, a process needs to be defined as precisely as possible and represented in all its variants, covering all possible exceptions. This is necessary, for example, if the model is intended to serve as the basis for IT-support of the work processes depicted. If aspects are omitted or abridged, the result is a discrepancy between the real work process and the support measures developed based on the model, which ultimately

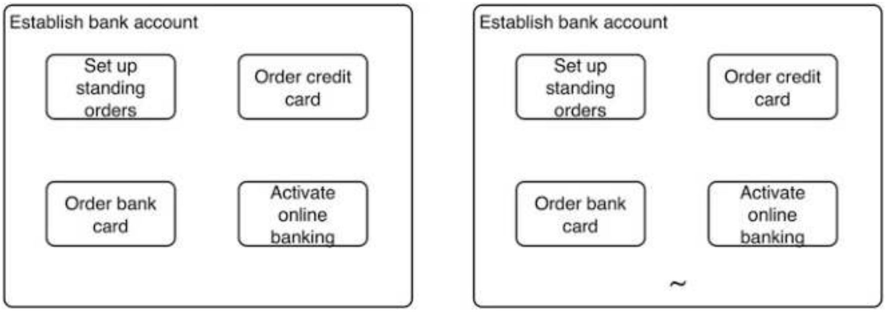


Figure 3.36: Parallel and ad-hoc subprocesses

would lead to unsatisfactory tools and work-arounds. This section describes the BPMN notation elements that enable more complex and comprehensive process descriptions. Due to the variety of scenarios that can be represented, examples are given here directly with the descriptions of the respective elements.

Variants of Activity Modeling

In the following section, special features for the general modeling of activities, as well as for the modeling of activities as subprocesses are explained in more detail.

Subprocesses

Processes can include detailed specification of tasks via subprocesses. This method is mostly used to maintain a comprehensive overview of a process when creating large, extensive models, while still being able to specify detailed task descriptions. Subprocesses can be collapsed to tasks, which are then shown in the overall process with a small plus sign. If appropriate tool support is available, collapsed tasks can be dynamically extended again to view the detailed subprocess specifying the exact execution of the task.

Subprocesses also can be used to combine several tasks in a single execution context without specifying their exact sequence. The model on the right (an ad hoc subprocess) in Figure 3.36 only indicates that any number of the embedded activities can be performed but makes no statement about their relationships. The left model specifies that all four embedded tasks must be executed before the subprocess is completed. It makes no assertion about their order or other relationships - the activities can be carried out in parallel or in any sequence.

Types of Tasks

Task types describe the character of a task in more detail, indicating for example whether it requires human involvement or can be executed automatically in an IT system. Modifiers as shown in Figure 3.37 are used to distinguish between service tasks, receive tasks, send tasks, user tasks, business rule tasks, script tasks, and manual tasks (illustration from top left to bottom right). These modifiers do not

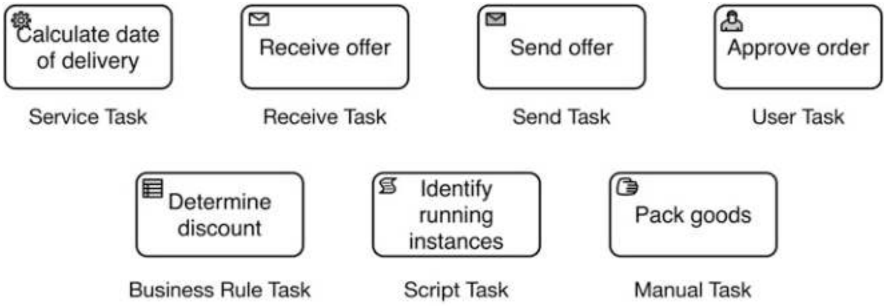


Figure 3.37: Different task types

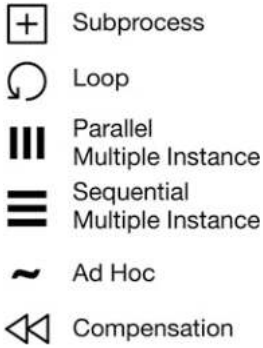


Figure 3.38: Task modifiers for diverse behavior specification

necessarily have to be used, but they do specify the semantics of a process model in more detail.

Execution Behavior of Tasks

Tasks can have different markers that describe their execution behavior. Single tasks, entire processes or subprocesses can be executed several times in loops, in parallel, sequentially, or can be marked as ad hoc tasks or as compensation tasks (cf., Figure 3.38).

For a looped task or process, a termination condition can be specified in addition to the symbol. If the termination condition is reached, the task or process in question is no longer executed and the superordinate process is continued. If a task can be executed several times in parallel, this is indicated by three vertical lines. For example, the "check application" task could be carried out by several agents for several received application documents. If parallel execution is not possible, but the individual cases are still independent of each other, the sequential multiple instance marker is used, which is indicated by three horizontal lines.

With ad hoc tasks, the exact sequence of the sub-tasks contained in the task is unknown a priori and is selected during the execution of the process. It is also

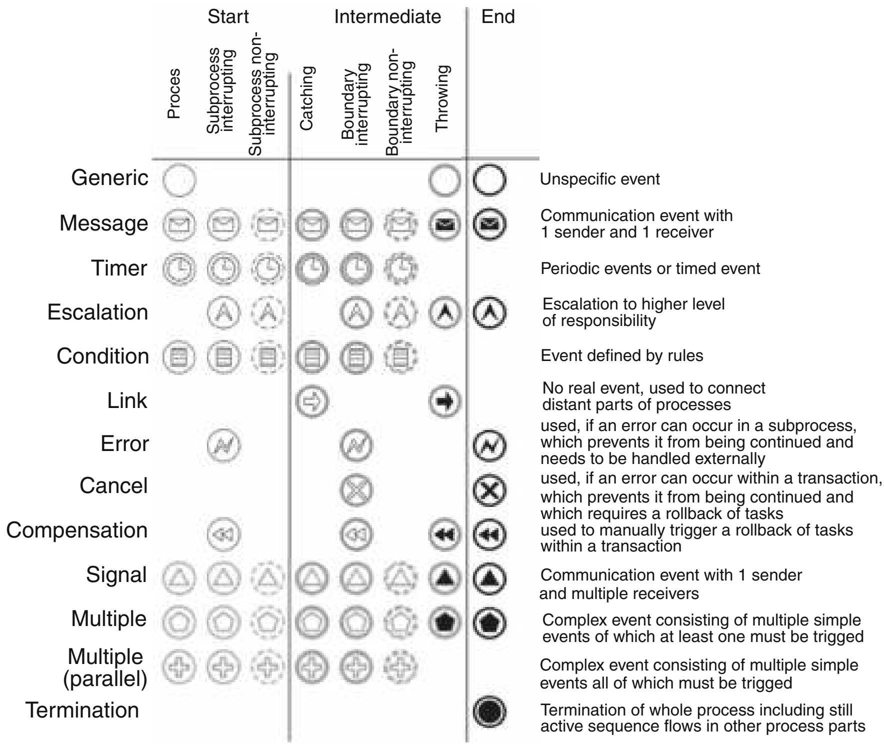


Figure 3.39: Different BPMN event types

possible to omit some of the sub-tasks and only execute those that are required in the specific situation. Such processes are indicated by a tilde (as shown above).

Compensation activities are used in transaction modeling and are described below.

Event Types

BPMN offers a large number of different events targeting different areas of application for detailed process control. While we will deal with these in detail in the following, an initial overview is provided in Figure 3.39 to show the underlying structure of event types.

In general, events can occur in three different variants, which we have already introduced above:

– *Start events trigger* new process instances, that is, they start the execution of a process. They are always "receiving" in nature, i.e., they react to stimuli from outside (e.g., incoming messages, time sequences, etc.).

– *End events* are events that are triggered when a process instance is finished. They are always "sending" in nature, thus potentially provide stimuli for other processes or process parts.
– *Intermediate events* occur within the sequence flow, i.e., they have both an incoming and an outgoing flow (exception: link event, see below).

Start events can also be differentiated according to their surrounding modeling element. They can be used to either trigger an entire process or to trigger subprocesses. The second case is called an "event subprocess" and can be specified as an "interrupting" or a "non-interrupting" form. An "interrupting" start event indicates that the control flow is completely transferred to the subprocess, i.e., all other task within the respective pool are interrupted and cannot be continued. "Non-interrupting" event subprocesses are started when the respective event occurs without interrupting the execution of the task currently running within the pool the subprocess is placed in. This can be used, for example, to react to events that should not or cannot be handled in the main process of a pool, but whose occurrence should result in a reaction without affecting the main process (such as customer inquiries about the status of an order processing while the order is being processed in the main process).

Intermediate events basically exist in "receiving" and "sending" variants. The "receiving" variants (occurring event) block the sequence flow until the specified event arrives. The process therefore cannot be continued until the event has occurred. The "sending" variant (triggered event) indicates the occurrence of certain events in the course of executing a process (or also an occurrence between processes from different pools). Events often are used reciprocally in comprehensively modeled processes, i.e., a receive event exists for each send event.

Receiving intermediate events also exist in a "*boundary*" form. These events are "pinned" to tasks (i.e., are graphically attached to the (lower) boundary of the task) and indicate that it is possible to react on the respective event during the execution of the task. The reaction is specified by a sequence flow originating from the attached event, which leads to the respective tasks to be performed. The boundary intermediate events in general (with some exceptions) again exist in an interrupting and a non-interrupting form. The interrupting form stops the execution of the task marked in this way and continues the sequence flow exclusively via the attached event. The non-interrupting form allows the further execution of the task marked in this way, and the sequence flow originating from the event is triggered in parallel.

The triggers that lead to the occurrence of boundary events can come from outside the tasks (for example, incoming messages from other pools) or also from within the task, provided that these triggers are detailed by a subprocess. For example, an error in the execution of a subprocess can lead to activities in the main process via an interrupting error boundary event (such as documenting the error and escalating it to superiors).

We have already used message and timer events in basic BPMN modeling. We will now consider the other event types in the context of their respective application areas.

Figure 3.40: Link event

The Link Event

For more complex or extensive processes, tracking sequence flows through the diagram can sometimes be difficult. Sequence flows crossing each other or sequence flows with many changes of direction are difficult to read and are detrimental to comprehensibility and clarity.

In such cases, the link event can be used (cf., Figure 3.40). In contrast to the other events, it semantically does not represent a real event, but merely serves as a connector between two sequence flows that are far apart. The coupling is carried out via the designation of the sending and the receiving event. There must always be a 1:1 assignment (implicit parallelization by one triggering and several receiving link events of the same name is therefore not permitted).

Link events should only be used in cases that cannot be resolved in any other way in order to increase clarity, since the effort involved in searching for related link events can even exceed the tracking of complex sequence flows (as long as there is no tool support for jumping to or visually marking related events). Choosing an alternative arrangement of activities or lanes is usually the better choice.

Use of Signals

In BPMN, messages can only be used for communication between pools. In addition, message-based communication always has exactly two endpoints, so it can only connect exactly one sender to exactly one receiver at a time. If information is to be made available globally within a collaborative process and this is to happen independently of pool boundaries, signals can be used. Signals can be triggered in a process (as intermediate or end events) and are then available both within the pool and in all other pools of the same collaboration.

Signals can be used, for instance, to inform all pools of a collaboration about the termination of one of the represented processes. This means that all other processes that are still running can complete their processes cleanly and there are no dangling processes left that can no longer be completed, e.g., because an expected incoming message no longer arrives due to an aborted process of a communication partner.

Handling of Exceptions and Interruptions

Activities, i.e., tasks and subprocesses, can be aborted or interrupted by certain events. This is indicated by event symbols attached to the respective task. Two solid outer circular lines in the event element indicate that the task is interrupted by the event; two dashed circular lines indicate that the task is not interrupted but can be continued while simultaneously reacting on the exception that occurred. In the example in Figure 3.41, we react on a deadline, i.e., we model that the execution

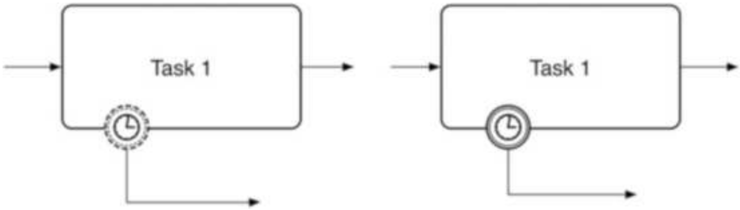


Figure 3.41: Non-interrupting and interrupting boundary events

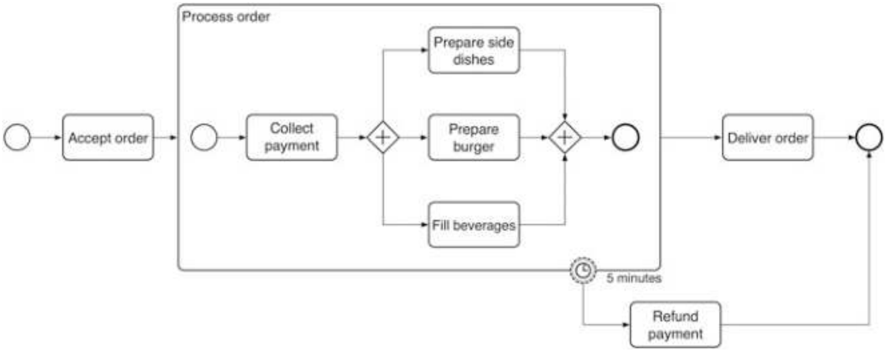


Figure 3.42: Example for use of a non-interrupting timer event

of the task must not take longer than a certain time span (on the left) or requires some reaction if it lasts longer than a certain time span (on the right). In either case, the model must contain information on what has to happen when task execution takes longer than anticipated. This information is modeled as a sequence flow emanating from the attached boundary event. Reactions to unforeseen triggers such as errors or escalations can also be modeled and displayed in the same way using the respective event type.

Example: Non-interrupting Timer Events

Figure 3.42 shows a subprocess that includes the activities of processing an order in a fast food restaurant, which should not take longer than 5 minutes. If the processing of the order takes longer than 5 minutes, the customer should receive their money back. The order should still be completely processed. If we had used an interrupting boundary event here, the customer would only get their money back, but not their order.

Different Ways of Terminating Processes

Sequence flows in a process are usually concluded with an end event. Such end events, however, only terminate the execution of the respective sequence flow. If other sequence flows are active in parallel (for example, because they were opened by a parallel or inclusive gateway or because they were triggered by boundary events), their execution will continue to be carried out. There are several ways to terminate a (sub)process completely and immediately (i.e., terminate execution in all branches).

The Terminate Event

The terminate event aborts all active branches of a process within a pool immediately. Processes in other pools are not affected and should therefore be informed of the termination by sending a signal before the termination, if necessary (e.g., if there is the risk of waiting for further input from an already terminated process instance).

The Error Event and the Escalation Event

The error event semantically indicates the occurrence of an unforeseen error in process execution and is usually used for subprocesses. It immediately terminates the execution of the whole subprocess. The reason for the error can be given to the event as a parameter. Receiving error events can be attached to the subprocess as a boundary event for the enclosing task element to react to these errors in the superordinate process and trigger corresponding activities. Attached error events are always interrupting, i.e., they terminate the execution of the subprocess (including all active sequence flows in branches in which no error occurred). As a "weaker" variant, the "escalation event" can also be used in an identical way. The escalation event also is available in a non-interrupting form and thus allows the continuation of the execution of the subprocess in which the problem occurred.

If the effects of activities already performed have to be reversed when subprocesses are terminated, the transaction handling mechanism and constructs provided in BPMN can be used. They are described in the next section.

Transactions

BPMN also offers the option of representing transactions in a process. A transaction is a set of tasks that is to be executed as a whole, either completely or not at all. In particular, if a task fails, the effects of other already completed tasks need to be reversed. BPMN introduces the concept of transactional subprocesses in combination with compensation events and tasks. Compensation tasks roll back the effects of process steps that have already been executed by means of countermeasures which are initiated in a further process step.

In a transaction subprocess (characterized in BPMN by a double border of the enclosing task element), each task is assigned a compensation task via a boundary compensation intermediate event (indicated by a "rewind" symbol). If the transaction is aborted or should explicitly be undone retroactively, the respective compensation task is executed for each task that has already been successfully completed. The abort end event (marked by an "X") can be used to abort a transaction while it is still being carried out. As an end event in a transaction subprocess, it causes its immediate termination and triggers the compensation tasks. When attached to the transaction task as a receiving intermediate boundary event, it determines the further course of the process after the transaction is terminated. As a result of the concept of

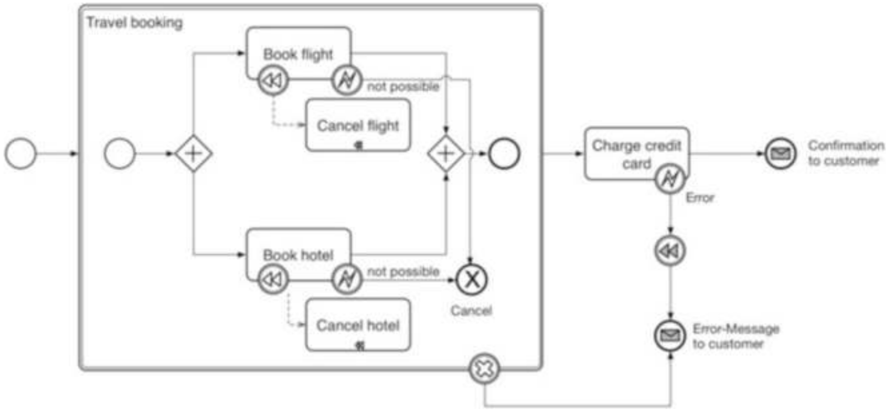


Figure 3.43: Exampe for transaction subprocesses and compensation events

compensation, transactions can also be rolled back after they have been successfully completed. Outside the transaction, a sending intermediate compensation event can be used to retrospectively trigger the compensation tasks contained in the referenced transaction.

Figure 3.43 illustrates these concepts using a travel booking process.

A travel booking consists of a flight booking and a hotel booking that can potentially be done in parallel. If one of the bookings is not possible, the other booking must be canceled, if it has already been done. An error in one of the bookings leads to the transaction being canceled (triggering the termination event) and leads to sending an error message to the customer. Outside the actual transaction, an error in charging the credit card leads to cancellation of the entire booking by triggering the compensation tasks retrospectively.

Event-Triggered Subprocesses

Event-triggered subprocesses are an alternative to boundary events when handling non-standard incidents that might occur in (sub)processes. While boundary events attached to subprocesses lead to the reaction to such incidents in the superordinate process, strictly local reactions (i.e., reactions that do not have any implications for the overall process) can be kept in the context of the subprocess by using event-driven subprocesses.

Figure 3.44 shows an example of a timer-controlled non-interrupting subprocess. It picks up on the scenario already used above to demonstrate non-interrupting boundary events and shows the process of preparing an order in a fast food restaurant.

Event-triggered subprocesses can be started with the same types of events that are available as boundary events, both in their interrupting and non-interrupting versions. Semantically, as already mentioned above, they differ only in the way the incident triggering the event is handled - locally within the subprocess or externally within the superordinate process. Depending on the process, one or the

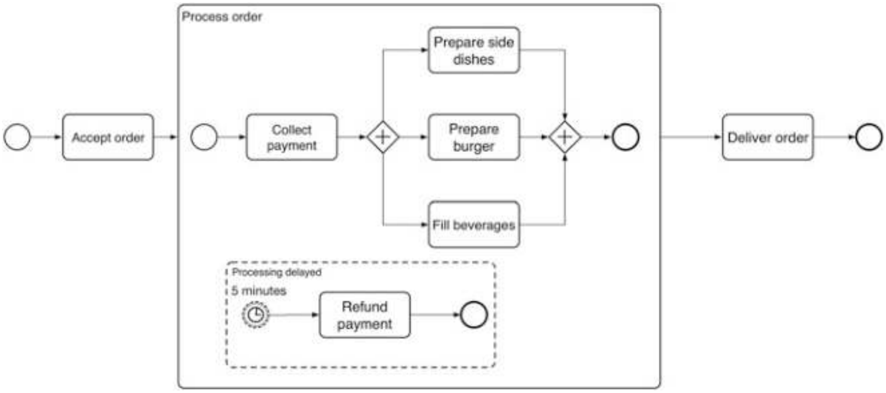


Figure 3.44: Example for non-interrupting event-triggered subprocesses

other variant can lead to a more meaningful and/or comprehensible form of representation.

3.5.7 Choreography Diagrams

The ability to explicitly model the interplay of actors in a collaborative process has been introduced in BPMN 2.0 in the form of choreography diagrams. A choreography depicts the process of exchanging messages between different actors. It thus provides a different view on a collaboration, focusing on the sequence of the transmitted messages independently of the processes of the individual actors.

Although a representation of communication is possible in BPMN process diagrams by means of collapsed pools and the messages exchanged, the exact sequence, conditional message flows or loops cannot be represented in this way. For instance, the example of an application process used to demonstrate the use of collapsed pools as shown in Figure 3.35 does not include information on whether the invitation message and the rejection message are mutually exclusive or can occur in parallel. This can be visualized with a choreography diagram.

Figure 3.45 shows the choreography representation of the application process shown above as a collaboration process. Here, the process of message exchange is in the focus of representation. Choreography tasks represent the exchange of one or more messages between two or more partners. In their simplest case they correspond to sending a single message from one partner to another.

Each choreography task is triggered by one of the partners involved by sending the first message. This triggering partner is entered in a box with a light-colored background at the upper or lower edge of the choreography activity. The names of the other party or parties involved are entered into boxes with darker backgrounds on the other border of the task. Which partner is entered at the top and at the bottom is at the discretion of the modeler. Usually, if there are several choreography tasks

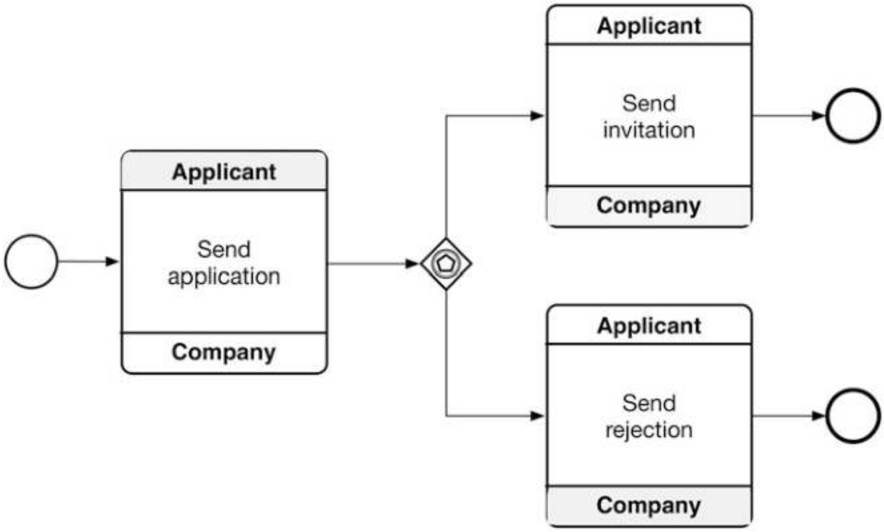


Figure 3.45: Example of a choreography diagram

between the same partners, the arrangement will remain identical to allow for better comprehensibility. If corresponding collaboration diagrams are modeled, it is recommendable to use the vertical arrangement of the pools as a basis for labeling the partners in the choreography tasks.

Choreography activities with more than two partners do not occur in the example shown above. In case more than two partners would be involved, several partner fields can be added at the top or bottom. However, only one field can have a light-colored background, since only one of the partners initiates message exchange with an initial message.

A sequence flow is defined in the choreography diagram between the choreography tasks. Modeling the sequence flow in choreography diagrams essentially corresponds to the sequence flow modeling of ordinary BPMN processes. However, certain elements of process modeling do not make sense in connection with choreography modeling and are therefore not permitted. For example, there are no message events within a sequence flow, since the message exchange is, by definition, part of the choreography tasks. Accordingly, in the diagram above, event-based gateways are not followed by events, but rather by choreography tasks. The path is selected for which the associated choreography task is first started by the respective triggering message.

If one wants to know which messages are exchanged in each choreography task, these can be added to the diagram in the form of letter symbols which are linked to the respective partner sending the message. The letters are color coded in the same way as partner fields. A letter symbol with a light-colored background represents the message with which a choreography task is triggered. The letter symbols of the other messages are displayed with darker backgrounds.

3.5.8 Classification

In recent years, BPMN has advanced to be the standard choice for modeling business processes in industrial practice. Its comprehensive set of language elements makes it suitable for many application areas, from documentation to the automation-supported execution of business processes in organizations. The extensive vocabulary can at the same time be seen as a shortcoming of BPMN due to the increased complexity of the notation. In particular, the large number of event types with semantics that are sometimes hard to distinguish leads to increased effort when learning the language.

Potential issues of comprehensibility of the models when using the full set of notation elements is usually countered by using a reduced set of elements in suitable cases. For the descriptive documentation of business processes, it is usually not necessary to use the complete set of events and more complex task types. Only when a process model is to be validated or executed, for example by simulation, is it necessary to enrich the models with information on non-standard cases or exceptions. In such cases, the simpler models can be used as a basis for supplementation.

BPMN is one of few Business Process Modeling languages that explicitly deals with communication processes between participating actors, and enables the modeling of the same. During the development of the language, however, the starting point for specifying communication flows was the coupling of technically distributed information systems. BPMN implicitly assumes that within a pool (i.e., between lanes) it is not necessary to explicitly represent communication between actors, because they all have access to the same information infrastructure. Message flows are only modeled between pools. They are used during execution to describe the mapping of the data structures used in the source pool to those of the target pool. A message flow therefore essentially corresponds to a data transfer from one information system to another and therefore always represents a communication process with exactly one sender and exactly one recipient - several recipients cannot be addressed with a single message. While this mechanism can also be used to represent non-technical communication, its expressiveness is limited. In particular, communication between two or more actors without clearly definable messages can only be modeled in non-standard-compliant and ambiguous ways. This limitation is owed to the claim of the executability of the created processes and also exists in other communication-oriented approaches.

In addition, BPMN focuses on processes with a fully specifiable control flow. It reaches its limits when process parts are strongly case-specific and cannot be described in detail in advance. In recent years, different approaches have emerged for such processes, which either adopt a declarative modeling approach for representing the execution conditions of process parts or focus on the communication processes between the actors involved. As an example for the latter category, we introduce Subject-oriented Business Process Modeling (S-BPM) in the next section.

3.6 S-BPM

In contrast to other modeling approaches, subject-oriented process modeling describes business processes primarily from the point of view of communicating actors or systems.

When modeling according to the subject-oriented approach, the subjects as representatives for those involved in a process are the focus and starting point of representation. It essentially describes who communicates with whom in which form and how the individual actors react to received messages. Communication is described by defining the messages that are exchanged between the subjects. The behavior of the subjects is described separately by state diagrams, whereby three different state types are used. A subject can wait for a message, send a message, or do something without communicating with other subjects. The latter state type is called the function state and is used to describe the actual behavior, that is, the activities of a subject.

3.6.1 Notation Elements

When modeling according to the subject-oriented approach, the subjects as representatives for those involved in a process are at the center of attention for modeling. The modeling of a process essentially takes place in two stages with an increasing level of detail. First the *interaction diagram is* created, in which the subjects and their message exchange are modeled. In a further stage, the behavior of each subject is described in a separate *behavior diagram.*

For the *interaction diagram (cf., Figure 3.46), the subjects* involved in a process are defined first. A subject is an active entity but does not necessarily have to be a human actor. Technical systems can also be subjects, as long as they play an active role in the process. Subjects must always be described abstractly, i.e., not for specific persons or machines, but on the basis of the necessary tasks to be fulfilled in the process (e.g., "application examiner" and not "Mr. Miller").

Messages are exchanged between the subjects. The interaction diagram only defines which messages exist and who sends and receives them. The order of the messages is not defined here.

For each subject, a *behavior diagram* describes the order in which it sends and receives messages or executes functions (cf., Figure 3.47). The individual states are

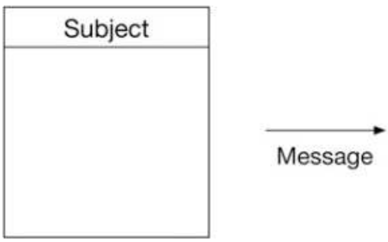


Figure 3.46: Notation of S-BPM interaction diagrams

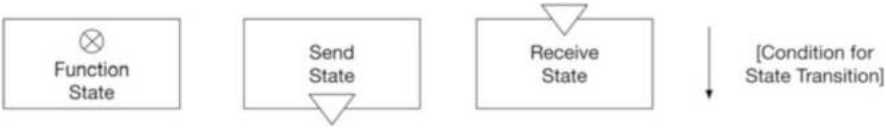


Figure 3.47: Notation of S-BPM behavior diagrams

related to each other by connections which describe the conditions of the transition from one state to the next. Their use depends on the type of condition used:

For each *function state,* what is to be done in the respective behavior step is described. The end conditions of a function state correspond to the outgoing connections that emanate from the respective function state. If the function can lead to different results, different subsequent states can be activated via different transition conditions. This enables the representation of alternative behavior patterns.

In a *send state,* a message is transmitted to a recipient. The subject remains in the state until the recipient is able to receive the message. Who the recipient of the message is and which message is transmitted is described at the outgoing connection of the send state.

The respective subject remains in a *receive state until* one of the messages that the receive state can accept has arrived. Since different messages can be accepted in any particular receive state, different subsequent states can be activated depending on the type of message received. For this purpose, several outgoing connections can be used to describe which message from which sender leads to the corresponding state transition. In this way, it is also possible to react differently to the same message from different senders.

3.6.2 Examples

In order to work out the differences and similarities to the previously discussed modeling languages, we again make use of the examples already used above. In the first example there is no communication, therefore we focus on the behavior diagram of the only subject involved.

The example in Figure 3.48 shows a process with a single task in which an application (for which we have no further information here) is processed. The process ends after the processing of the application has been completed. A behavior diagram must always have a start state, which is marked by a triangle in the upper left corner. There must also be an end state, marked by a triangle in the lower right corner. To fully describe the behavior of the subject, we need a state transition that identifies under what conditions the state “Check application” can be left. Therefore, we insert a state "Done" here, which we mark as the final state and which does not contain any expected activities.

In Figure 3.49, the process is extended by a decision with three possible results that are mutually exclusive. The application is checked, and the result of this

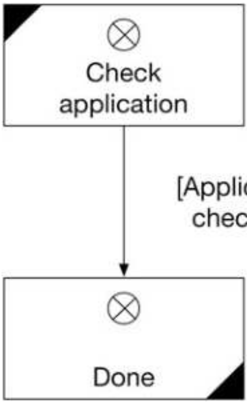


Figure 3.48: Simple S-BPM behavior diagram

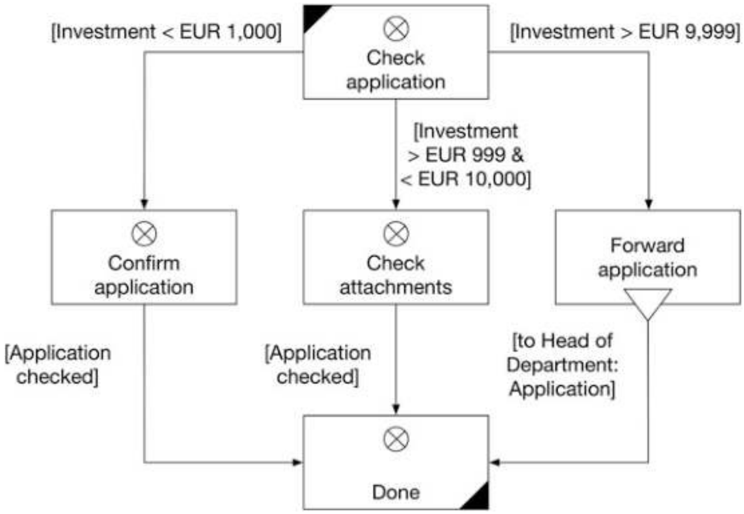


Figure 3.49: S-BPM behavior diagram with a multiple-outcome decision

examination allows a decision to be taken on further processing. In the case of an investment sum of EUR 10,000 or more, the application will be forwarded. We indicate this by a send state and specify at the outgoing connection who is to receive the request. For the process to be fully specified, a behavior diagram for the head of department would also have to be created at this point.

S-BPM also allows the execution of parts of a process repeatedly. For this purpose, a connection is inserted at the end of the part to be repeated, provided with a repetition condition and returned to the first state of the part to be repeated

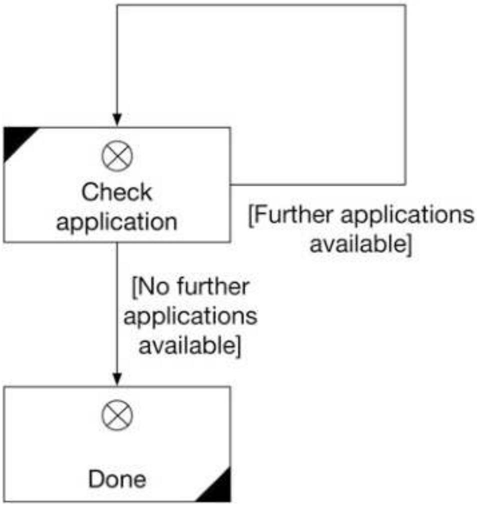


Figure 3.50: S-BPM behavior diagram with repeated execution of process parts

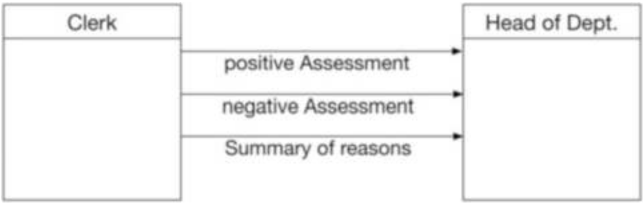


Figure 3.51: Example of S-BPM interaction diagram

(cf., Figure 3.50). The other outgoing connection continues the process after repetition completion.

The example in Figure 3.51 shows the interaction diagram of an application process with two subjects, an clerk and a head of department. Figure 3.52 shows the behavior diagrams of the clerk and head of department respectively. The positive or negative assessment of an application is transmitted as a message. The summary of reasons giving details of the assessment is only sent in the case of a negative assessment. If the application is confirmed, the "Summary of reasons" message is not transmitted, so we can assume that no further justification will be given in this case.

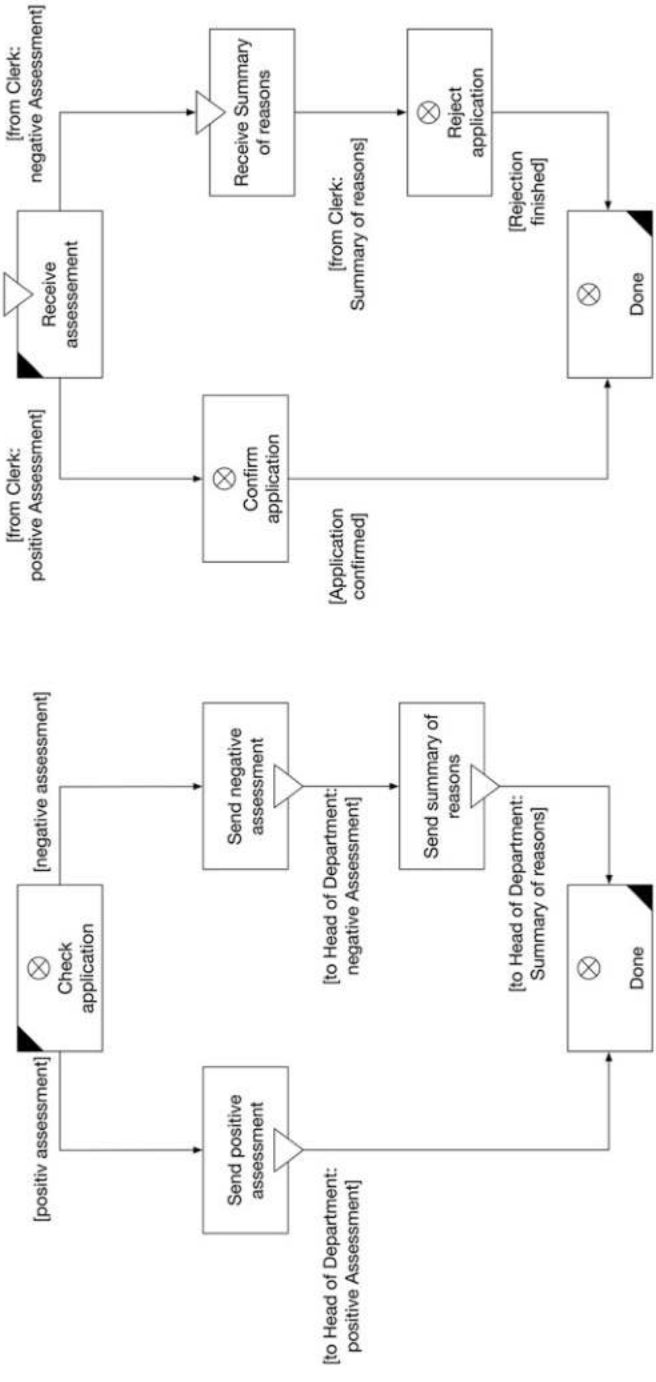


Figure 3.52: Examples of S-BPM behavior diagrams

3.6.3 Advanced Forms of Communication Modeling and Exception Handling

The focus of S-BPM on representing communication processes is reflected in a more comprehensive and flexible description of communication than in all the modeling languages considered above. In particular, S-BPM allows the representation of more complex communication scenarios through the use of input pools and the detailed description of the data exchanged in messages by means of business objects, as well as reacting on unforeseen messages by means of exception handling via message guards.

Input pools

An *input pool* provides a subject with a mailbox in which incoming messages are stored until they are required in the behavior diagram. In contrast to a simple mailbox, an input pool is configurable. For each subject, the number of storable messages can be specified per type. If the input pool is not able to accept a message according to its configuration, the sender must remain in the send state until the message can be delivered. This allows different communication scenarios to be represented. If the capacity of the input pool for a particular message type is reduced to 0, the sender must always wait until the recipient is ready to receive the message. This is referred to as synchronous communication. If the input pool is configured to accept an arbitrary number of messages, the sender never has to wait until the receiver is in the state in which it can accept the message. This is called asynchronous communication (BPMN only allows the representation of this type of communication). Input pools also enable messages to be received in any order. The messages do not have to be processed in the order they arrive but can be processed according to the recipient's requirements.

Input pools have no graphical equivalent in S-BPM but are a concept of execution semantics. They are described for each subject textually or in a configuration tool. If no input pools are defined, the default configuration allows for an unlimited number of messages of any type to be stored. The communication behavior therefore corresponds to the message flows of BPMN (asynchronous communication).

Business Objects

Business objects are used to specify the data that is required for executing the tasks in a business process. Business objects are passive, i.e., they do not initiate any interactions or trigger any actions. Business objects are processed and modified by subjects and can be assigned to messages in order to specify their content in more detail.

As for input pools, there is no graphical equivalent for business objects in the notation of the modeling language. Business objects are concepts of execution semantics and therefore are dependent on the technical execution environment. They are usually described in tabular form. The basic structure of business objects consists of an identifier, data structures and data elements. The identifier of a business object is derived from the business environment in which it is used. Examples are business trip request, order, delivery note, invoice, etc.

Business objects consist of data structures whose components can be simple data elements of a certain type (for example, character string or number) or nested complex data structures. To allow for better comprehensibility, it is recommended to describe the meaning of the data elements in more detail, especially if the meaning cannot be derived from the identifiers.

Figure 3.53 shows an example of a business trip request. This consists, among other things, of the data structure 'Data on requester (employee)' with the data elements for last name, first name and personnel number, and the structure 'Data on business trip' with the data elements for start, end and purpose of the trip.

In many cases, the semantics of a business object can change during process execution, for example when a delivery note is transformed to an invoice. Several different statuses can therefore be defined for a business object. When the status is changed, only the data structures or data elements of the previous status that are

Data structure	Meaning	Data type	Can/must	Value range/Default
Data of requester				
Name	Last name	Character	M	
First name	First name	Character	M	
Personnel number	...	Integer	M	
Organizational unit	...		C	
Pay group	...		C	
Data of trip				
Start trip	...	Date	M	Whitin 1 year from current date/ current date
End trip	...	Date	M	Start trip plus 1 year/ start trip
International trip	...	Boolean? ?	C	y/n; n
Travel destination (city7country)	...	Character	M	...
Reason for traveling	...	Character	M	...
Desired advance money	...	Integer	C	...
Data of approval				
Approval	Approval comment	Boolean? ?	M	y/n; n
Cost center	...	Integer	M	...
Desired advance money	...	Integer	C	...

Figure 3.53: Example of S-BPM business object (business trip request)

required in the new status are transferred, and new components are added if required. This ensures that a subject only receives the data it really needs for its work. This also facilitates compliance with data protection regulations.

In the business trip request example, the status "Business trip booking" can be derived from the original status "Travel request" of the business object (cf., Figure 3.54). In particular, data elements with internal specifications such as personnel number, compensation group or reason for trip are removed, and thus do not to leave the company if the business object is sent to a travel agency for booking the travel arrangements. As shown in the following Figure 3.54, a new data structure "Booking data" is inserted for this purpose. It contains data elements with which a deadline can be specified for the travel agency to return booking confirmations or to specify certain hotel chains which are preferred by the company.

Message Guards

Handling of an exception (also termed message guard, message control, message monitoring, message observer) is a behavioral description of a subject that becomes relevant when a specific, exceptional situation occurs while executing a subject behavior specification. It is activated when a corresponding message is received, and the subject is in a state in which it is able to respond to the exception handling. In such a case, the transition to exception handling has the highest priority and will be enforced.

Exception handling is characterized by the fact that it can occur in a process in many behavior states of subjects. The receipt of certain messages, e.g., to abort the process, always results in the same processing pattern. This pattern would have to be modeled for each state in which it is relevant. Exception handlings cause high modeling effort and lead to complex process models, since from each affected state a corresponding transition has to be specified.

To illustrate the compact description of exception handlings, we use a service management process with the subject “service desk” (cf., Figure 3.55). This subject identifies a need for a business trip in the context of processing a customer order - an employee needs to visit the customer to provide a service locally. The subject “service desk” passes on a service order to an employee. Hence, the employee issues a business trip request. In principle, the service order may be canceled at any stage during processing up to its completion. Consequently, this also applies to the business trip application and its subsequent activities.

This relatively simple example already shows that taking such exception messages into account can quickly make behavior descriptions difficult to understand because additional receive states have to be added. In these additional receive states it is checked whether a corresponding cancel message has arrived. The concept of exception handling enables supplementing exceptions to the default behavior of subjects in a structured and compact form. Figure 3.55 below shows how such a concept affects the behavior of the employee.

Instead of incorporating additional receive states with a timeout zero in order to check whether a message has arrived which interrupts the standard control flow, the behavior description is enriched with an exception handling for the message “service

Data structure / Data element	Meaning	Data type	Can/must	Value range/Default
Data of requester				
Name	Last name	Character	M	
First name	First name	Character	M	
Date of trip				
Start trip		Date	M	Within 1 year from current date/ current date
End trip		Date	M	Start trip plus 1 year/ start trip
Travel destination (city/country)		Character	M	
Data of booking				
Contracted hotel chains	Approval comment	Character	M	
Deadline of booking confirmation		Date	C	
Booking confirmation		Date	M	y/n

Figure 3.54: Example of business object in different state (Business trip booking)

Figure 3.55: Behavior of subject “employee” with exception handling

cancellation”. Its initial state is labeled with the states from which it is branched to, once the message ‘service cancellation’ is received. In the example, these are the states ‘fill out Bt-request’ and ‘receive answer from manager’. Each of them is marked by a triangle on the right edge of the state symbol. The exception behavior leads to an exit of the subject, after the message ‘service cancellation’ has been sent to the subject ‘manager’.

A subject behavior does not necessarily have to be brought to an end by an exception handling; it can also return from there to the specified default behavior. Exception handling behavior in a subject may vary, depending on from which state or by what type of message (cancellation, temporary stopping of the process, etc.) it is triggered. The initial state of exception handling can be a receive state or a function state.

Messages, like ‘service cancellation’, that trigger exception handling always have higher priority than other messages. This is how modelers express that specific messages are read in a preferred way. For instance, when the approval message from the manager is received in the input pool of the employee, and shortly thereafter the cancellation message, the latter is read first. This leads to the corresponding abort consequences.

Since now additional messages can be exchanged between subjects, it may be necessary to adjust the corresponding conditions for the input pool structure. In particular, the input pool conditions should allow storing an interrupt message in the input pool.

Behavior Extensions

When exceptions occur, currently running operations are interrupted. This can lead to inconsistencies in the processing of business objects. For example, the completion of the business trip form is interrupted once a cancellation message is received, and the business trip application is only partially completed. Such consequences are considered acceptable, due to the urgency of cancellation messages. In less urgent cases, the modeler would like to extend the behavior of subjects in a similar way, however, without causing inconsistencies. This can be achieved by using a notation analogous to exception handling. Instead of denoting the corresponding diagram with ‘exception’, it is labeled with ‘extension’.

Behavior extensions enrich a subject’s behavior with behavior sequences that can be reached from several states equivocally.

For example, the employee may be able to decide on his own that the business trip is no longer required and withdraw his trip request. Figure 3.56 shows that the employee is able to cancel a business trip request in the states ‘send business trip request to manager’ and ‘receive answer from manager’. If the transition ‘withdraw business trip request’ is executed in the state ‘send business trip request to manager’, then the extension ‘F1’ is activated. It leads merely to canceling of the application. Since the manager has not yet received a request, he does not need to be informed.

In case the employee decides to withdraw the business trip request in the state ‘receive answer from manager’, then extension ‘F2’ is activated. Here, first the supervisor is informed, and then the business trip is canceled.

Choice Segments

So far, the behavior of subjects has been regarded as a distinct sequence of internal functions, send, and receive activities. In many cases, however, the sequence of internal execution is not important.

Certain sequences of actions can be executed in arbitrary order, this is called freedom of choice. In this case, the modeler does not specify a strict sequence of activities. Rather, a subject (or concrete entity assigned to a subject) will organize to a certain extent its own behavior at runtime.

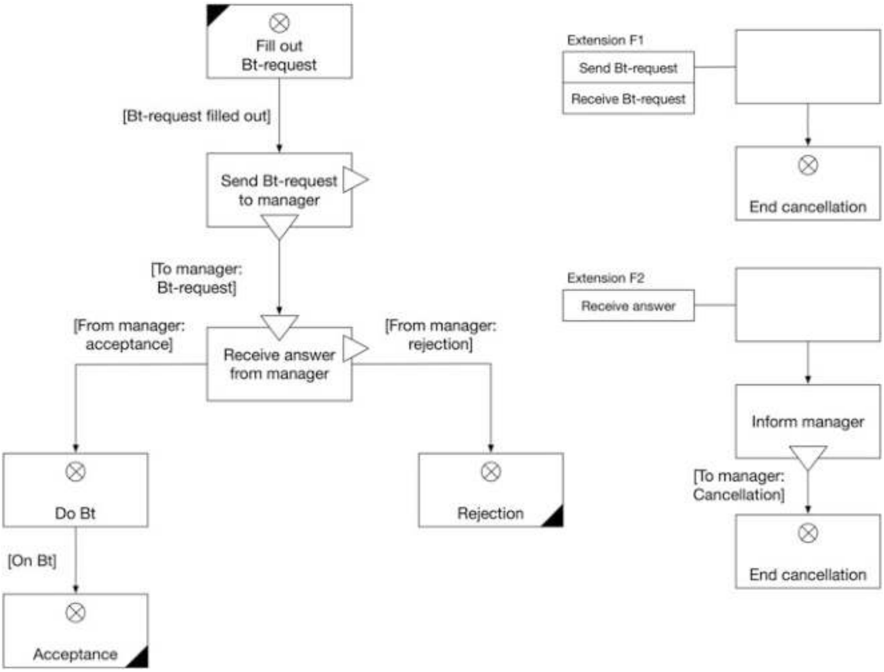


Figure 3.56: Example for S-BPM behavior extensions

The freedom of choice with respect to behavior is described as a set of alternative clauses which outline a number of parallel paths. At the beginning and end of each alternative, switches are used: A switch set at the beginning means that this alternative path is mandatory to get started, a switch set at the end means that this alternative path must be completely traversed. This leads to the following constellations:

- Beginning is set / end is set: Alternative needs to be processed to the end.
- Beginning is set / end is open: Alternative must be started, but does not need to be finished.
- Beginning is open / end is set: Alternative may be processed, but if so must be completed.
- Beginning is open / end is open: Alternative may be processed, but does not have to be completed.

The execution of an alternative clause is considered complete when all alternative sequences, which were begun and had to be completed, have actually been entirely processed and have reached the end operator of the alternative clause.

Transitions between the alternative paths of an alternative clause are not allowed. An alternate sequence starts in its start point and ends entirely within its end point.

Figure 3.57: Example of process alternatives

Thus, the core property of a state chart, namely that a state chart always only contains a single active state is not violated

Figure 3.57 shows an example for modeling alternative clauses. After receiving an order from the customer, three alternative behavioral sequences can be started, whereby the leftmost sequence, with the internal function 'update order' and sending the message 'deliver order' to the subject 'warehouse', must be started in any case. This is determined by the 'X' in the symbol for the start of the alternative sequences (gray bar is the starting point for alternatives). This sequence must be processed through to the end of the alternative because it is also marked in the end symbol of this alternative with an 'X' (gray bar as the end point of the alternative).

The other two sequences may, but do not have to be, started. However, in case the middle sequence is started, i.e., the message 'order arrived' is sent to the sales department, it must be processed to the end. This is defined by an appropriate marking in the end symbol of the alternative ('X' in the lower gray bar as the endpoint of the alternative). The rightmost path can be started but does not need to be completed. This kind of visualization radically simplifies the representation of the state transitions that would be necessary in case a traditional state chart visualization would be used (using a single connection for each potential state transition).

The individual actions in the alternative paths of an alternative clause may be arbitrarily executed in parallel and overlapping, or in other words: A step can be executed in an alternative sequence, and then be followed by an action in any other sequence. This gives the performer of a subject the appropriate freedom of choice while executing his actions.

3.6.4 Classification

In contrast to the other modeling languages discussed so far, there is no single diagram in S-BPM that fully describes a business process. Rather, separate behavior diagrams are created for all subjects, which are linked by an interaction diagram describing the message exchange. S-BPM thus enables a loose coupling of process parts and an easier adaptation of the behavior of a subject, as long as its communication interface, i.e., the set of received and sent messages and their sequence, remains unchanged.

The use of state diagrams to describe the behavior of a subject also constitutes a fundamental difference to the other languages discussed so far. A state diagram – as already indicated by its name - describes the state of a system (here: a subject - this can be a human as well as a machine) and the events that lead to a state transition. A subject can only be in exactly one state at any one time - it is therefore by definition not able to execute process steps in parallel. Rather, all subjects work in parallel and independently of each other. This requires a different approach to modeling, since constructs such as AND connectors (in EPCs), split/joins (in activity diagrams) or parallel gateways (in BPMN) are not available. At the same time, this modeling approach leads to simpler, more compact models and, in contrast to BPMN, a significantly reduced range of language constructs, which contributes to the comprehensibility of the models.

3.7 Comparison

The modeling languages considered here have different expressive power and, due to their historical development, have different focal points in their approach to represent business process [9]. The following section attempts to summarize these differences again systematically using the process definition presented in the former chapter and thus to compare the languages with respect to their expressiveness. We use the semantics of the presented modeling elements as a starting point.

The point of reference for the following considerations is the process definition from the last chapter, which we will reiterate here for simplicity's sake:

Table 3.1: Concepts included in process definition

Definition model	Concept
1a	Beginning
	Input
1b	End
	Result
1c	Customers necessity
2a	Activities / tasks
2b	Start event
	Doer
2c	Logical order
	Chronological order
2d	Premise
3a	Human
	Machine
3b	Material resources
	Information
	Application program
	General aids

- Process strategy: A process has
  - a defined start and input (start event),
  - and has a defined end with a result,
  - that contributes to the satisfaction of a customer's needs (and thus to the creation of value)
- Process logic: A process
  - is the sum of linked activities (tasks),
  - which, after the start event, are used by actors
  - in logical and chronological order
  - for processing a business object in order to
  - generate the desired result.
- Process realization: A process is realized
  - with people and/or machines, that take over the tasks of the respective actor, and carry these tasks out
  - with tools (equipment, information, application programs, etc.).

On the basis of this definition, the concepts can be identified which should be representable in a process model in order to be able to model processes comprehensively (according to this definition). Table 3.1 shows these concepts. Concepts that occur more than once are only mentioned when they occur for the first time - such as "result". In the case of concepts which are described in different degrees of detail, only the more concrete concepts are considered - e.g., "linked activities" as a more general formulation of "logical and chronological order".

If one now assigns the modeling elements of the languages considered to these concepts, Table 3.2 results.

Table 3.2: Concept allocation to the notation elements in different modeling languages

Definition part	Concept	Flowchart	eEPK	UML activity diagram	BPMN	S-BPM
1a	Beginning	Termination element	Event	Starting node	Start event	Starting state
	Input	-	Information object	Data object / received signal	Data object, message	Business object, message
1b	End	Termination element	Event	End node	End event	End state
	Result	-	Information object	Signal	Data object, message	Business object, message
1c	Customers' necessity	-	-	-	-	-
2a	Activities/ tasks	Operation	Function	Action	Tasks (different types)	Functional status
2b	Start event	Only unspecific	Specifically named event	Only unspecific, perhaps by received signal	Start event (message, timer, regulate)	Starting state, usually via the receipt of a message
	Actor	-	Organizational unit	Partition	Pool, Lane	Subject
2c	Logical order/ Chronological order	Flow arrow & decision	Flow arrow & Connectors for alternative and parallel processes, data flow	Flow arrow, decision, Split/Join	Sequence flow, alternate and parallel gateways, message flow, exception handling, transactions, choreography	Messages between subjects, conditions for state transitions in subject behaviors
2d	Business object	-	Information object	Data object	Data object, message	Data object, message
3a	Human	-	Organizational unit	Partition	Pool, Lane. User task, manual task	Subject
	Machine	-	-	-	Service Task	Subject

(continued)

Table 3.2: (continued)

Definition part	Concept	Flowchart	eEPK	UML activity diagram	BPMN	S-BPM
3b	Material resources	-	-	-	-	-
	Information	-	Information object	Data object	Data object, message, data storage	Business object, message
	Application program	-	Application system	-	Service-Task	Subject
	General aids	-	-	-	-	-

The conceptual coverage obviously varies across languages. The table also shows that not all languages address all concepts to the same extent or with the same level of expressiveness. The allocation of the modeling elements to the concepts provides a first starting point for estimating the expressiveness of the respective languages.

In this overview, the different approaches in the illustration of the logical and chronological connections are only partially recognizable. Here, the languages differ considerably: flowcharts do not offer the possibility of representing parallel processes, EPCs only allow for strong coupling of parallel activity branches by linking them within a process by means of AND or OR operators. UML activity diagrams and BPMN offer the same mechanisms (under different names), but also allow for loose coupling of processes or process parts by means of signals (for activity diagrams) or message flows (for BPMN). Especially the latter mechanism allows a detailed description of communication processes of basically independent process parts. Flexibility, however, is restricted by the necessary unique assignment of sender and receiver for each single message. S-BPM offers a similar communication mechanism, but is more flexible here (especially when using input pools that are not shown in the graphical representation of the language). In S-BPM, a description of process parts running in parallel is only possible by distributing them to different subjects - within a subject, only one functional state can be active at a time, i.e., only alternative branches in the behavior of a subject can be represented.

In general, BPMN offers the greatest flexibility in the choice of how to represent a process. Due to the large number of modeling elements, even complex real-world phenomena can be represented in a compact way. This, however, leads to higher demands on language comprehension for the model users. Activity diagrams or S-BPM, which are based on a compact set of modeling elements, follow a different approach here. Their approach leads to larger models in complex contexts, which in turn places higher demands on the model users with regard to their understanding of the model. S-BPM reduces the immediately visible complexity of models by distributing a process over different subjects. While this leads to partial models which can be more easily grasped, it in turn places higher demands on model users when it comes to grasping the overall context within the process.

When selecting a modeling language that is suitable for a given task and target group, not only the object of the model (i.e., the business process under consideration) and the target of modeling should be considered. The known or assumed competencies of the modelers and model users also need to be taken into consideration [3, 10, 11]. A fundamental distinction can be made between languages that focus on the flow of activity (such as flowcharts and EPCs) and those that focus on the actors in a process and their communication (such as S-BPM). BPMN and activity diagrams are basically suitable for both types of representation, whereby BPMN offers more expressive means for representing communication processes. The final selection of a modeling language after determining the fundamentally pursued representation approach (activity flow vs. communication flow) is ultimately dependent on the preferences of the modelers or model users.

References

1. Curtis, B., Kellner, M. I., & Over, J. (1992). Process modeling. *Communications of the ACM, 35*, 75–90. https://doi.org/10.1145/130994.130998.
2. Kunze, M., Luebbe, A., Weidlich, M., & Weske, M. (2011). Towards understanding process modeling – The case of the BPM academic initiative. In *International workshop on business process modeling notation* (pp. 44–58). Berlin: Springer.
3. Recker, J., Safrudin, N., & Rosemann, M. (2012). How novices design business processes. *Information Systems, 37*, 557–573. https://doi.org/10.1016/j.is.2011.07.001.
4. Herrmann, T., & Loser, K. (1999). Vagueness in models of socio-technical systems. *Behaviour & Information Technology, 18*, 313–323.
5. Nüttgens, M., & Rump, F. (2002). Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In *Promise* (pp. 64–77).
6. Dumas, M., & ter Hofstede, A. (2001). UML activity diagrams as a workflow specification language. In *UML 2001 – The unified modeling language modeling languages, concepts, and tools* (pp. 76–90). Berlin: Springer.
7. White, S., & Miers, D. (2008). BPMN modeling and reference guide: Understanding and using BPMN. *The Journal of Strategic Information Systems, 3*, 23–40. https://doi.org/10.1016/0963-8687(94)90004-3.
8. Fleischmann, A. (2010). What is S-BPM? In *S-BPM ONE – Setting the stage for subject-oriented business process management* (Vol. 85, pp. 85–106). Heidelberg: Springer.
9. Soderstrom, E., Andersson, B., Johannesson, P., et al. (2002). Towards a framework for comparing process modelling languages. In *Proceedings of the 14th international conference on advanced information systems engineering* (CAiSE) (Vol. 2348, pp. 600–611). Berlin: Springer. https://doi.org/10.1007/3-540-47961-9_41.
10. Oppl, S. (2018). Which concepts do inexperienced modelers use to model work? – An exploratory study. In *Proceedings of MKWI 2018.*
11. Soffer, P., Kaner, M., & Wand, Y. (2012). Towards understanding the process of process modeling: Theoretical and empirical considerations. In *Business process management workshops* (pp. 357–369). Berlin: Springer.

4 Contemporary Challenges in Business Process Modeling / Management

In the previous chapters we have considered the means for describing the sequence of activities executed in a process. In the course of digitalization, processes in an organization and, in particular across organizations, are becoming more complex. In order to describe them nonetheless in a transparent way, it is necessary to organize them as networks of processes, or as a hierarchy of subprocesses.

When processes are increasingly supported by IT, two major aspects of IT support seem to be crucial. One aspect concerns the support of the activities specific for a process, e.g., creating a purchase order. The corresponding software applications are specific for each process. Normally, activities in a process are implemented by functions of such applications or, the other way around, activities of a process represent functions of application systems.

The other aspect concerns the control of the allowed sequences of activities, which is managed by software solutions in the form of so-called workflow systems. This software for controlling the execution sequence of a process can be derived directly from the process model if the syntax and semantics of the modeling language are precisely defined. This is a very important aspect for the digitalization of processes.

In workflow systems functions of software applications which implement certain actions of a process are incorporated into the associated functions of the model. A more detailed discussion on the structure of business process implementations can found in Chapter 7.

In the following sections we describe how the different Business Process Modeling languages support the structuring of complex process systems and to what extent the derivation of digital workflow is supported.

A. Fleischmann et al., *Contextual Process Digitalization*,
https://doi.org/10.1007/978-3-030-38300-8_4

4.1 Handling of Complex Processes

In an organization, business processes are connected with each other either directly or indirectly. The sales process is connected with the order handling process, the order handling process initiates the delivery and invoice processes, and so on. A language for specifying process behaviors should also offer possibilities for structuring complex interconnected process systems. It should allow the representation of the environment of a considered process, which means it should be possible to illustrate the relationships to other processes. If we want to define the activity sequences of the order handling process, we must be able to include the relationships to the sales and shipment processes. The interfaces to these neighboring processes should include the methods for exchanging the data and switching the control flow between the processes. In this chapter we describe the various possibilities to structure complex processes in the modeling languages defined in the previous chapter.

4.1.1 Structuring Complex Processes in Flowcharts

The only way to structure complex processes in flowcharts is by means of predefined processes. Predefined processes are named processes which are defined elsewhere. The following figure shows an example for using predefined processes.

In Figure 4.1 the process on the left side uses the predefined process “do shipment”. This process is shown on the right side of the figure.

In flowcharts the control flow switches from the process which initiates a predefined process to the predefined process itself. There is no standardized way to describe a process architecture providing an overview of which predefined processes exist and in which other processes they are used.

In flowcharts it is assumed that all processes share all data. Hence, it is not necessary to specify the data required by a predefined process explicitly. Due to this tight coupling of processes to predefined processes, flowcharts are not the ideal

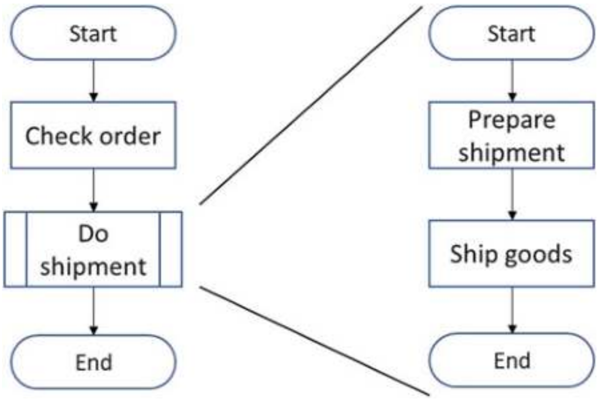


Figure 4.1: Structuring complex processes using flowcharts

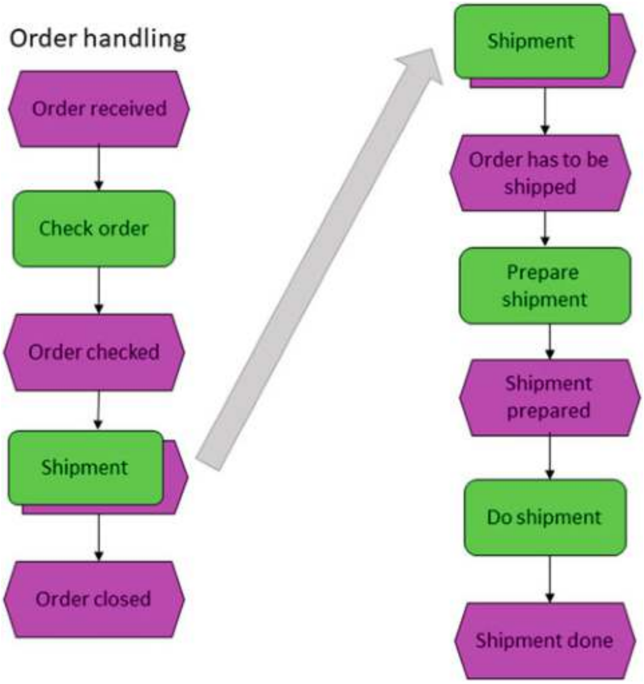


Figure 4.2: Structuring complex processes using EPCs

choice for loosely coupled organizations. In particular, in the case of cross-company processes, the ability to define where the process data is stored is essential.

4.1.2 Structuring Complex Processes in Event-Driven Process Chains

EPCs use subprocesses for structuring complex processes. The incorporation of subprocesses in EPCs is similar to predefined processes in flowcharts. Figure 4.2 shows the use of the interface symbol for integrating subprocesses.

The subprocess approach is not sufficiently expressive to describe complex process systems. Especially if it is necessary to describe the relations between various processes, the problems are the same as with flowcharts. Therefore, Value Chain Diagrams (VCD) have been introduced as an additional model type. VCDs allow describing which processes belong to a complex process system and how they are related to each other. Figure 4.3 shows that the process "order handling" starts the process "shipment".

In addition to this successor relationship, there is also the possibility to describe hierarchical relations. Figure 4.4 shows that the process system "order management"

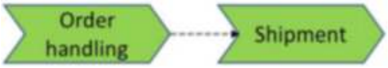


Figure 4.3: Order handling invokes Shipment

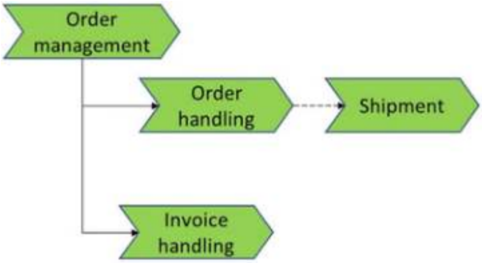


Figure 4.4: Process hierarchy

consists of the process sequence "order handling" and "shipment", and the process "invoice handling".

4.1.3 Structuring Complex Processes as UML Activity Diagrams

UML activity diagrams do not contain a specific notational support for structuring complex processes. It is not allowed to use an activity diagram within another activity diagram (recursive use). The only possibility for structuring complex processes is to connect different activity diagrams by means of messages exchanged between them. However, there is no diagram type to specify a communication relationship between activity diagrams.

4.1.4 Structuring Complex Processes in BPMN

In BPMN the highest modeling level is represented by collaboration diagrams. If there are many involved parties in a process, and thus many pools with several swim lanes, then the 'big picture' of a process system may become quite difficult to understand. In order to overcome this problem, conversation diagrams have been introduced. Conversation diagrams represent an overview of a network of partners and how they communicate with each other. Figure 4.5 shows an example of a conversation diagram.

In this diagram a process system with the pools "order handling" and "shipment" is depicted. These pools communicate with each other via the message "start shipment". Conversation diagrams represent a top-level view of a BPMN collaboration diagram. Since nesting conversation diagrams is not allowed, the rectangle must be a pool and cannot consist of a conversation diagram on a lower level. Analogously to the predefined process concept in flowcharts or the subprocess

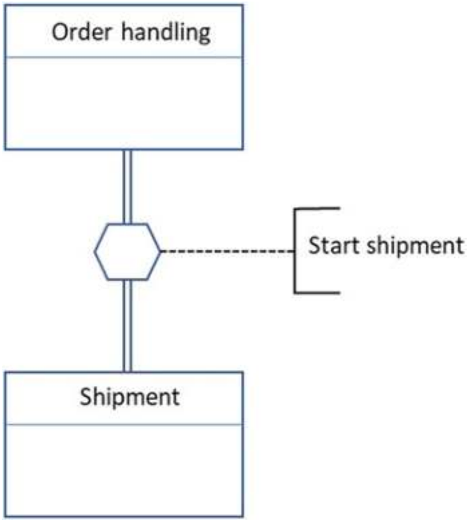


Figure 4.5. Structuring complex processes using conversation diagrams

concept in EPCs, there is also a subprocess concept in BPMN. This concept has already been described in the previous chapter.

4.1.5 Structuring of Complex Processes in S-BPM

In Chapter 3 we introduced the modeling language S-BPM for representing subject-oriented business processes. S-BPM is based on the Parallel Activity Specification Scheme (PASS) [1]. PASS offers useful features for structuring complex process systems in a hierarchical way. Arbitrary levels of descriptions are allowed. We exemplify how levels of communication networks can be used to describe complex process systems with an example of a process for service provision in the case of a car accident service. This service (of an actual company) consists of several connected processes. It encompasses the main process for handling the car accident as well as supporting processes, e.g., for organizing towing and repair shop services, for handling insurance claims, for receiving and paying invoices, etc.

These processes are executed by various organizations, such as help desk service companies, towing service companies, car repair workshops, banks, etc. In most business process projects overall processes are not described completely in detail, but rather only in parts of a process. Which part of the process is represented in detail at a specific point in time depends on the perspective taken by the participants or departments involved in that part of the process. For instance, from the perspective of the help desk, only the help desk process needs to be considered in detail. However, we indeed have to take into account the environment in which a considered process is embedded. We must know which relations exist to other processes. It is necessary to know which inputs are required by neighboring processes and which

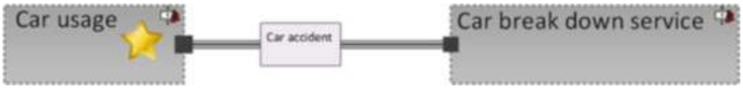


Figure 4.6: Highest process structure level of car accident service

results they deliver. A help desk process which organizes the towing services has to know how the towing service is requested and which further interactions are required. For example, it needs to be agreed whether the towing service, or rather the help desk, informs the client with respect to the arrival time of the tow truck.

Process Architecture

In the following we specify the required process architecture. In the diagrams, rectangles represent processes. Each process has a name. Processes consist of other processes and/or subjects. The lines between the rectangles represent the communication channels between processes. Each communication channel has a name and can contain other communication channels and/or messages.

Figure 4.6 shows the highest process level of the car accident service. In the “car usage” process the event "car accident" happens. In order to organize support an interaction is initiated with the process "car accident service". These processes exchange messages which are elements of the communication channel "car accident handling".

Figure 4.7 shows the next process structure level of the process “car accident service”. In this level the process "car accident service" is decomposed into 10 processes. Eight of these processes, namely, the processes "incident management", "mobility service", "towing service", "insurance service", "car repair workshop", "banking", “payment handling” and “payment services” have a communication channel to the process "car usage". This means the communication channel "car accident handling" is separated into eight communication channels. Each of them covers the communication with the related process, e.g., the communication channel "accident notification" is the communication channel between the processes "car usage" and "incident management".

A process can also encompass other processes. This means that different levels of processes can be built. Figure 4.8 shows the next deeper level of our process hierarchy. The process "car repair workshop" is broken down into 6 processes. According to this separation the communication sets are also split, e.g., the communication set "handling repair service" is split into three parts, one part is handled by the process "service scheduling" the other by the process "car dropping" and the third by the process “customer satisfaction”.

As already mentioned, processes cannot communicate directly with each other, but rather the active entities of a process, the subjects, communicate with each other. This means messages which are sent from one process to another process are received by a subject in that process. Messages belonging to a channel are assigned to respective sending or receiving subjects at the lowest level of a process architecture. This lowest level of a process description is the Subject Interaction Diagram

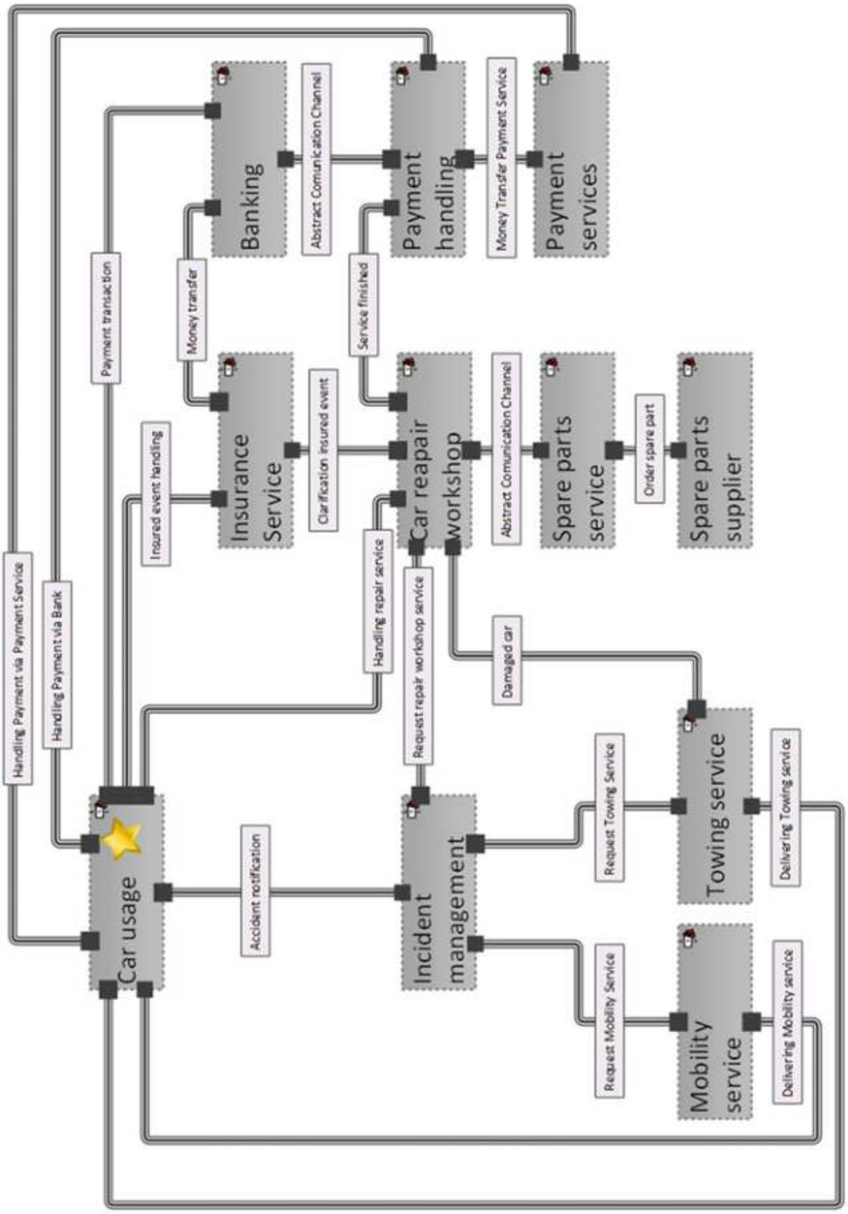


Figure 4.7: Structure of the "car accident service" process

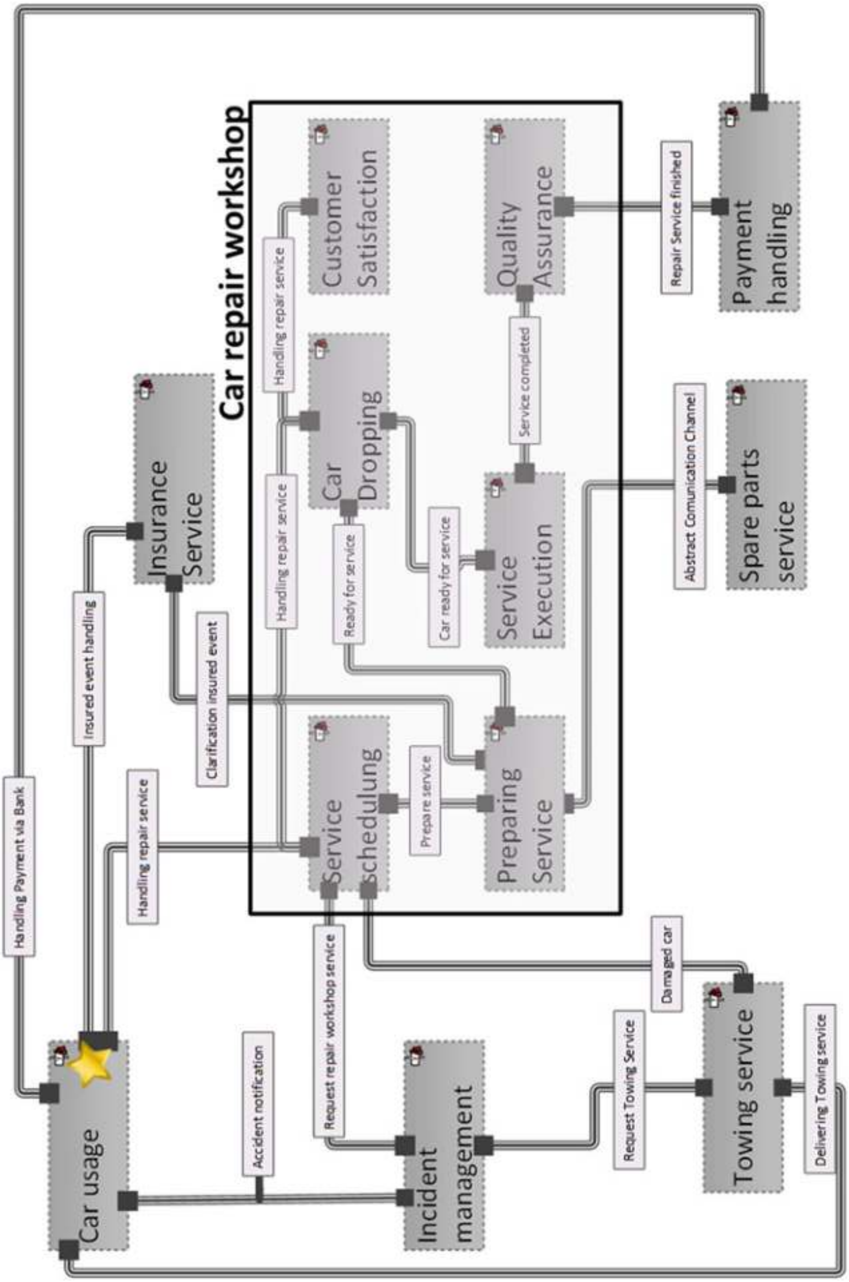


Figure 4.8: Details of the "car accident service" process

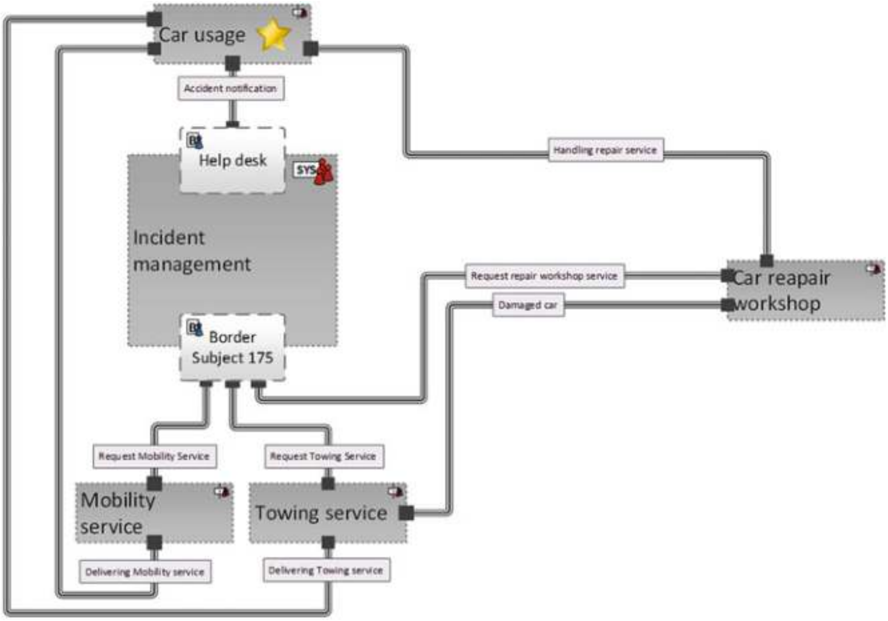


Figure 4.9: Interface (border) subjects of the “incident management” process

(SID) which shows the involved subjects of a process and the messages they exchange.

In the following we consider the process "incident management". This process does not contain other processes as in the case of the process “car repair workshop”. The process “incident management” contains a SID. Some of the subjects of a process communicate with subjects of other processes. These subjects are called border subjects because they are at the border of a process to other processes. Figure 4.9 shows the process “incident management” with its border subjects. There is a border subject “help agent” which communicates with the processes “towing service”, “mobility service”, and “car repair workshop”, or more precisely, it communicates with a respective subject in each one of these processes. Another border subject of the process “incident management” which is called “help desk” communicates with a subject in the process “car usage”.

The border subjects of the process “incident management” must have corresponding border subjects in those processes with which it communicates. The border subject "help desk" communicates with the associated border subject of the process "car usage" and the border subject "help agent" communicates with the respective border subjects of the processes "car repair workshop", "towing service” and "mobility service". The process “incident management” with all of its border subjects is shown in Figure 4.10.

The border subjects of the processes “mobility Service”, “towing Service”, and “car repair workshop” have the same name “service agent” but are different subjects

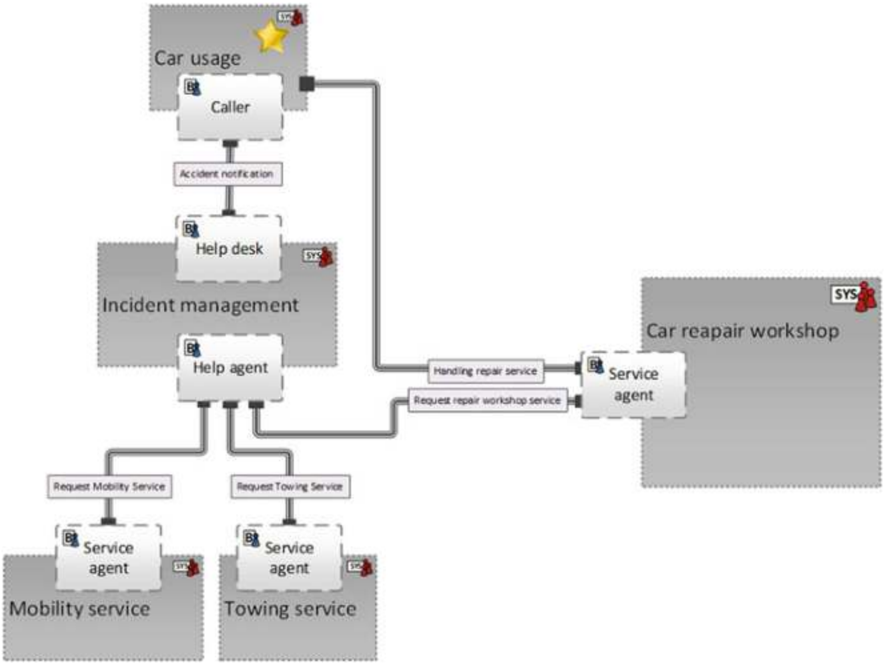


Figure 4.10: "Incident management" process with all border subjects

because they belong to different processes. Since the process "car repair workshop" consists of several layers, the corresponding border subject can belong to a process which is part of the process "car repair workshop" on a lower level.

From the perspective of the subjects inside the process "incident management", the border subjects of the processes "mobility service", towing service", and "car repair workshop" are interfaces to these processes, therefore they are called interface subjects in the (SID) Subject Interaction Diagram of a process. Figure 4.11 shows the SID of the process "incident management".

Behavioral Interface

Processes to which a considered process has communication relationships are called process neighbors, or just neighbors. Now we want to consider the details of the communication relationships between two neighbors. The interface between two processes is defined by the related border subjects and the allowed sequences in which the messages in a communication channel are exchanged between them [2]. As already described above, each message is defined by a name, and the data which are transported is the so-called payload.

A border subject observes the behavior of the border subject of the neighbor process and vice versa. Figure 4.12 shows the border subject "help desk" of the process "incident management" which communicates with the border subject "caller" of the process "car usage". Since we are considering the process "incident

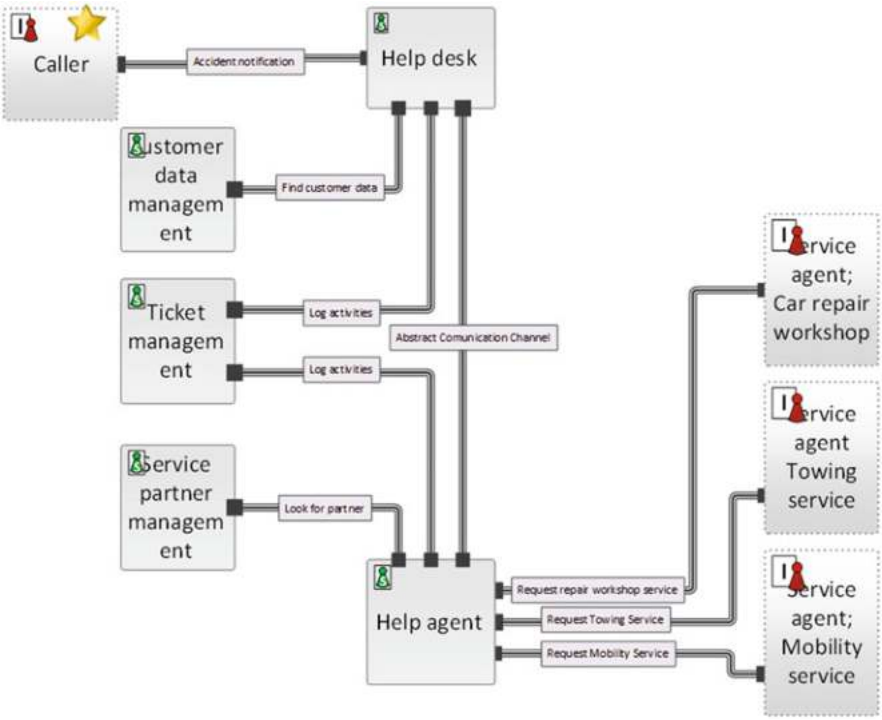


Figure 4.11: Subject Interaction Diagram of the "incident management" process

management", the border subject "caller" of the process "car usage" becomes an interface subject in the SID of the process "incident management".

Figure 4.13 shows the SID of the subject "help desk". Rather than specifying all of the channels, only the messages required for a towing service request are shown. A message "request towing service" stems from the interface subject "caller". This message is accepted by the subject "help desk". The subject "help desk" checks the customer data received with this message by sending a corresponding message "get customer data" to the subject "customer data management". This subject sends the complete customer data back to the subject "help desk" via the message "customer data". The subject "help desk" then checks the customer data. If the data are invalid a message "invalid customer data" is sent to the subject "caller" and the process is finished.

If the customer data are valid, the subject "help desk" creates a trouble ticket with this data which is sent to the subject "ticket management" via the message "store ticket". After that, the message "towing service requested" is sent to the "help agent" that organizes the towing service. The part of the communication structure of the subject "help agent" for organizing the towing service is not shown in Figure 4.13. We only see that the subject "help agent" sends the message "towing service ordered" to the subject "help desk". This message contains all the data about the

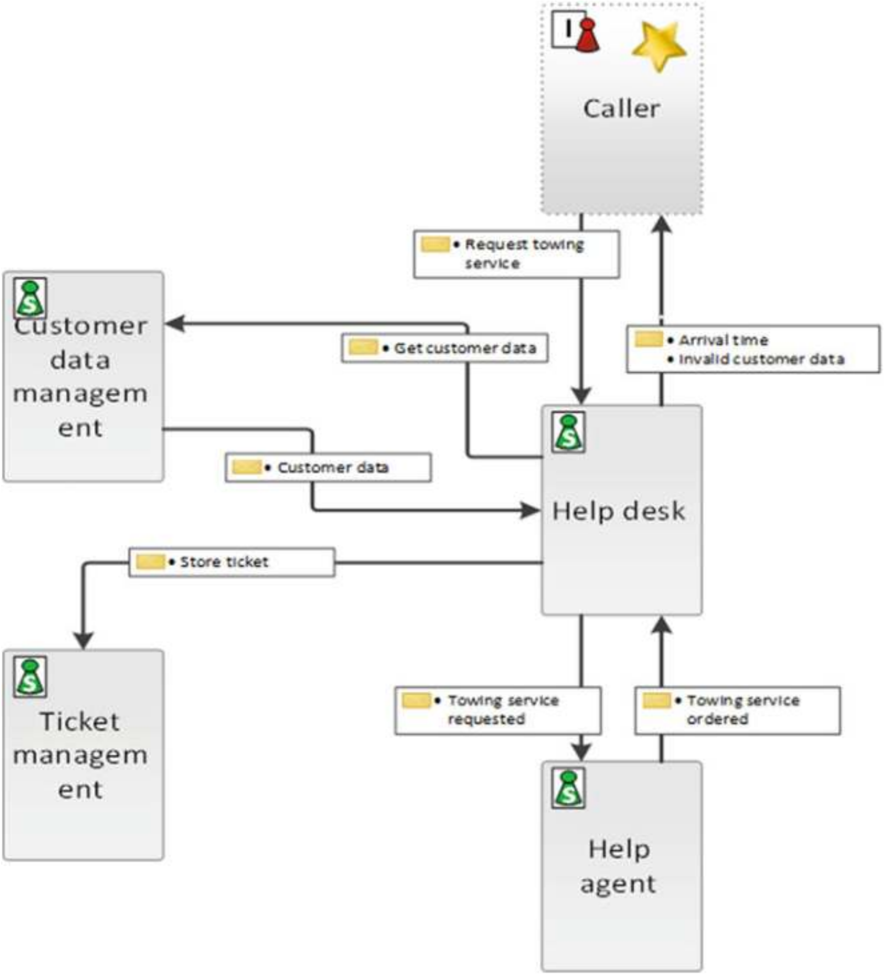


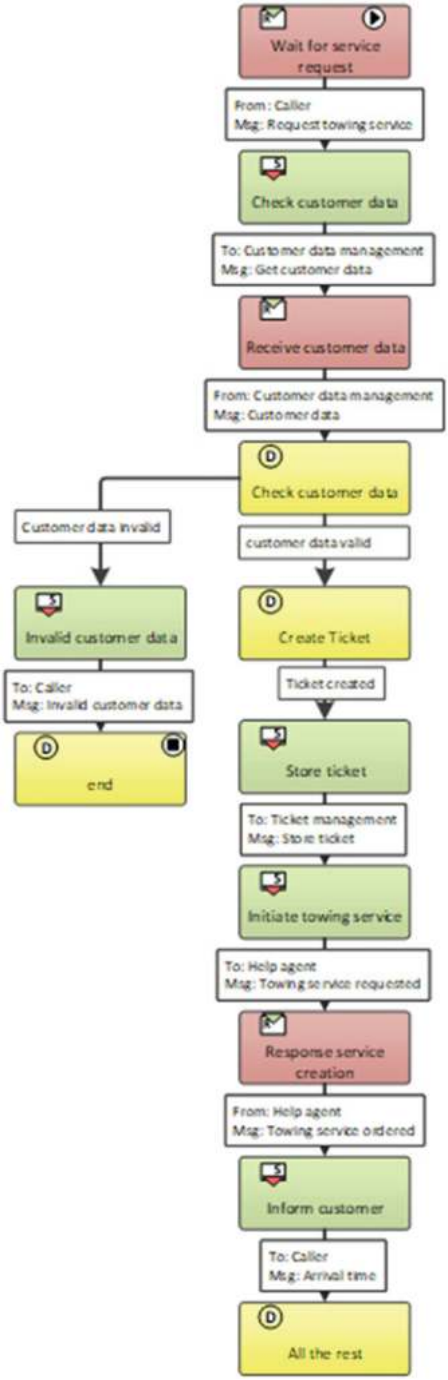
Figure 4.12: Subject interactions of the subject "help desk"

service, e.g., name of the towing company and arrival time. The subject "help desk" forwards this data to the interface subject "caller".

The behavior described in Figure 4.13 contains the communication of the subject "help desk" with all neighbor subjects, including the communication with the interface subject "caller". From the perspective of this subject, the communication of the subject "help desk" with its other neighbor subjects is not relevant. For the subject "caller" only the communication sequence between itself and the subject "help desk" is relevant. These allowed communication sequences are called the behavioral interface.

The behavioral interface between two subjects can be derived from the complete behavior of one of these subjects by deleting the interactions with all the other

Figure 4.13: Part of the behavior diagram of the subject "help desk"



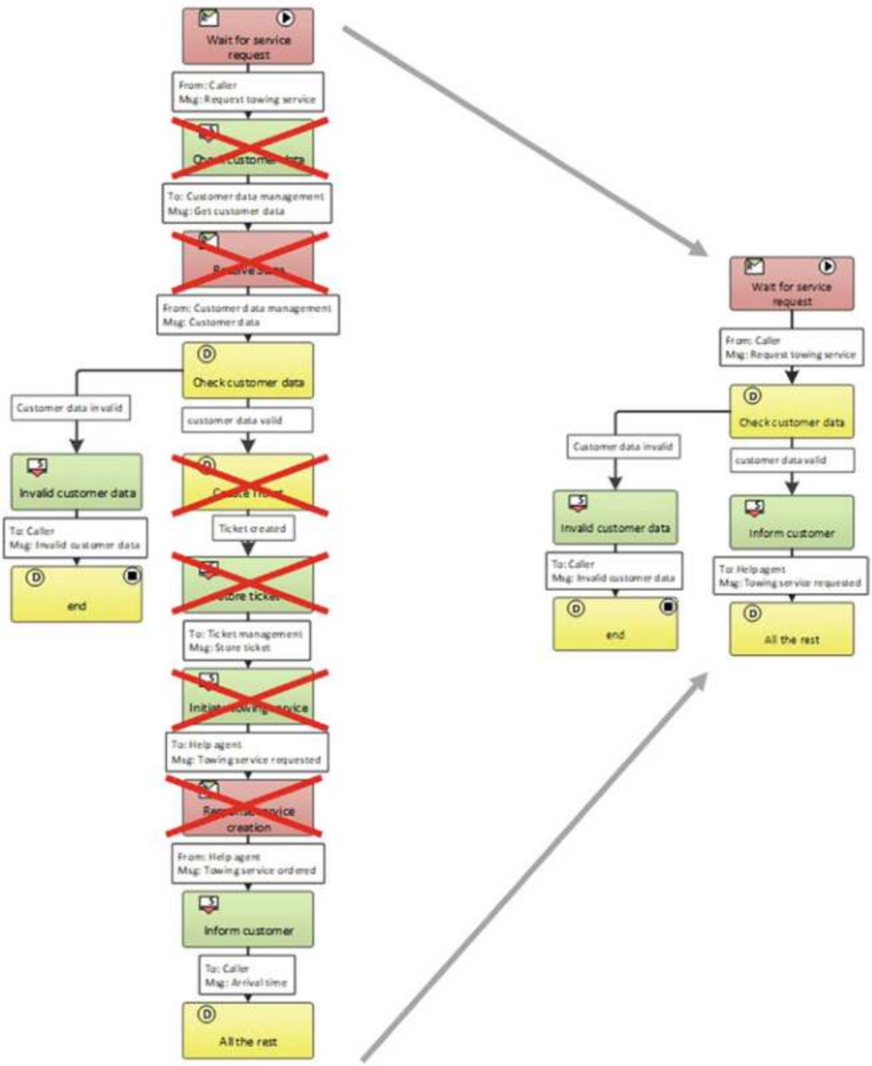


Figure 4.14. Sample behavioral interface

subjects [3]. Figure 4.14 shows how the communication sequence relevant for the communication between the subjects “help desk” and “caller” is derived from the complete behavior of the subject “help desk”.

4.2 Readiness for Digitalization

In this section we investigate to what extent process models are specified in a precise syntactic and semantic notation in the different modeling languages to enable generation of digital workflow support automatically. Readiness for digitalization denotes the capability to support semantically rich process specifications in an accurate way, allowing for a corresponding automated execution of process models. We shed light from this perspective on some notational schemes before demonstrating for subject-oriented representations how behavior models could be handled from a well-defined semantic representation and processing perspective.

4.2.1 Readiness for Digitalization of Flowcharts

The standard for flowcharts does not contain precise syntax and semantic descriptions. Due to their long history and widespread use, the importance of flowcharts is more or less common knowledge. Flowcharts have no underlying data model to share data between various diagrammatic editing tools for flowcharts, databases or other programs, such as project management systems or spreadsheets. There exist many applications and visual programming languages that use flowcharts to represent and execute programs [4]. However, these tools focus on programming, and not on business processes.

4.2.2 Readiness for Digitalization of Event-Driven Process Chains

There are many tools for creating EPC-based process models. Up to now there is no standardized way for storing these models. This means models created, for example, with tool A cannot be used in tool B. The tool most commonly used for creating EPC-based process models is ARIS. ARIS uses its own data model. Hence, process descriptions produced with ARIS cannot be processed with another tool. There have been some research activities to define a general data model [5] and also some research projects with respect to the direct execution of EPC models [6]. In practice, however, only the ARIS tool suite is used for creating EPC specifications and the authors are not aware of any projects in which EPCs are executed directly.

4.2.3 Readiness for Digitalization of UML Activity Diagrams

There are many UML-based tools - a list of them can be found in Wikipedia [7] - which support the creation of activity diagrams. The OMG XMI standard specifies a structure that uses XML for interchanging models between various tools. Although in principle, this standard allows the handling of process models with different tools, in practice the transformation of model descriptions between various tools can be cumbersome: Many tool manufacturers create variations of the standard in different ways, e.g., they do not support some notational elements or they add other notational elements.

Most UML tools support the generation of code for several target languages. A special code generation for activity diagrams has been proposed in existing research [8]. This code generation targets real time systems and it still needs to be investigated as to what extent it can be used for business processes. The authors could currently not find any UML-based tool suite which is recommended for implementing business processes.

4.2.4 Readiness for Digitalization of BPMN

The BPMN standard contains an XML data model which allows the processing of a process model by different tools. Since in practice each tool vendor places its focus on different aspects of BPMN and interprets the standard in a special way, the transfer of a process model from one tool to another can be tedious [9].

In the standard's documentation the execution semantics is described in natural language. At the Software Competence Center in Hagenberg, Austria a formal semantic for the process diagrams of BPMN has been defined [10]. For the formal description of the BPMN semantic, the Abstract State Machine formalism has been used [11, 12]. In this project, several ambiguities, inconsistencies and gaps in BPMN have been identified [10].

There are many tools which also support the execution of BPMN processes. However, the majority of these tools do not fully implement the BPMN standard. Most tools support a limited set of execution elements and do not interpret them in fully compatible ways, leading to partially differing execution outcomes for identical process models [13].

4.2.5 Readiness for Digitalization in Subject-Oriented Process Specifications

For the S-BPM language, which is based on PASS [1], a standard ontology in OWL [14] has been defined [15]. This ontology allows a PASS specification to be processed with different tools, if these tools follow the standard ontology.

Each subject has a base behavior and may have additional subject behaviors for macros and guards. All these behaviors are subclasses of the class SubjectBehavior. The details of these behaviors are defined as state transition diagrams (PASS behavior diagrams). These behavior diagrams are represented in the ontology with the class BehaviorDescribingComponent (see Figure 4.15). The behavior diagrams have the relation "BelongsTo" to the class SubjectBehavior. The other classes are needed for embedding subjects into the Subject Interaction Diagram (SID) of a S-BPM model (see section 3.6).

The following figure shows the details of the class BehaviorDescribingComponent. This class has the subclasses State, Transition and TransitionCondition. The subclasses of the state represent the various types of states (class relations 025, 014 und 024 in the

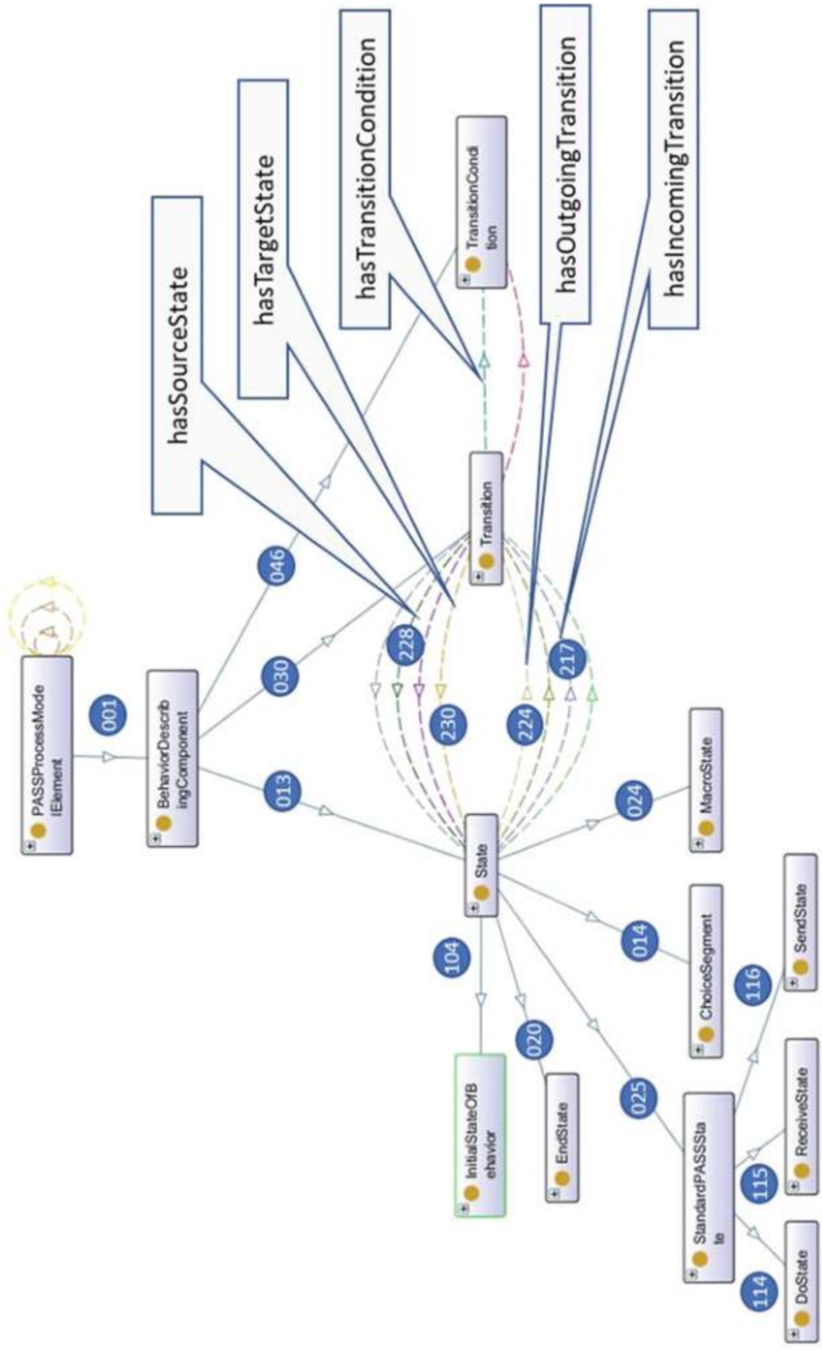


Figure 4.15: Class BehaviorDescribingComponent and its subclasses and properties

```
Behavior_subj (D) = {Behavior(subj, node) | node ∈ Node(D)}
Behavior(subj, state) =
    if SID_state(subj) = state then
        if Completed(subj, service(state), state) then
            let edge = select_Edge({e ∈ OutEdge(state) | ExitCond(e)(subj, state)})
                Proceed(subj, service(target(edge)), target(edge))
        else Perform(subj, service(state), state)
    where
        Proceed(subj, X, node) =
            SID_state(subj) := node
            Start(subj, X, node)
```

Figure 4.16: Main ASM functions of the interpreter

figure above). The standard states “DoState”, “SendState” and “ReceiveState” are subclasses of the class “StandardPASSState” (subclass relations 114, 115 und 116). The subclass relations 104 and 020 allow a start state (class “InitialStatOfBehavior”) and none or several end states (see subclass relation 020). The fact that there must be at least one start state and none or several end states is defined by so-called axioms which are not shown in the figure above.

States can be start and/or end points of transitions (see properties 228 and 230). This means a state may have outgoing and/or incoming transitions (see properties 224 and 217). Each transition is controlled by a transition condition which must be true before a behavior follows a transition from the source state to the target state.

The ontology defines only the structure of a process description. The dynamic aspect is not covered yet. The execution semantic of S-BPM models is described with Abstract State Machines (ASM) [16]. The ASM defines the algorithm of an interpreter that will “crawl” through a Subject Behavior Diagram (SBD) of a process model defined in the, not explicitly named, Subject-oriented Process Modeling language PASS as defined in the previous section 3.

Figure 4.16 shows the ASM-code for the interpretation of the behavior specification [17].

The behavior of a single subject *subj* is specified by the *Behaviorsubj (D)* rule, which takes the Subject Behavior Diagram *D* as a parameter. From there on the *Behavior(subj, node)* rule defines how a single node behaves. As long as the service of that node is not *completed* the Perform rule will be called, which is refined for all given services *X*. Once the node is completed the outgoing transition will be determined by $select_{Edge}$, the *Proceed* rule updates the current *SID_state* and initializes the new node with the *Start* rule, which also is refined for all services *X*.

The following table shows the relationship between the ASM interpreter specification and the classes and properties of the ontology.

The meaning of the colors is as following:

- OWL classes (brass coloured)
- Object properties (blue)

Interpreter Spec	Description	Corresponding OWL-Model Element
SID_state	Execution concept – no model representation, not to be confused by a model "state" in an SBD Diagram. State in the SBD diagram define possible SID States.	X - Execution concept – the state the subject is currently in as defined by a **State** in the model
D	A **D**iagram that is a completely connected SBD	**SubjectBehavior** – under the assumption that it is complete and sound.
node	A specific element of diagram D - Every node 1:1 to state	**State**
state	The current active state of a diagram determined by the *nodes* of Diagram D	**State**
initial state	The interpreter expects and SBD Graph D to contain exactly one *initial (start)* state and at least one *end state*	**InitialStateOfBehavior**
end state		**EndState**
edge / outEdge	"Passive Element" of an edge in an SBD-graph	**Transition**
ExitCondition	Static Concept that represents a Data condition	**TransitionCondition**
subj	Identifier for a specific Subject Carrier that may be responsible for multiple Subjects	Execution Concept – ID of a Subject Carrier responsible possible multiple Instances of according to specific **SubjectBehavior**
ExternalSubject	A representation of a service execution entity outside of the boundaries of the interpreter (The PASS-OWL Standardization community decided on the new Term of Interface Subject to replace the often-misleading older term of External Subject)	Represented in the model with **InterfaceSubject**
subject-SBD / $SBD_{subject}$	Names for completely connected graphs / diagrams representing SBDs	**SubjectBehavior** or rather **SubjectBaseBehavior** as **MacroBehaviors** and **GuardBehaviors**
service(state) / service(node)	Rule/Function that reads/returns the service of function of a given state/node:	Object Property: **hasFunctionSpecification** (linking **State,** and **FunctionSpecification** --> (**State hasFunctionSpecification FunctionSpecification**)

Interpreter Spec	Description	Corresponding OWL-Model Element
function state	The ASM spec does not itself contain these terms. The description text, however, uses them to describe states with an according service, e.g., a state in which a (*ComAct = Send)* service is executed is referred to as a *send state* Seen from the other side: a **SendState** is a state with *service(state) = Send)* Both send and receive services are a *ComAct* service. The *ComAct* service is used to define common rules of these communication services.	**DoState**
send state		**SendState**
receive state		**ReceiveState**
ComAct	Specialized version of Perform-ASM Rule for communication, either send or receive. These rules distinguish internally between send and receive.	**CommunicationAct**s with subclasses (**ReceiveFunction SendFunction)** **DefaultFunctionReceive1_EnvironmentChoice** **DefaultFunctionReceive2_AutoReceiveEarliest** **DefaultFunctionSend**

The interpreter ASM Spec has the following main function or rules that are being executed while interpreted.

- BEHAVIOR(*subj,state*),
- PROCEED(*subj,service(state),state*),
- PERFORM(*subj,service(state),state*)
- START (*subj,X, node)*

The following table shows the relationships between the ASM main functions and the classes of the OWL model elements.

Interpreter Spec	Description	Corresponding OWL-Model Element
BEHAVIOR(subj;state)	Main interpreter ASM-rule/Method	Execution concept
BEHAVIOR(subj;node)	ASM-Rule to interpret a specific node of Diagram D for a specific subject	Execution concept
Behaviorsubj (D)	Set of all ASM rules to interprete all nodes/states in a SBD(iagram) D for a given subj (set of all *BEHAVIOR(subj;node)*)	Execution concept
PERFORM(subj ; service(state); state)	The main Perform ASM Rule/Method that prompts a PASS interpreter to execute functions defined for states	**State** **hasFunctionSpecification** **FunctionSpecification** Specialized in: **DoFunction** and. **CommunicationActs** with **ReceiveFunction** **SendFunction** There exist a few default activities: **DefaultFunctionDo1_EnvoironmentChoice** **DefaultFunctionDo2_AutomaticEvaluation**
PERFORM(subj ;CommAct; state)	ASM-Rule specifying the execution of a Communication act in an according state)	**CommunicationActs** with **ReceiveFunction** **SendFunction** **DefaultFunctionReceive1_EnvironmentChoice** **DefaultFunctionReceive2_AutoReceiveEarliest** **DefaultFunctionSend**

There are some prototypes of modeling tools which follow the standard ontology of PASS and a prototype of a workflow engine which interprets the standard ontology [18].

References

1. Fleischmann, A. (1994). *Distributed systems: Software design and implementation*. Berlin: Springer.
2. Meyer, N., Thomas, F., Radmayr, M., Blei, D., & Fleischmann, A. (2010). Dynamic creation and execution of cross organisational business processes- The jCPEX! approach. In *S-BPM-ONE 2010; CCIS 138*. Heidelberg: Springer.
3. Flowgorithm. Flowgorithm, September 2019. [Online]. Last accessed September 2019, from http://www.flowgorithm.org/
4. Myers, B. A. (1986). Visual programming, programming by example, and program visualization: A taxonomy. *ACM SIGCHI Bulletin, 17*(4), 59–66.

5. Nüüttgens, M., & Mendling, J. (2004). Transformation of ARIS XE "ARIS" markup language to EPML. Proceedings of the 3rd GI Workshop on Event-Driven Process Chains (EPK 2004), Luxembourg.
6. Kopp, O. (2005). Abbildung von EPKs nach BPEL anhand des Prozessmodellierungswerkzeugs Nautilus, Stuttgart: Diploma Thesis, University of Stuttgart.
7. Wikipedia [Online]. Last accessed October 2019, from https://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools
8. Backhauß, S. (2016). Code generation for UML XE "UML" activity diagrams in real-time system. Hamburg: Master Thesis; Hamburg University of Technology.
9. Dirndorfer, M., Fischer, H., & Sneed, S. (2013). Case study on the interoperability of business process management software. BPM ONE-Running Processes, CCIS 360. Springer.
10. Kossak, F. et al.. (2014). A rigorous semantics for BPMN 2.0 process diagrams. Springer.
11. Börger, E. (2003). *Abstract state machines: A method for high-level system design and analysis*. Heidelberg: Springer.
12. Elstermann, M. (2017). Proposal for using semantic technologies as a means to store and exchange subject-oriented process models. Darmstadt, Germany.
13. Geiger, M., Harrer, S., & Wirtz, G. (2018). BPMN 2.0: The state of support and implementation. Future Generation Computer Systems. pp. 250-262.
14. McGuiness, D. L., & van Harmelen F. (2004). OWL XE "OWL" web ontology language overview. *W3C recommendation, 10*(10).
15. Elstermann, M., & Krenn, F. (2018). *The semantic exchange standard for subject-oriented process models*. Linz: ACM.
16. Börger, E., & Raschke, A. (2018). *Modeling companion for software practitioners*. Springer.
17. Börger, E. (2012). A precise description of the S-BPM modeling method. In *Subject-oriented business process management*. Heidelberg: Springer.
18. Fleischmann, A., Borgert, S., Krenn, F., Singer, R., & Elstermann, M. (2018). An overview to S-BPM oriented tool suites. S-BPM-ONE 2018. ACM Digital Library

5 From Modeling To Digitalization

5.1 Overall Context

In the previous chapters, we have shown that what happens in organizations (companies, administrations, etc.) is based on models from various disciplines. Business models, which represent enterprise architectures with models for products and services, organizational structure, processes, data and IT infrastructure, describe in which area a company does business, how it does this, which exchange relationships it has with partners, which technical infrastructure it is supported by, etc.

In chapter 3, we have focused on approaches and notations for the specification of business processes and thus the design and representation of operational processes. In the course of digitalization, these processes must be augmented, as far as possible and economically sensible, with information and communication technology. Both the appropriate incremental improvements of existing processes, as well as fundamental process innovations, are based on creative design accomplishments, which should lead from process models to executable systems. In this chapter we will therefore first deal with the concept and typical activities of Business Process Management. With the approach of Design Thinking, we then illuminate a methodical approach to creatively produce something new and solve complex problems. Subsequently, we put the two concepts into relationship with one another.

5.2 Activity Bundles in Business Process Management

5.2.1 Overview

In chapter 1 we have already mentioned that the design of business processes up to their execution as instances in the processing of concrete business transactions ("operational business") itself represents a process. This is often understood as a

A. Fleischmann et al., *Contextual Process Digitalization*,
https://doi.org/10.1007/978-3-030-38300-8_5

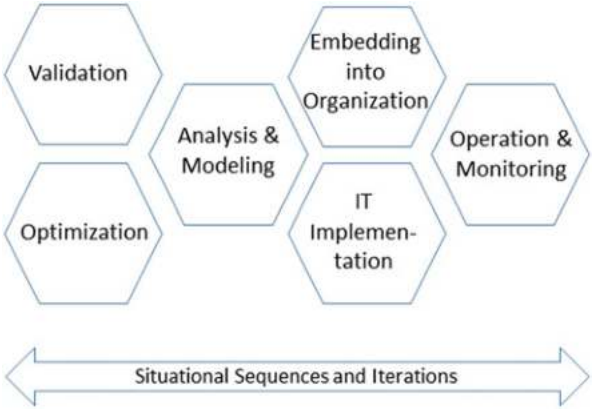


Figure 5.1: Activity bundles in Business Process Management

Business Process Management cycle with phases such as strategy, design, implementation and controlling [1].

In practice, however, the sub-activities are often not clearly distinguishable from each other. We therefore see them less as a circle with a sequential sequence, but rather as networked and interwoven, as the honeycomb structure in Figure 5.1 suggests. The diagram also shows that we differentiate the tasks somewhat more and identify them as bundles of activities: analysis and modeling, validation, optimization, embedding into organization, IT implementation and execution and monitoring.

Although the usual representation as a cycle suggests that in process management projects all activity bundles are run through as a sequence, their selection and sequence depend on the concrete situation, e.g., the maturity level of a process. The sections 5.2.2 to 5.2.7 explain the activities using an example process. Here, the steps are first run through completely and also in the specified sequence. Such a scenario is realistic, for example, when a process is designed for the first time or completely reorganized. In section 5.2.8, we then discuss several scenarios for improvements that can be derived from the experience gained during operation of the originally designed process environment. They illustrate the situationally different paths through the activity bundles in the further development of the process.

5.2.2 Analysis and Modeling

The analysis serves to gather information about why a process exists or should be implemented, which goals an organization pursues with it within the framework of its strategy, and how it is currently working. The objectives are the documentation and the acquisition of indications for improvements. The modeling uses, among other things, the results of the analysis and deals with the design of future working methods, i.e., process changes and innovations. If further information is required, the participants switch back to the analysis mode to collect it and then act again in a creative manner. Therefore, analysis and modeling cannot be clearly distinguished from each other. Validation and optimization also usually take place here, when the participants develop the model iteratively to the best of their knowledge and belief, taking into account the weak points identified in the analysis and trying out and discussing possible solutions.

In addition to considering determining factors such as strategic significance, objectives and risks, analysis and modeling are essentially concerned with analyzing or specifying (see also section 1.3)

- which actors (e.g., people, machines),
- perform which activities,
- according to which business rules,
- on which business objects (for example, information linked to certain carriers, physical objects),
- using which tools (e.g., IT systems) and
- how they interact in order to achieve the desired process goals and results.

For the development of process models based on these findings, the modeling languages presented in chapter 3 with the corresponding graphical notations are used.

During analysis and modeling, usually also the ground is laid for operational process controlling in the operating phase. In addition to the process attributes already mentioned, performance parameters (indicators), in particular Process Performance Indicators (PPIs), are defined, systematized in a measurement system and provided with target values [2, p. 265].Typical examples of PPIs are lead time, output per time unit, error rate, customer satisfaction, etc. The PPIs and the target values planned for them form the basis for business process monitoring, that is, operational process control during execution (see sections 5.2.7 and 7.3.3).

Analysis and Modeling in a Case Study

As a case study, we use the strongly simplified process of credit application processing in a bank. There, applications are received from interested parties for the granting of a real estate loan. Before preparing an offer, clerks check the creditworthiness of the respective customer and the value of the property to be loaned on. If the result of both checks is positive, the clerk prepares an offer with data such as loan amount, interest rate, repayment rate and term. If the loan amount is less than €200,000, he signs the offer and sends it to the customer. Otherwise, he must first obtain the approval of his department head and, in the case of more than €500,000, that of the executive board. If the creditworthiness check or the object check reveal any risks, a clerk contacts the interested party to agree on further procedures, such as reducing the loan amount. As part of an analysis of the process, this information was collected and structured (see Table 5.1). Together with additional information, this collected information serves as a basis for modeling the process.

Table 5.1: Characteristics of the example process and information obtained for it

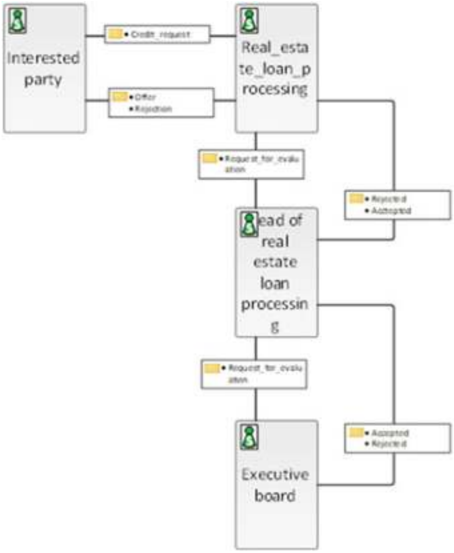
Process characteristic	Result of analysis
Actors / Roles:	Interested party, real estate loan processing, head of real estate loan department, executive board
Activities:	Apply for credit, check credit application (completeness), check customer creditworthiness, estimate value of object to be financed, determine financing conditions, Create offer, approve offer, send offer
Business objects:	Credit application with attachments, credit offer
Business rules:	Credit offers over €200,000 must be approved by the department management, over €500,000 by the board.
Interactions:	Interested party - real estate loan processing (incoming mail), real estate loan processing - head of real estate loan department, head of real estate loan department - executive board, real estate loan processing – interested party
Tools:	Internet portal of the bank (web form), backend system of bank, workflow system of bank, credit office web form, e-mail, telephone
PPIs:	Observation of the behavior of instances using: • Processing time from the receipt of a credit application to the dispatch of an offer (target: average max. 3 days), • Frequency per week including distribution • Rejection rate of applications by the bank • Rate of rejection of offers by interested parties

In the case at hand we use the modeling language S-BPM described in Chapter 3 for the representation of the process, since it exists in a strongly interaction-oriented description. Generally speaking, the representation would also be possible in any other modeling language which allows the depiction of responsibilities (e.g., eEPC or BPMN)

The following figures show an excerpt of the associated process model using S-BPM. Figure 5.2 shows the subjects (agents) involved and the messages they exchange during process flow.

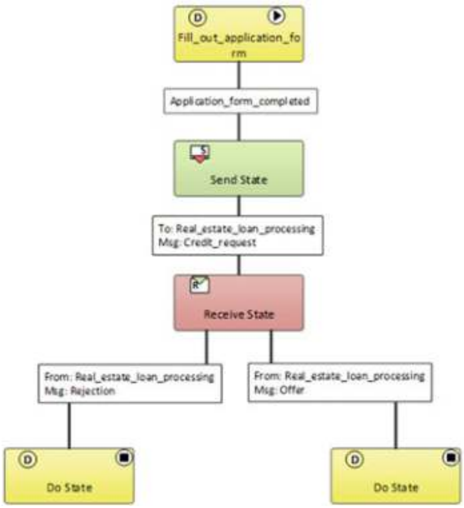
The communication structure (Subject Interaction Diagram, SID) does not yet contain a sequence in which the respective messages are exchanged. The sequences are described in the behavior of the subjects. Figures 5.3 and 5.4 show the behavior of the subject "interested party" and a behavior excerpt of the "real estate loan processing" subject.

The behavior of the subjects "Head of real estate loan department" and "Executive board" is described analogously. They would be involved in the process if the desired loan amount exceeded 200,000 or 500,000 euros. These cases are not modeled in Figure 5.4 – they would be represented by additional branches after the state "approval required".



The picture shows the communication structure for applying for a loan. The communication is initiated by the subject "interested party". This subject sends the message "credit request" to the subject "real estate loan processing" and receives the messages "offer" or "rejection". According to the process requirements the other subjects exchange necessary messages. The communication structure does not yet imply a sequence in which the messages are exchanged.

Figure 5.2: Communication structure of the actors in the credit request process

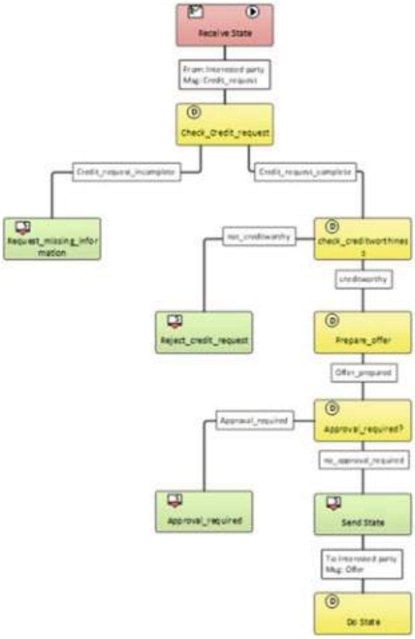


The picture shows the behaviour of the subject "interested party". A credit request is formulated in the initial state (marked by a triangle in the circle). Usually this is done by filling out a corresponding form. The contents of the form would be part of the specification of the business object "credit request". After the data object "credit request" has been created, it is sent to the subject "Real estate loan processing". Then the subject "interested party" waits for an answer. This can be either the message "Offer" or "Rejection". In both cases the subject's behaviour is terminated by reaching the respective final states which are marked as such by dots.

Figure 5.3: Behavior of the subject "interested party"

5.2.3 Validation

In the BPM context, validation means checking whether the designed process generates the output expected by the (external) customer and process owner, for



The picture shows an excerpt of the behavioral description of the subject "Real estate loan processing". This excerpt shows only the positive case. In the initial state (state marked with a triangle in a circle), the subject receives the message "Credit request" from the subject "interested party". With this message the subject receives as a payload the business object with the credit application data. In the next state it is checked whether these data are complete. If not a corresponding query message is sent to the subject "interested party". This branch (left part of the picture) is not further described in the behavior description. If the data are complete the creditworthiness of the interested party is examined. If the interested party is creditworthy an offer is provided. If the desired credit is less than 200 thousand euros, the message "Offer" is sent to the subject interested party.

Figure 5.4: Behavior of the subject "real estate loan processing"

example in the form of a service or product. This question with regards to effectiveness already refers to the results of partial steps, i.e., process participants of the own organization as customers. For example, it is necessary to evaluate whether an upstream process step provides all the information that a processor requires for a decision (for instance, approval) in his or her subtask. We have already mentioned that parts of a model are repeatedly subjected to validation, even during its step-by-step development. In addition, the object of validation is the completely finished process model, the effectiveness of which should be ensured before it is implemented in terms of Information Technology (cf., section 5.2.6). Otherwise, errors are discovered too late and lead to correspondingly high costs for their elimination.

Validation in the Case Study

The process model for the example process was validated during its development as well as at the end. In doing so, it was initially discovered that the original loan application did not include a field for the applicant's employment status and thus lacked an important risk assessment factor. This led to the extension of the business object and to a positive validation result in the corresponding iteration.

5.2.4 Optimization

While validation aims at ensuring the effectiveness of business processes, optimization is about efficiency. Process efficiency can be expressed by process attributes for resource consumption such as duration and costs. Optimization means finding the optimal design of a process with regard to such process parameters. Essential starting points are improvements in operational and structural organizational design as well as IT support. Strictly speaking, optimization is not an independent activity bundle, but makes use of modeling, organizational implementation, and IT implementation (see sections 5.2.2, 5.2.5, and 5.2.6).

Simulation is a well-known method for comparing alternatives in process execution or resource allocation. It can be used to obtain quantitative information on the development of process parameters for a large number of process instances (orders, production pieces, etc.). The simulation enables the evaluation of a process model with a certain combination of parameters. These can be deterministic or stochastic quantities described by probability distributions. Through the use of parameter changes and alternative process designs, different design options can be analyzed with respect to their behavior. This allows insights to be gained into bottlenecks or inefficiencies and the sensitivity of parameters. The extension of a process model and the gathering of necessary information for conducting simulations can cause considerable effort. Attributes relevant for optimization are often interdependent and contradictory, which makes optimization difficult and requires balancing efforts. A process alternative can, for instance, have a shorter lead time relative to another, but cause higher costs. The decision for an alternative therefore also depends on the priority of the process objectives.

Optimization in the Case Study

The model could already be optimized during its design. The steps for checking customer creditworthiness and value of the property, which were initially sequentially planned, were redesigned to be executed in parallel.

5.2.5 Embedding into an Organizational Context

For productive operation, validated processes need to be embedded into the existing, redesigned or newly created organizational environment. This is also referred to as the organizational implementation of a process. It quite often requires an adaptation of the surrounding operational and organizational structure.

A single process is usually part of an entire value creation environment (value chain, value creation network) into which it must be seamlessly assimilated. Therefore, with regard to the operational integration into the process map, particularly the interfaces with other processes must be considered. This can lead to changes to be carried out at the interfaces of an upstream or downstream process. Such circumstances usually are already taken into account in the upstream activity bundles. Therefore, the implementation should be limited to the chronological coordination of the go-live procedure. This means that processes that are connected via interfaces must go live again at the same time if a change has occurred at an interface that has also made modifications necessary in the partner process.

The organizational embedding comprises the assignment of concrete actors, i.e., people as job or role holders, to the actors abstractly specified in the model. One of the challenges is the consideration of the organizational context when using workflow engines. These must be able to dissolve, for instance, dynamic surrogate regulations at runtime, as well as the fact that persons can assume different roles in the same process. For example, a superior in a vacation request process can be the approver of vacation requests for his own employees, but can also be the applicant for his own vacation, which in turn must be approved by his own superior. Therefore, the software must have organizational knowledge, facilitating the correct routing of a process instance through the processing units and steps.

Further qualitative and quantitative aspects have to be taken into account during organizational embedding. Care must be taken to ensure that the employees have the necessary qualifications (skills) to carry out the modeled behavior or can gain them through training. Adequate qualification is not only a prerequisite for successful work in the currently valid version of the process; it can also foster improvement proposals by process participants.

The number of people assigned to the abstract actors in the model influences the capacity for processing process instances and thus affects parameters such as lead time.

Embedding into an Organizational Context in the Case Study

Column 2 in table 5.2 shows the number of employees who in general are qualified to be assigned to the identified and modeled roles. The third column contains the actual capacity used (short-term absences, e.g., due to illness, are not taken into account). The heads of real estate and consumer credit departments stand in for each other, both in disciplinary and domain-specific matters. This also applies to the members of the executive board.

Table 5.2: Potential and concrete assignment of roles

Actor/Role	Total number	Assigned according to personnel deployment plan
Real estate loan processing	9	5 full-time employees
Head of real estate loan department	1	1 manager (Head of consumer credit department) 1 deputy
Executive board	3	1 head of private customers division 1 head of business customers department 1 head of investment management department

5.2.6 IT Implementation

Most processes cannot be carried out economically without IT support. Especially when a high degree of automation is strived for, the quality of the mapping of the process in IT becomes very important. But also, and in particular, for steps where human actors are involved (e.g., entering data, making decisions), the user-centered/friendly design of the IT systems is of high importance.

IT-related implementation of a process means mapping it as an IT-supported workflow with integration of a suitable user interface, the execution logic and the IT systems involved. For this purpose, it is necessary first of all to transfer the more or less formal model description (see chapter 3) into a language interpretable by a workflow engine, i.e., into an executable program. This enables the engine to control the execution of a process instance at runtime according to the model. For the completion of individual subtasks during processing, a whole series of software applications and services usually have to be integrated into the process. Typical examples are ERP transactions and document and content management systems.

A relatively new approach to automation of standardized, repetitive work procedures is Robotic Process Automation (RPA). This term stands for tools that "perform [if, then, else] statements on structured data, typically using a combination of user interface interactions, or by connecting to APIs to drive client servers, mainframes or HTML code" [13]. Thus, software robots for example imitate the behavior of humans when using the graphical user interface of information systems [14]. This allows, for instance, the quick linking of heterogeneous IT systems and automated data transfer between, or data input in, various existing applications without the need for these to be modified. Artificial Intelligence and machine learning functionality promises to facilitate learning and automated adaptation of RPA tools to changes in the underlying IT systems [15].

Extensive testing of the implemented overall solutions must ensure the quality of the process support provided by IT.

IT Implementation in the Case Study
Table 5.3 shows the essential elements of the IT environment, which was designed to support the process and its sub-steps, together with their most important process-relevant functions.

5.2.7 Operation and Monitoring

Implemented processes go live after their approval by the responsible authorities. This means that those involved in a process execute it in the form of instances in the organizational and IT environment set up for day-to-day business.

In order to obtain information for the deliberate management of processes, it is necessary to observe their behavior during everyday operations. This monitoring records measurement data and calculates actual values for the Key Performance Indicators defined during analysis and modeling. An immediate comparison with

Table 5.3: Realized IT environment of the process

IT system/service	Selection of functions that are essential for the process
Portal of the bank	• Provides information material on financing and an electronic form for the customer or a real estate loan officer to fill out the loan application. • . . .
Workflow engine	• Instantiates the process when the customer saves the request in the portal. • Controls instances according to the model, including users and other systems or services as needed. • Records log data for the operations. • Generates messages and reports based on log data. • . . .
Backend system of the bank	• Manages customers • Categorizes customers (scoring) • Determines conditions • Generates offers • . . .

defined target values leads to escalations along the management hierarchy and, if necessary, to short-term measures in the event of deviations. Medium- and longer-term evaluations reveal structural opportunities for improvement. The analysis of the process behavior and possible deviations allow conclusions to be drawn about causes and triggers feedback into other activity bundles.

Operation and Monitoring in the Case Study

Since the release of the process, the bank has been processing credit applications from interested parties in the described form and environment. Monitoring for the past quarter revealed the following *average* figures:

- Interested parties had submitted 50 applications per week.
- The bank rejected 20% of them, half of it due to lack of creditworthiness.
- For the remaining applications the interested parties received an offer within 4 days.
- In 30% of the cases the interested party accepted the offer and signed a contract.

Since competitors advertise with very short processing times, the bank assumes that the 4 days until applicants receive an offer, all other conditions being equal, is one of the reasons why customers do not sign a contract. This duration also deviates significantly from the previously formulated target of 3 days.

5.2.8 Optimization Scenarios in the Case Study

The following scenarios show how further analyses of the monitoring results can be used to investigate the causes of the long lead time and to branch out into suitable activity bundles for improvement measures (optimization).

In each case, the target point of the branch-out determines the further path through the activity bundles, i.e., which subsequent activities are necessary before the redesigned process can be put into day-to-day operation. In the interests of simplification, we limit our consideration to one measure per scenario. In reality, several optimization possibilities will usually be pursued in parallel.

Optimization Scenario 1

The frequency distribution for credit application occurrence has shown that on Mondays 25, Tuesdays 15, Wednesdays 6, Thursdays 2 and Fridays also 2 applications are submitted. This could be due to the fact that interested parties view real estate, take purchasing decisions and think about financing mainly on weekends. The analysis of the dormancy period until the real estate loan department processes a request reveals a bottleneck at the beginning of the week, due to the high number of parallel applications. With the currently available capacity of 5 full-time clerks, the average dormancy period is 2 days. In order to reduce the latter and thus also the overall lead time, additional processing capacity, such as available part-time staff, could be employed on Mondays and Tuesdays. In this case, only organizational implementation in terms of staffing is concerned; the process does not change and no further activities are necessary.

Optimization Scenario 2

A more detailed analysis has shown that the high average lead time is caused by the applications with amounts between €200,000 and €500,000, because the dormancy period until the department management approves the application is very high relative to the other proportions of the total duration. This is due to the fact that the availability of department heads and surrogates for approvals is limited, e.g., due to frequent business trips.

The bank's internal process analyst proposes a change of the business rule for approval. In the future, clerks should be allowed to sign and send offers for amounts up to €500,000 themselves. This reorganization affects several bundles of activities. First of all, it requires a change in the model, as the approval loop via the department management is no longer required. The model change requires subsequent validation to ensure that the changed process (still) leads to the desired result. As part of the IT implementation, the modification of the model must also be transferred to the workflow software and tested. The omission of the approval changes the task structure of the department management. For the clerks the tasks remain the same, but competence and responsibility increase. These changes have to be taken into account in the organizational implementation, for example through updated task descriptions and possibly through the qualification of the clerks through training, e.g., for a more comprehensive risk assessment. Compared to case 1, this scenario intervenes massively in the way the process is conducted and therefore requires much more extensive activities.

Optimization Scenario 3

The bank obtains creditworthiness data on the applicants from the associated credit bureau. For this purpose, the real estate loan processing clerks transfer the necessary customer data from the loan application to the credit bureau's web form. They then enter the results of this query into the banking system for further processing in the bank's own scoring system. The clerks give an account of the time-consuming copying of the data using copy & paste, the errors that occur in doing so, and the resulting reworking. In order to push forward digitalization, the bank decides to use the credit bureau's web service instead of their webform-based internet information service. The web service can be integrated into the workflow in such a way that the process engine triggers it when the clerk pushes a button and transfers the customer data as parameters. The service automatically returns the result to the process engine, which then transfers it to the banking system.

In this case, the only activity bundle to be dealt with is the IT implementation, including corresponding software adaptations and subsequent tests, before going live with the modified solution. The work procedures of the participants change only slightly; qualification measures are not necessary. The elimination of manual data transfer relieves them of mindless, time-consuming, and thus cost-intensive and error-prone routine tasks. The more intensive use of IT saves processing and lead time as well as costs, while increasing customer satisfaction.

Optimization Scenario 4

In section 5.2.4, we described that the customer creditworthiness check and the property value check were deliberately parallelized during modeling in order to save lead time. The analysis showed that the bank rejects five applications per week for creditworthiness reasons. In these cases, however, clerks had already spent effort doing the parallel value check of the real estate in question. Saving on this could initially speak in favor of first checking the creditworthiness and only carrying out the value check if the result is positive. A model change with validation and adaptation of the workflow application would be necessary.

However, the sequential order would again increase the lead time and lead to a conflict of objectives. Therefore, it is important to further consider whether the effort for value checking could be reduced through automation so that unnecessary value checks no longer play a role. It is conceivable, for example, that the banking system could be enhanced with valuation functions. It could then calculate a value index after automatic transfer of parameters from the loan application (type, size, year of construction, address, etc.) and being enriched with comparative information (values from the bank's own experience and reference value tables) and geo information (infrastructure with schools, shopping facilities, transportation connections, etc.). This index would accelerate the final value estimation carried out by the clerk. With this option, the parallel execution of the steps could remain. Instead of a model change, the additional functionality in the IT implementation would have to be realized and tested, and the clerks would have to be trained in how to use the software extension.

5.3 Introduction to Design Thinking

Design Thinking (DT) is a methodical approach to be creative and constructive in order to develop something new and to solve complex problems. It is characterized by innovative approaches to solution-oriented design. Problems can be better solved by focusing on the needs of the (potential) users during continuous iterations and "making comprehensible and graspable" through prototypes. This basic understanding of Design Thinking is shared by practitioners and scientists alike (for an overview of examples of definitions see [3]). The spectrum covered by the approach, on the other hand, is not seen uniformly.

There are, for example, interpretations which see it as a mind-set, as a process or as a toolbox [4]. An empirical investigation proves the perception in the continuum between the two poles of toolbox and mind-set (cf., Figure 5.5) [5].

On the one hand, this is due to the different roots, but on the other hand it is also a consequence of the inherent, constant, experience-led further development and adaptation of the concept in different contexts.

Larry Leifer, one of the protagonists of the approach at the d.school in Stanford states, that its permanent enhancements are an important part of Design Thinking, and that it would make itself unrecognizable if it were to publish a fixed manifesto one day [6].

The approach traces back to David Kelley from the design agency IDEO and professors Larry Leifer and Terry Winograd from Stanford University. In particular, the latter two recognized, when training engineering students, that the development of marketable products should focus much more on user-related aspects and less on purely technical aspects. This insight led to the development of the DT concept from the 1980s onwards and is still manifested today in the Stanford course on Mechanical Engineering 310 - Design Innovation (me310.stanford.edu). Hasso Plattner made a significant contribution to the further dissemination of this knowledge in research, academic teaching and business practice with his support of the institutions named after him at the d.school Institute of Design at Stanford University and the School of Design Thinking at Potsdam University (HPI D-School).

Due to its origin, DT was originally primarily concerned with the development of physical products. However, it is now used in a wide variety of areas, such as the development of services or entire business models, and is increasingly gaining in importance in organizational design and Business Process Management.

5.3.1 Core Elements

Core elements are a mixture of mindset, procedures and concrete facilities such as work areas. This is reflected in the rough division into the three "Ps", namely into the areas people, process and place.

Design Thinking begins with the creation of deep empathy for those affected by a (problem) situation. It identifies the optimal solution in the overlapping area of human desires (human-psychological aspect), feasibility (technological aspect) and

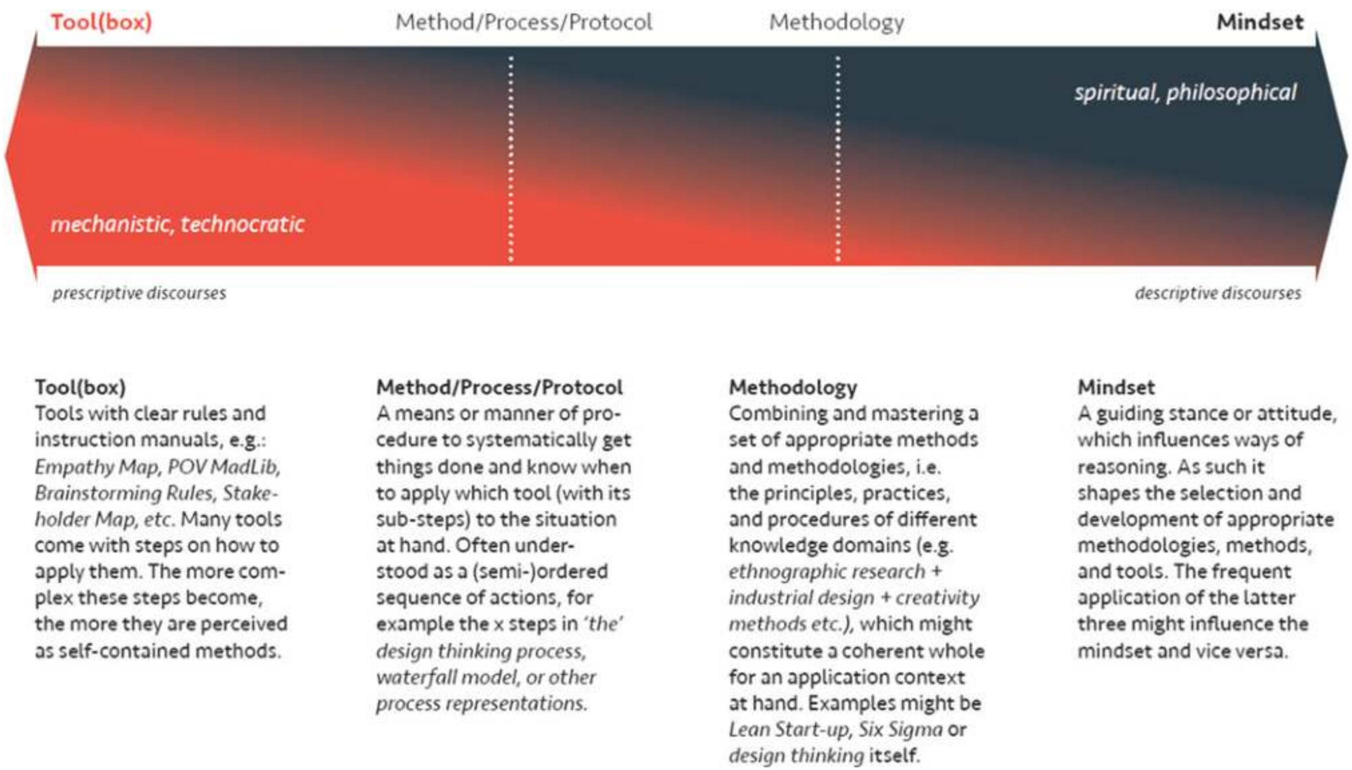


Figure 5.5: Understanding Design Thinking

profitability (business aspect) [7]. The innovation to be developed should be something,

- that people really like (desirability),
- that is feasible from a technological and process-related point of view (feasibility), and
- that is successful from an economic point of view (viability).

To achieve this, an interdisciplinary team (people) on variable, creativity-promoting premises (place) goes through a procedure (process) with many iterations, whereas a variety of methods can be used.

People

Focusing on the human being takes place in two respects:

On the one hand, representatives of the target group of the innovation, i.e., customers or users, with their needs are in the center of interest. Developing empathy for them, putting oneself in their position in the context under consideration, and thus gaining a deep understanding of the problem is the cornerstone for successful innovation and encompasses a large part of the process described in the "*Process*" subsection below.

On the other hand, Design Thinking strongly focuses on the people involved in the project as individuals and as a team. It aims for increasing the quality of results by using the diversity in interdisciplinary teams. Team members should be "T-Shaped", i.e., both experts and generalists. As experts, they are deeply rooted in their specialized field and bring in the appropriate expertise (vertical line of the T). This can also involve the professional representation of a stakeholder group (e.g., sales, production, IT). Looking at a problem from different perspectives and synthesizing know-how and experience from different domains often helps to develop new approaches to solutions. The quality as generalists is a prerequisite for changing from one's own perspective to that of other participants and for being open to cooperation at the (functional) interfaces (horizontal line of the T, "We" thinking) [8, p. 122]. Lewrick et al. plead for interdisciplinary versus multidisciplinary teams because the former truly and collectively generate ideas and stand behind them, whereas the members of multidisciplinary teams often overestimate their own perspective when finding solutions. The latter rather leads to compromise solutions, which are not fully supported by all [8]. Ideally, an interdisciplinary team is made up heterogeneously of representatives not only from as many areas of expertise as possible, but also from different age groups, nationalities, sexes etc.

The success of a Design Thinking project is largely determined by the individual characteristics of the members of the associated team and the resulting collaborative working culture and way of thinking. The focus here is on showing high esteem and empathy for people as the starting point for all activities, both for colleagues in the team as well as for users or customers. In addition, the team members should have qualities such as the ability to cooperate, curiosity, joy of experimentation, integrative thinking and optimism. A further success factor is the guidance of the team by a

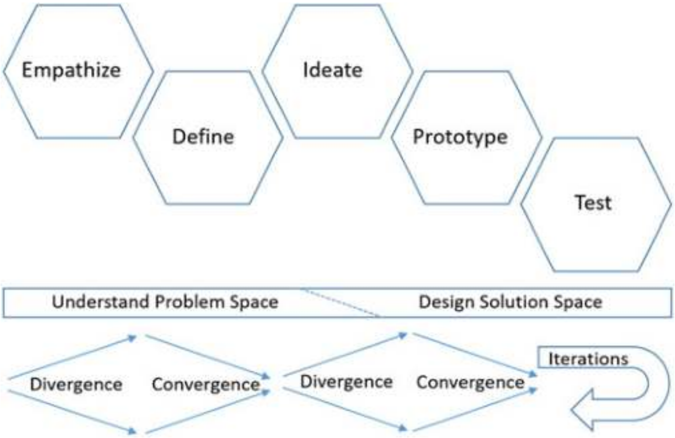


Figure 5.6: Design Thinking process according to Stanford d.school

facilitator who is experienced in the process and the use of the method. This person provides orientation for the respective phase of the process and for which instruments can best be used there without, however, intervening in terms of content.

Process

Design Thinking follows a process model with a series of steps. Although slightly different variants of the so-called microcycle with alternative names of the phases have developed over time, their content only differs marginally. We follow the model of the d.school in Stanford, which is based on five phases of the Design Thinking process, also known as working modes (cf., Fig. 5.6).

All models are characterized by the alternation between divergent and convergent viewing, thinking and acting, both in understanding the problem space and shaping the solution space. The transition between the expansion of the creative space with an ever-increasing amount of information and focusing through containment is called a groan zone, i.e., a "creaking" hinge [8, p. 28f.]. Deliberate iteration is also a fundamental element in the DT process, which is expressed in the motto "fail early, fail often" or "fail forward" in an open culture allowing errors. It describes the idea that ideal solutions can only be found through multiple and early experimentation, testing and consideration of the feedback of the target group, which can lead to more or less extensive new runs of previous modes. The multiple iterations as a so-called macrocycle should lead from the understanding of the problem to the concretization of a vision for a solution and finally to an implementation plan [8, p., 37]. In all process phases the principle "Be visual & show" applies, which means that thoughts, ideas and results are to be visually and vividly documented and presented, for example through post-its with keywords and drawings, mind maps, process maps, tangible prototypes, etc. Because of this principle it is sometimes suggested to speak of Design Doing rather than Design Thinking.

For the successful work of a team along the process, a number of rules and tips (rules of engagement) have proven to be helpful, some of which should also be followed in group work in general. They should be communicated at the beginning of a project and called to mind later by the facilitator if necessary:

- User-oriented way of thinking and working (development of empathy)
- No specification or limitation of ways of thinking, ways of finding solutions or fixed solutions (openness, autonomy, "Go for quantity")
- Concentration on the topic (focus), therefore no distraction by smart phones, computers, smart watches, etc. unless they are to be used deliberately, e.g., for research or prototype creation
- Active participation of each team member (own articulation, disciplined listening ("One conversation at a time") and building on ideas of others
- Encourage the development of different perspectives and wild ideas
- No value judgement during idea generation ("Defer judgement", "No killer phrases")
- Time boxing to avoid falling in love with a particular idea

The following sections briefly describe each phase together with tables including a selection of methods and tools that are used in each phase respectively. More detailed explanations can be found for example in the Bootcamp Bootleg of the d. school [9] and in Lewrick et al. [8, p., 36].

Empathize (Building Empathy)

Empathy is the heart of a human-centered design process. To develop it means to build a deep understanding for the members of the target group with regard to the problem and its context. The aim is to understand why and how people do things, how they think about the world and what is important and useful to them, as well as to learn about their physical and emotional needs. This means observing users, listening to them, interacting with them, imagining and empathizing with their situation and thus immersing into their conscious and unconscious world of feelings, values and needs (engage, observe, immerse). This is the basic prerequisite to steer innovations in the right direction.

So-called personas are an essential instrument for documenting the learnings about the target group. Personas represent fictional customers or users together with their objectives, behavior, needs and attributes relevant for the solution to be developed.

In the model of the Hasso Plattner Institute, the "Empathize" phase encompasses the stages "Understanding" and "Observing". Table 5.4 shows common instruments for the activities included in this phase.

Define (Defining Problem)

This mode is about building on the findings from the "Empathize" mode, sharing and bringing them together, structuring, weighing and interpreting them. This synthesis serves to test and further develop the personas for ideal-type users and, if necessary, to adopt the perspectives of various stakeholders. The results are a deeper understanding of the users and the problem space as well as a more concrete, meaningful

Table 5.4: Methods and tools for the "Empathize" phase

Activity/Theme	Selected methods and tools
Exploit the context of the problem and develop a common understanding of it	• Brain dump • Business Process and Value Maps, Concept Maps, Business Model Canvas
Structure the problem context	• Structuring/Clustering frameworks
Identify and understand target group (user, customer)	User Profile Canvas with • Persona • Jobs-to-be-done • Gains & Pains • Use cases
Understand the target group (user, customer)	• Need-finding discussion, interview for empathy (incl. preparation), e.g., with W questions (What? How? Why? Who?) according to the AEIOU method (Activities, Environment, Interaction, Objects, User) • Empathy map • Future user

problem definition (design challenge). The latter is reflected in a single sentence which, as a so-called Point of View (POV), forms the question for the subsequent phase of idea generation [8, p. 73]. In practice, different POV questions are used. A typical formulation is the "How might we?"-question, for example, "How might we help [user, customer] to reach [a certain goal]?" [8, p. 74].

In the Hasso Plattner Institute model, "Define" corresponds to the "Define point of view" phase. Table 5.5 lists common tools for the activities included in this phase.

Ideate (Finding Ideas)

The aim of idea generation is to develop a wide range of solutions, i.e., to develop and visualize as many ideas, and as many different ideas, as possible. The starting point is the point-of-view question, however all the insights gained so far are incorporated into this phase, including user profile canvas, empathy map and customer/user experience journey. The basic instrument is brainstorming, which can be further and repeatedly stimulated by creativity techniques and specific tasks (e.g., generating ideas for certain functions). The design and testing of initial "low-fidelity" prototypes can also provide further food for thought for solutions and trigger iterations. The use of methods in this mode should enable going beyond obvious solutions, and thus increase the innovation potential by using the collective perspectives and strengths of the team. Unexpected solution directions should be able to emerge and contribute to the quantity and diversity of ideas. This results in a multitude of ideas, which are sorted, condensed and evaluated. The entire process should be strictly separated between the generation and evaluation of ideas, so as not to restrict the creative flow at an early phase.

In the Hasso Plattner Institute model, "Ideate" corresponds to the "Finding Ideas" phase. Common instruments for the activities contained in this mode are shown in Table 5.6.

Table 5.5: Methods and tools for the "Define" phase

Activity/Theme	Selected methods and tools
Share insights	• Story share and capture (Storytelling)
Interpret insights, draw conclusions	• Saturate and group • Empathy map • Customer/User Experience Journey (with actions, mindset, touch points, pain points, moments of truth)
Understand the target group (user, customer) even better	• Persona • Composite Character Profiles • Power of Ten • 2x2 Matrix • Why-How-Laddering • Point of view . . . • 360-degree view • A day in the life of . . .

Table 5.6: Methods and tools for the "Ideate" phase

Activity/Theme	Selected methods and tools
Generate ideas (iteratively)	• General brainstorming based on POV (e.g., How might we?) with stimulation through creativity techniques • Targeted brainstorming (critical functionalities, benchmark, dark horse, funky prototype) • Power of Ten, Bodystorming • Quick&Dirty Prototyping
Sort and condense ideas	• Swap Sort, 2x2 Matrix • Concept/Systems/Mind Maps, idea profiles
Evaluate and prioritize ideas	• Four-category method; Post-it voting, Spend your budget

Prototype (Creating prototypes)

Prototyping picks up the most highly rated ideas from idea development and continues to develop them further. In doing so, the principle of Design Thinking is implemented: to visualize issues, products and results as early as possible and to test, discuss and further develop them with potential users, incorporating their feedback into tangible models. Prototypes are thus created in order to learn to clarify open questions and discrepancies, to start a conversation or a discourse and to recognize dead ends quickly and at an early stage, which in turn saves costs. In addition to the change requests, the feedback can also lead to complete rejection and thus to a fundamental iteration over more distant previous phases. This procedure is also described by the slogan "Love it, change it or leave it".

Prototyping transfers ideas from the mind into the physical world. A prototype can therefore be anything that takes on a physical form and follows the maxim "don't tell me, show me!": a wall with post-it notes, a role play, a room, an object, a storyboard or any combination of different means of expression.

The granularity of the prototype should correspond to the progress of the project. In the early stages of a project, prototypes should be created that can be made quickly

Table 5.7: Methods and tools for the “Prototype” phase

Activity/ Theme	Selected methods and tools
Create prototypes	• Low-fidelity prototypes, e.g., from handcrafting material (Lego, modeling clay, etc.) • Role plays, Storytelling, Storyboards • Wireframes, Screen-design tools • Shooting and editing video

and cost-effectively (low fidelity, quick & dirty), but already generate useful feedback from users and colleagues. In later stages, the prototypes should be refined and allow careful investigating of specific issues. They serve to deepen empathy, to test and to gain further ideas and inspiration.

In the Hasso Plattner Institute model, "Prototype" corresponds to the "Develop prototype" phase. Table 5.7 shows common tools for the activities included in this phase.

Test (Testing prototypes)

As discussed in the previous section, testing is closely linked to prototyping. The recommendation "Prototype as if you know you're right, but test as if you know you're wrong" describes the way of thinking that illustrates this relation. Testing offers the opportunity to receive qualitative feedback on the prototypical solutions, to make them better, to learn more about the users, and thus to deepen the empathy for them. The test mode is an iterative learning mode in which the prototypes are placed in the context of the potential user, then used and evaluated by him. Important principles are: “Don’t talk, show!”, create experiences, and enable the user to make comparisons. The feedback during the tests can lead not only to changes, but also to complete rejection and thus again to a fundamental iteration over more distant previous phases. This procedure is also described by [8, p. 34] the slogan "Love it, change it or leave it". In principle, each iteration loop, regardless of its scope, must reflect which previous results (e.g., personas, user/customer experience journey) have to be adapted as a result of the feedback.

In the Hasso Plattner Institute model, "Test" corresponds to the "testing" phase. Table 5.8 contains instruments for the activities carried out during the phase.

Place

For the work of the interdisciplinary teams in the described modes, it is necessary to create a creativity-fostering environment, so-called make or creative spaces. This applies in particular to the availability, size and furnishing of premises as well as to visualization and prototype design tools and materials. The main aim is to provide the teams with freely and permanently available work, interaction, relaxation and storage areas. Flexible furniture with castors, describable and erasable surfaces (walls, tables, boards), as well as good and fast access to information (Internet, libraries, etc.), tools, working materials and catering add to a suitable environment

Table 5.8: Methods and tools for the “Test” phase

Activity/Theme	Selected methods and tools
Gather feedback	• Feedback grid • "I like, I wish, What if?" • A/B testing with digital tools

[7, p., 216]. Lighting, ventilation and air conditioning are also important factors to be considered.

In practice, teams are sometimes given the opportunity to design the environment themselves (e.g., build their own furniture), especially in the case of long-term projects. Doorley and Witthoft have published instructions and experiences, among other things, in the design of creative environments for the d.school [10].

5.4 Connecting the Concepts

5.4.1 Overview

As shown, Design Thinking aims at the innovation of products and services, business models and business processes. The focus is on user centricity, creativity and agility in an experimental, iterative process that interdisciplinary teams traverse.

Process management pursues a comparable objective with agile and creative process design of new, or redesign of existing, processes under consideration of customer needs.

In the following, we put the concepts into relation to one another and discuss the promising use of Design Thinking elements for process management.

We pay particular attention to the digitalization of processes, i.e., the reasonable use of information and communication technology for process improvement and innovation. Prototypes and final solutions are therefore always workflow applications with different degrees of automation.

5.4.2 User Centricity

During process analysis traditional BPM approaches usually involve those participating with interviews and workshops. However, the 'hard' facts of the work in the process with the characteristics listed in Chapter 5.2.2 are in the foreground of the activity-related and process-related interview questions, card techniques or observations. Aspects such as understanding users' motivation, ways of thinking, and values, which are expressly emphasized for the development of empathy in Design Thinking, are largely ignored here. Newer concepts such as Social BPM have changed little in this regard.

The Subject-oriented Business Process Modeling (S-BPM) approach, which was already used in the case study at the beginning of this chapter, can build an interesting bridge. It focuses on the subjects as actors in the process. With the associated methodology and language (see Section 3.6) as well as suitable tool environments, representatives of subjects involved in the process can participate in iterative solution development not only as respondents or observers, but also as active designers. They not only explicitly specify the behavior of the subject they represent and its interactions with other participants, but can also immediately test and change the result of their design by executing the resulting model. In doing so, they can implicitly bring in the 'soft' factors mentioned above.

5.4.3 Agile Process with Iterations

In practice, more extensive process management projects are often still carried out using traditional project management methods in clearly defined phases with milestones, comparable to the waterfall model in software development.

This means that the path from analysis of the design of the business model and its organizational and IT implementation to an executable workflow application takes an extensive amount of time. It also increases the likelihood that the resulting IT solution will deviate from the evolving needs and desires of users.

For process digitalization in particular, it is therefore advantageous to adapt the agile, iterative process of Design Thinking. This opens up the possibility of meeting the increasing dynamics with regard to the emergence of new processes and changes to existing processes, for example due to new or changed business models such as servitization. For further considerations, we compare the modes of Design Thinking with the activity bundles in process management (cf., Fig. 5.7).

Together with the explanations in sections 5.2.2. and 5.3.2 the illustration shows that DT makes a stronger distinction between problem understanding and solution design. The latter only begins with "Ideate". Before this, the actual situation is extensively illuminated and, for example, documented and visualized along the way via personas in the customer/user experience journey, before the point-of-view question is formulated as the starting point for generating ideas.

In process management, on the other hand, the problem is usually clearly formulated at the outset. When renewing existing processes, it is usually derived from the desire for improved process performance (e.g., shorter lead times). In practice, therefore, only weak points in the current state are documented and analysis information is used to develop and visualize a new target model, just like in a new process. The creative, design-related part begins earlier than with Design Thinking and tends to be underpinned by less information when being started. It is driven more analytically (e.g., by performance indicators) than by the "soft" factors identified in the course of empathy development in Design Thinking. Regardless of the somewhat different concrete design, the activity bundle **Analysis & Modeling** can be assigned to the DT modes **Empathize**, **Define** and **Ideate**.

The use of proven DT instruments is ideal for a more comprehensive capturing of the problem context and the resulting expansion of the spectrum of solutions for process innovation. Especially when developing a new process, the team members

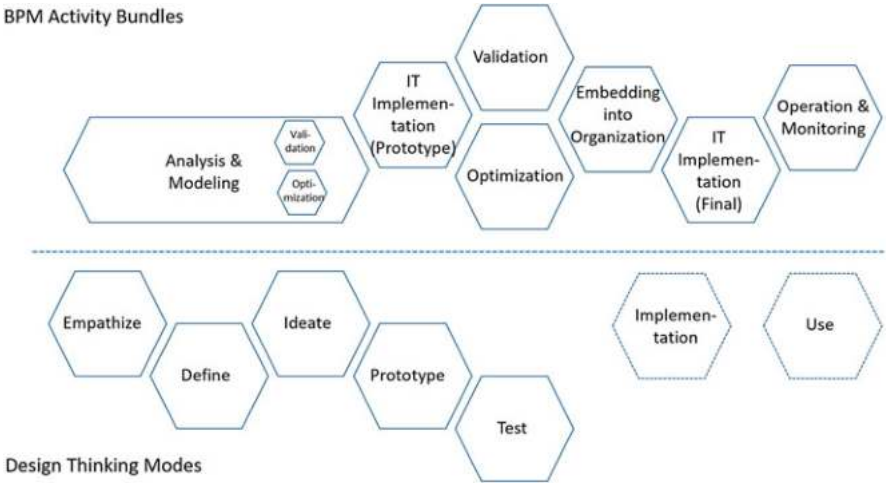


Figure 5.7: Assignment of Design Thinking modes and BPM activity bundles

can broaden their horizon and develop a common understanding with a brain dump concerning the problem environment and the discussion of the results.

With the help of personas for the process participants in their respective roles as well as interviews and observations, customer/user experience journeys can be described.

If participants in the process are team members themselves, they can also visualize their own experiences as journeys. This extends the information base beyond the classic, objective process characteristics to include the user's perspective. This broader foundation for the development of solution ideas should justify the higher effort.

In section 5.2.2 we had explained that effectiveness and efficiency, at least of model excerpts, are already taken into account during the analysis and modeling of processes. This is especially true if the future users do this themselves, as in subject orientation. Therefore, the activity bundles **Validation** and **Optimization** are assigned to the DT-mode **Test**, but also cover **Empathize**, Define and Ideate.

With the focus on process digitalization, the DT **Prototype** mode corresponds to the activity bundle **IT implementation** in process management. Prototyping in Design Thinking makes the claim to producing a prototype quickly and with simple means, i.e., cost-effectively, in order to quickly obtain feedback from the user (Test mode) and to utilize it. By applying this "fail early, fail often" principle to process management, the team must be able to create an executable model with minimal effort. The focus is therefore on creating a functional prototype in the form of software that allows users to experience what their work with the IT solution would look like. However, assigning the prototype to the IT implementation should not mean that programming is necessary. Rather, it must be possible in the interests of rapid iterations to generate a prototype automatically from the model and have it tested by the users in the activity bundle validation. In the same way, subsequent model changes based on feedback again lead to a new prototype, until a version is found that satisfies the users. Such low-cost and early prototyping possibly prevents more complex reworking during the later realization of the real runtime environment. Using a comparable approach, the user interface can be designed according to the principles of user experience design.

Since the respective process model is not only the basis for prototypes, but in its ultimately adopted version also for the workflow application strived for, the activity bundle **IT implementation** includes also their realization in the way described in section 5.2.6. If software which goes beyond the model-based workflow control has to be developed for this purpose, SCRUM as a user-centered, agile software development method serves as a good choice. In the context of IT implementation, it is also important to decide to what extent the strategy of minimum viable products, often used by software start-ups, should be pursued. This would mean making software with minimal functionality available to customers or users not only prototypically, but also productively in order to obtain their feedback for further development. This could be risky with IT solutions for business-critical processes; on the other hand, it could possibly give an edge over competitors by familiarizing customers with features at an early stage.

As explained in sections 5.2.5 and 5.2.7, the model must also be embedded into the organization (**Organizational Embedding**) before the process can go live (**Operation & Monitoring**).

In Design Thinking, comparable steps follow for the implementation, for example, of a product on the basis of an accepted prototype as well as for its use. However, these steps no longer belong to the modes in the narrower sense (dotted forms in Fig. 5.7).

Conclusion

In order to meet the requirements of digitalizing processes, a process management approach should combine Design Thinking and process management concepts.

It must be suitable for quickly mapping processes and their changes both (business) domain-related and in IT, while at the same time adequately involving the users in short iteration cycles in order to approximate the resulting solution to their ideas.

In addition to the instruments that can be used for the Design Thinking modes Empathize, Define and Ideate, easy-to-handle methods and tools are particularly necessary for this purpose, with which the team and/or the process participants themselves are able to:

1) articulate their individually different mental models of work
2) harmonize these different mental models
3) develop ideas for solutions and concrete proposals for solutions in the form of models
4) automatically convert these models into executable prototypes and test them
5) transfer released models to live workflow applications with limited effort

An example of a concept to support 1) and 2) is Compare/WP (see chapter 6.1.2). Requirements 3), 4) and 5) are, for example, covered by the S-BPM approach and BPM tools based on it [11, 12].

5.4.4 Interdisciplinary Team

The importance of the interdisciplinarity of the facilitator-led team in Design Thinking was explained in section 5.3.2.

BPM projects are also usually carried out by teams. We distinguish thereby between four roles:

- **Governors** set the determining factors for the project. These essentially comprise the scope, i.e., the delimitation of the process system worked on in the project, as well as the methodology and tools, specifically related to analysis and modeling.
- **Actors** are the present or future actors who carry out the actions in the runtime instances of the process to be changed or developed. They are therefore the carriers of concrete execution-related process knowledge for their part in the creation of the process result, i.e., they know which sub-steps they carry out in which order, which information and tools they need for this and with whom they interact.
- **Experts** support the other roles with methodical and domain-related knowledge. They are, for example, domain experts (specialists) who have expertise in the relevant field that goes beyond that of the actors, and which they can contribute as such. Method experts help the participants, especially the actors, to articulate and harmonize their mental models and to implement them with one of the modeling languages presented in Chapter 3. Finally, IT experts are called in for the technical implementation of the business process models. If required, experts in the various fields from outside the company's own organization are also involved as external consultants.
- **Facilitators** moderate and coordinate the approach and the cooperation of the participants. For example, they ensure that actors coordinate the interfaces between their work steps and, if necessary, identify and involve suitable experts for particular problems. In the course of all this, facilitators motivate involved people to act compliantly in accordance with the determining factors set by governors and monitor their behavior respectively.

With the traditional phase-oriented approach in BPM, usually a project leader as a facilitator coordinates the collaboration of domain and method specialists as actors and experts during analysis, modeling, validation and optimization.

The implementation of the approved business process models is then carried out by IT experts. In section 5.4.3., we have already identified long duration and the resulting deviation from the stakeholders' needs as probable disadvantages of this approach.

For some time now, the **BizDevOps approach,** which is intended to take into account the increasing agility requirements in the course of digitalization, has been becoming more widespread. It strives for a comparatively closer integration of the business departments (business, biz), IT development units (development) and IT operation units (operations)[1]. Right from the start, the agile team includes representatives from all areas in the roles described, who jointly design the process solution. The concept can thus both improve business and IT alignment and also foster enabling through IT. The former means that the degree of coverage of the business departments' needs increases through appropriate IT services. In enabling, IT gives impetus to the use of information and communication technology for business model and business process innovations.

Like Design Thinking, the BizDevOps approach therefore involves an interdisciplinary team. The challenge in such teams is to establish the "We" thinking among "T-shaped" individuals. This is due to the fact that line units from which the participants originate (various business departments involved in the process, IT development, IT operations) pursue different goals and often tend to give them a higher priority than a goal to be achieved jointly (innovative or improved process solution). In addition, creativity may be hampered by the involvement of domain experts. On the one hand, process participants are often aware of weak points in existing processes and ways for improvement. On the other hand, they may be 'operationally blind' and too restricted in their consideration of the problem space and, in particular, the solution space. Recruitment should therefore not only take into account diversity aspects such as gender, age and cultural background. Rather, a meaningful balance needs to be found between domain experts and team members who have a background in other fields and have no strong self-interest in the appearance of a solution. The shift in emphasis can be made dependent on whether the goal is more a process improvement in which the experiences and inputs of those familiar with the existing process can be helpful. If, on the other hand, the focus is on a more radical process innovation, this could possibly have a limiting effect. Of course, even with the original objective of improvement, ideas for a fundamental innovation of the process under consideration should not be ignored.

References

1. Allweyer, T. (2009). Business process management: Strategie, Entwurf, Implementierung, Controlling, W3L, Herdecke
2. Schmelzer, H., & Sesselmann, W. (2013). Geschäftsprozessmanagement in der Praxis – Kunden zufriedenstellen, Produktivität steigern, Wert erhöhen, 8. Auflage, Munich, Carl Hanser.
3. Schallmo D. R. (2017). Design thinking erfolgreich anwenden. Springer Gabler.

[1]When looking beyond company boundaries, one could add partners in the value creation network such as suppliers, customers or logistics service providers (Network Partners) and speak of NetBizDevOps.

4. Brenner, W., et al. (2016). Design thinking as mindset, process and toolbox. In W. Brenner & F. Uebernickel (Eds.), *Design thinking for innovation* (pp. 3–21). Cham: Springer.
5. Schmiedgen, J., Rhinow, H., Köppen, E., & Meinel, C. (2015). Parts without a whole? The current state of design thinking practice in organizations. Technische Berichte des Hasso-Plattner-Instituts für Softwaresystemtechnik an der Universität Potsdam Nr. 97, Univ.-Verl., Potsdam.
6. Leifer, L. (2012). Über design thinking, bad guys, experimente, Jagd und organisationalen Wandel. In: Organisations Entwicklung, Nr. 2, 2012, pp. 8–13.
7. Uebernickel, F., Brenner, W., Pukall, B., Naef, T., & Schindlholzer, B. (2015). *Design thinking – Das handbuch*. Frankfurt am Main: Frankfurter Allgemeine Buch.
8. Lewrick, M., Link, P., & Leifer, L. (Eds.). (2017). *Das design thinking playbook – Das handbuch*. München: Vahlen.
9. Last access on May 1, 2018, from https://dschool.stanford.edu/resources/design-thinking-bootleg
10. Doorley, S., & Witthoft, S. (2012). *Make space: How to set the stage for creative collaboration*. Hoboken, NJ: Wiley.
11. Fleischmann, A., Schmidt, W., & Stary, C. (2013). Subject-oriented BPM = Socially executable BPM. In *Proceedings of the 15th IEEE Conference on Business Informatics (CBI 2013)* (pp. 399–406). Vienna: IEEE Computer Society.
12. Fleischmann, A., Borgert, S., Elstermann, M., Krenn, F., & Singer, R. (2017). An overview of S-BPM-oriented tool suites. Proceedings of the 9th S-BPM ONE, Darmstadt.
13. Tornbohm, C., & Dunie, R. (2017). Market guide for robotic process automation software. Gartner Report G00319864, Stamford.
14. Geyer-Klingeberg, J., Nakladal, N., Baldauf, F., & Veit, F. (2018). Process mining and robotic process automation: A perfect match. In F. Casati et al. (Eds.), *Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018.* Last access on May 2, 2019, from http://ceur-ws.org/Vol-2196/BPM_2018_paper_28.pdf
15. van der Aalst, W., Bichler, M., & Heinzl, A. (2018). Robotic process automation. *Business & Information Systems Engineering, 60*, 269–272. https://doi.org/10.1007/s12599-018-0542-4.

Preparation of Process Implementation 6

The aim of the process preparation in the sense of a subsequent implementation is a precise description of the process with a description of the process strategy and process logic. The preparation includes the activity bundles on the left side of the open cycle, i.e., analysis, modeling, validation and optimization (see figure below). The result of these activity bundles is a process description that is sufficiently precise for implementation. The preparation is split into the activities analysis combined with modeling, validation, and optimization. These activities are not carried out in a strict order, but rather the respective priorities can change frequently between activities. Figure 6.1 shows these various activities and their relationships.

The following sections present selected methods for these activity bundles.

6.1 Analysis and Modeling

Analysis and modeling cannot be sharply separated. The analysis focuses on the strategic aspects of processes, while modeling focuses on the process logic. In the analysis the starting point with its associated input, the end state with its generated output, and the therewith satisfied customer needs are clarified. In the analysis, the framework and the essential aspects of the process logic are also defined.

In practice, however, the process logic of the actual state is hardly explicitly described when revising processes in this phase. The analysis of the current process logic is accomplished within the framework of the definition of the desired target process. An exclusive reference to the current situation usually makes little sense and is also as a rule unpleasant for all participants to document - What has been "done wrong" lately?

As long as one is using the tool 'natural language', the focus is more on the activity of analysis than on that of modeling. The transition from natural language to a more formal process modeling language corresponds to the transition to modeling activities. The modeling can be preceded by a more or less intense analysis method. In extreme cases, a process model is immediately created without prior natural

A. Fleischmann et al., *Contextual Process Digitalization*,
https://doi.org/10.1007/978-3-030-38300-8_6

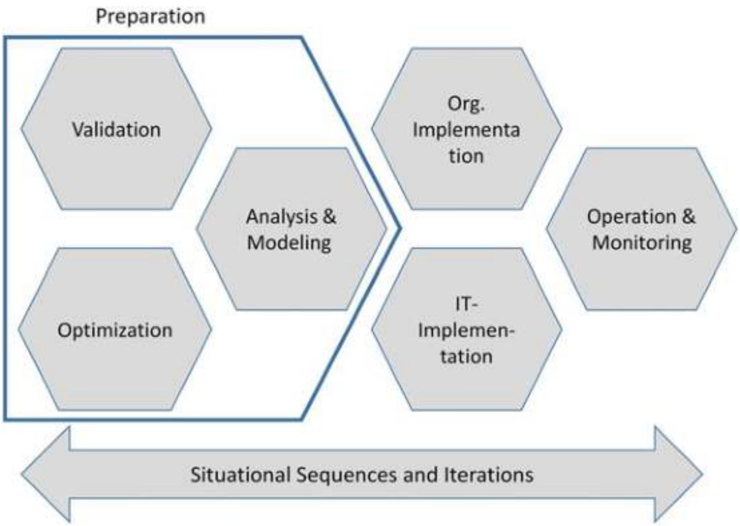


Fig. 6.1 Integration of the preparation in the process management model

linguistic analysis. However, it is recommended that at least the strategic aspects of the process under consideration are known and defined.

In the following, guidelines for the articulation and coordination of process-relevant knowledge, which can be supported methodically and tool-wise, are presented. An essential element is the understanding of roles which the participants consider relevant for the handling of processes. In addition, it is advisable to consider the exchange relationships between actors, to evaluate their quality for the further design of processes and, if necessary, to derive potential for change from this information.

6.1.1 General Information on Articulation and Coordination

In most cases, knowledge about workflows and organizational processes rests in the minds of the actors. A context-sensitive, structured survey and analysis is therefore of crucial importance. The survey serves to articulate experiential knowledge and in most cases is carried out within the framework of modeling. However, if it is already handled in advance, the variety of approaches to solving tasks or problems in BPM projects can be dealt with in a more structured way. Nevertheless, in the context, the coordination and alignment of different approaches plays an essential role. Supporting these contributes significantly to the development of solutions that are capable of integration, despite a high degree of diversity and individual approaches to the fulfillment of tasks. This section therefore deals with the survey and negotiation methods and instruments that enable individuals to articulate within the framework of collective reflection and negotiation processes.

How people carry out their work, how they react to perceived specifications or deviations and how they cooperate with others is essentially determined by their perception of organizational reality. The interpretation of the perceived determining factors as well as the derivation of the reaction considered adequate of the acting workers can be explained by the cognitive theory of mental models. This theory can also be used as a basis for explaining learning and change processes in organizations that are initiated by operatively active persons. In this section, it therefore forms the basis for the derivation of measures which should enable workers to become aware of their work processes and the organizational interrelationships and determining factors characterizing them.

The concept of "mental models" is used to explain how people understand the world - more precisely: how they use their knowledge to make certain phenomena of the world subjectively plausible [1]. Mental models are explanatory models of the world that are formed by people on the basis of everyday experience, previous knowledge, and conclusions based on these. A mental model is used by each individual as a basis to understand the world and, if necessary, to make predictions about its behavior [1].

The knowledge that shapes mental models can be based on everyday experience or can be founded on conveyance or instruction. Seel [1] describes the modification and expansion of one's own knowledge bases and the (further) development of the cognitive abilities necessary for drawing conclusions as "learning". Learning is linked to the processing of individual experiences with, and information about, the world, its structure and evidence, and can be understood as a process of permanent conceptual change [1]. Learning thus presupposes the ability and willingness to understand and accept conveyed world views and then to base one's own mental constructions on them [1].

There are two basic difficulties in changing mental models about work processes. In the case of mental models that have already been recognized as inadequate, there is a fundamental willingness to change (in the sense of adapting the mental model to the environmental conditions perceived as changed), but the challenge is to obtain the necessary information and have it adequately presented. A further difficulty arises in situations in which not all individuals involved perceive the situation as 'problematic' and therefore show no fundamental willingness to change their underlying assumptions with regard to their way of working (i.e., their mental models). This occurs especially in situations where collaborative reflection is not carried out from a generally perceived problem situation, but is either initiated with a purely planning character, or in situations which are perceived as 'problematic' only by certain individuals involved.

These problems can be countered with explicit support for the reflection process. Such support must ensure that artifacts are created to represent the individual mental models, which can then serve as the basis for mutual understanding of the respective views on the work process. Such artifacts can serve to coordinate aspects of a work process and to ensure that the ostensive view of a work process encoded in artifacts can be implemented in work practice through performative subjective action know-how based on it. From a methodological point of view, it must be ensured that all persons involved in the real work process are organizationally and methodologically capable of participating in the collaborative learning process. This requires above all that they can understand and actively use the forms of expression utilized. This, in turn, is a learning challenge that must be explicitly addressed.

A widely accepted option for externalizing and harmonizing mental models in the educational sciences is the formation of conceptual models. At the same time, such models can form the basis for the specification of work processes and the configuration of work support systems, as long as they make use of a formally specified semantics (such as BPMN or S-BPM). In accordance with the objective of this section, conceptual models thus represent a means of enabling workers to reflect on their work, to coordinate it, to make the results of these coordination processes accessible to third parties, and to make them usable within the framework of existing system boundaries to support their own work processes.

Models are representations of reality that are provided for a particular purpose. Models never represent the real phenomenon as a whole, but contain only those aspects of reality that the modeler considers relevant for the achievement of the respective goal. For modeling, this raises the question of the defining power of these models and the social reality they represent. If a model does not only fulfill an objective of the modeling individual, but is used by other persons, the model influences the mental models of these persons, and thus also their behavior.

The active involvement of operative workers in the specification of work processes is therefore an opportunity for their self-empowered development of organizing their work. To this end, however, it is necessary to enable workers to understand such models, to design them by themselves, and to assess their impact on their work processes. Current approaches, on the other hand, continue to assume the need for a process analyst who translates the views of workers into a process model. This can lead to deviations between the real work process and its model representation. In addition, this approach deprives the operatively active persons of the opportunity to sharpen their mental models in the sense of model-based learning and to coordinate them with those of the other participants.

In order to enable workers to understand such models, the learning of basic approaches to the creation and interpretation of conceptual models must be the subject of education or training. Workers must be able to identify the models underlying the systems in which they are embedded. In addition, they should be able to assess the implications of external or self-implemented changes to these models and to plan interventions accordingly.

For this purpose, the following points need to be methodically supported:

- to enable the individual articulation of one's own mental models with respect to work in order to enable individual reflection and thus to make gaps and inconsistencies individually perceptible, as well as to prevent that perspectives of individual persons are not taken into account and that these cannot subsequently establish a reference to their working reality
- to support the agreement on a common vocabulary in order to identify different understandings of terms and subsequently to be able to communicate clearly

about the work in question and to prevent the same real phenomenon from being described by different terms - or vice versa the same term from being used for different real phenomena
– to support the development of a common understanding of collaborative work to provide a basis for reflection of individual mental models
– in the context of the above, to enable the identification and resolution of conflicting points of view, in order to make differences in those mental models that directly affect collaboration between workers visible, and to facilitate their coordination.

The following sections show some of the methods used to implement these requirements.

6.1.2 CoMPArE/WP

The requirements described above are implemented exemplarily in the "CoMPArE/WP" method. CoMPArE/WP stands for "Collaborative Multi-perspective Articulation and Elicitation of Work Processes". In the application of CoMPArE/WP, the reflection on a real collaborative work process creates awareness of the cooperation in a concrete individual case. Due to its anchoring in concrete work processes, the method is also suitable as a means of organizational development. The form of cooperation of the method is basically determined by its implementation with a card laying technique. The participating operative workers are the essential actors and carry out the components autonomously, whereby articulation and inquiry roles can change. The concrete form of the cooperation differs in the components and is therefore described in the procedure mentioned there. A facilitator is available to support the implementation, but he does not intervene in regard to content.

The method should support the articulation and coordination of mental models with respect to work, and at the same time impart basic skills for their expression in conceptual models. The combination of these two sub-objectives has an impact on the framework of the method. From the point of view of teaching modeling competence, it makes sense to introduce the necessary skills step by step with increasing complexity. From the point of view of articulation support, an approach can be contended in three components. Figure 6.2 gives an overview of these three components.

Component 1 is used to find a common understanding of where and how the work process to be coordinated begins and ends and to find a common vocabulary. Component 2 is used for articulation and reflection of the respective individual work contribution. Each participant creates here a structured model of their point of view on their respective work contribution, individually and without interaction with others. Due to the uniformly structured presentation of the individual contributions, a collaborative alignment of these is possible in component 3. This alignment is intended to uncover conflicting points of view and to create a common view of the overall work process.

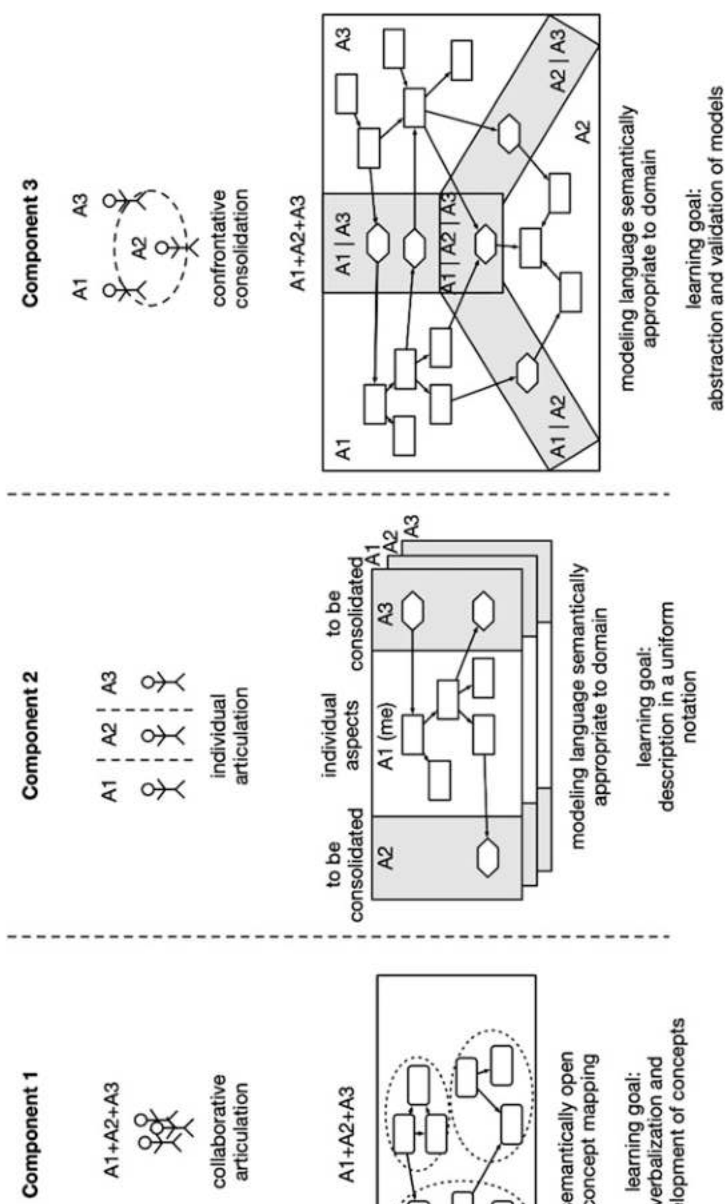


Figure 6.2: Three components of CoMPArE/WP

The objectives of skill development in modeling are anchored in these components. Component 1 aims to convey the verbalization of mental models and the concept formation based on them. In component 2, the description of the verbalized contents must be represented by means of a predefined category scheme and a notation. In doing so, it needs to be determined which elements of the category scheme remain in the responsibility of the articulating individual and which need to be validated and possibly abstracted or become the subject of negotiation during alignment in component 3.

Concrete Implementation of Component 1. It cannot be assumed that all participants have a common understanding of the concepts they use when describing their work. Collaborative concept mapping can be used to align the existing mental models to such an extent that a common vocabulary enables collaboration. In addition, it cannot be assumed that there is a common understanding concerning the borders of the work process to be coordinated. Concept mapping can also help to clarify this issue. In addition to the content dimension, concept maps provide a low-threshold entry into the world of conceptual modeling, since they do not predetermine the meaning of model elements, but rather allow the persons involved to define them during modeling. This facilitates the mapping of the individual mental models into the explicit representation and avoids the necessity of having to carry out a translation to a model with formally defined semantics in addition to the coordination with the other persons involved.

Within the scope of this component, the participants are asked to describe all relevant aspects of the environment in which the work process to be reflected is embedded. This is done by individually writing each aspect on a separate card. When the collected individual aspects are brought together, the cards are arranged in turn on a common work surface. The aspects can be put into relation to each other. Cards with different terms for the same aspect are arranged overlapping. Hierarchical or causal relationships between aspects can be represented by drawing explicit connections, but also by the spatial arrangement of the cards. The example in Figure 6.3, which is used throughout this section to explain the method, shows a concept map with relevant aspects for applying for a vacation in a company. The aspects were related to each other by spatial arrangement. The overlapping elements show aspects that are mentioned by several participants and are described using different terms.

Concrete Implementation of Components 2 and 3. Components 2 and 3 focus on the articulation and alignment of the perceived course of a work process and the associated interaction within this process. The modeling in component 2 is carried out individually by all persons involved, without interacting with others. This avoids overlapping effects and explicitly reveals different perspectives for the next component. The persons involved describe which of their work steps they see as contributing to the achievement of the work goal, with whom they interact and in what form this interaction takes place. The negotiation of a common point of view in component 3 and the associated creation of a common model is again carried out by

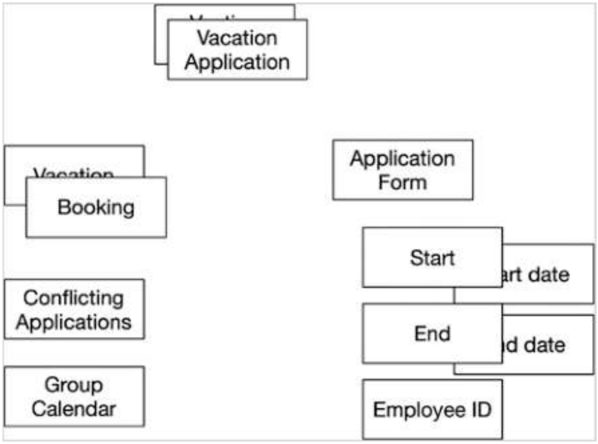


Figure 6.3: Concept map on applying for a vacation

means of a structured procedure, which is to introduce more complex modeling tasks and guarantees a uniformly prepared model representation. In doing so, the previously created models are used further. The structuring scheme separates those model aspects that remain in the responsibility of the individual worker from those that are the subject of negotiation.

In this step, a structured form of representation with pre-specified semantics is used to represent the work processes. It is based on the current category schemes for the description of collaborative work processes. The categories WHO, WHAT and EXCHANGE (M3) are used. WHO (blue in Figure 6.4) refers to the actors in the work process. WHAT (red in Figure 6.4) is used to describe active contributions in the scope of the work process. EXCHANGE (yellow in Figure 6.4) is used in the context of collaborative work processes to characterize the sharing or exchange of information or material between actors in the context of their own activities. For the sake of usability, these categories are not specified exactly and deliberately leave room for interpretation in concrete use, for example, a WHO element can represent a concrete person, a role, a department or an entire organization. WHAT elements remain the responsibility of the individual participants. WHO and EXCHANGE elements are the subject of coordination in component 3 and must be developed toward a common understanding.

Individual Articulation. In component 2, the persons involved individually describe with the help of the elements, what they contribute in the work process, who interacts with them, and in what form this exchange takes place. In order to support the articulation process, a structuring scheme was developed that prepares the models in a coherent form and allows them to be combined in the next step. As shown in Figure 6.4, the structuring scheme defines the spatial arrangement of the model elements. Operative workers represent themselves through a WHO element, under which the perceived contributions to the work process are placed as sequentially arranged WHAT elements. For all other workers with whom an interaction is

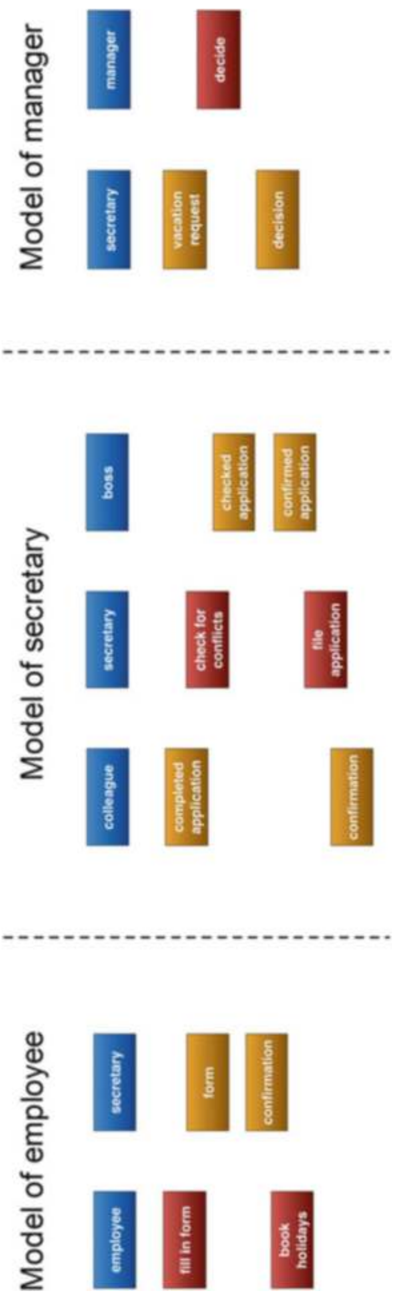


Figure 6.4: Individual models in the presented notation

perceived, another WHO element is placed, under which the interaction is specified in more detail by EXCHANGE elements. Their vertical positioning determines whether an incoming resource is expected (placement above the dependent WHAT element) or provided (placement below the generating WHAT element).

Figure 6.4 shows three individual models for the example process described above, which were created according to this structuring scheme. In the example, it can be seen that at this point there may be divergent representations with regard to content, especially in the area of exchange elements (cf., "Application" vs. "Completed application" in the figure above). These divergences become explicitly visible in component 3 and are then subject of the negotiation of a common perspective.

Collaborative Alignment. Collaborative alignment is based on the individual conceptional models created in component 2. Figure 6.5 shows an exemplary alignment process for two of the actors represented in the example. The common modeling again takes place on a common work surface (see Figure 6.5 in the middle). The participant, who triggers the real work process, begins by describing his own contributions to the process and adding the corresponding model elements to the surface (steps 1-2 in the following figure).

The other participants intervene here only inquiringly to avoid misunderstandings or to disclose ambiguities. An active participation of the others takes place as soon as the first EXCHANGE element is used (steps 3-4). If a fundamentally common view of the work process exists, one of the participants should be able to introduce a correspondingly assigned EXCHANGE element at this point (steps 5-7).

If this is the case, the description process is continued by this person (from step 8). In the case of a basic fit, which differs, however, in the designation of the element, e.g., by different abstraction levels, this conflicting designation must be resolved, or the semantic equivalence of the two elements must be represented by overlapping arrangement (e.g., step 7). If there is no element to be assigned, fundamental differences in the representation become visible. This may be due to a lack of awareness of the relevance of an exchange, which means that the addressed participant was aware of the interaction, but did not consider it relevant in the context of the work process. However, if a participant's perceived need for interaction is not reciprocated, this must lead to more in-depth alignment processes.

The initial alignment process ends as soon as all of the participants have explained their individual models and added them to the common model. This externalization phase is followed by a collaborative reflection phase, during which the work process is examined on the basis of the common model and discussed with regard to its adaptation to the individual views of the participants. Any necessary modifications are carried out at this point, after consensus has been reached among those affected.

The ***result of*** the application of the method now represents a consensual representation of the collaborative work process. Due to the limited expressiveness of the modeling language used, it is not possible at this point to map work variants or decisions that are otherwise common in process descriptions. The decision for a

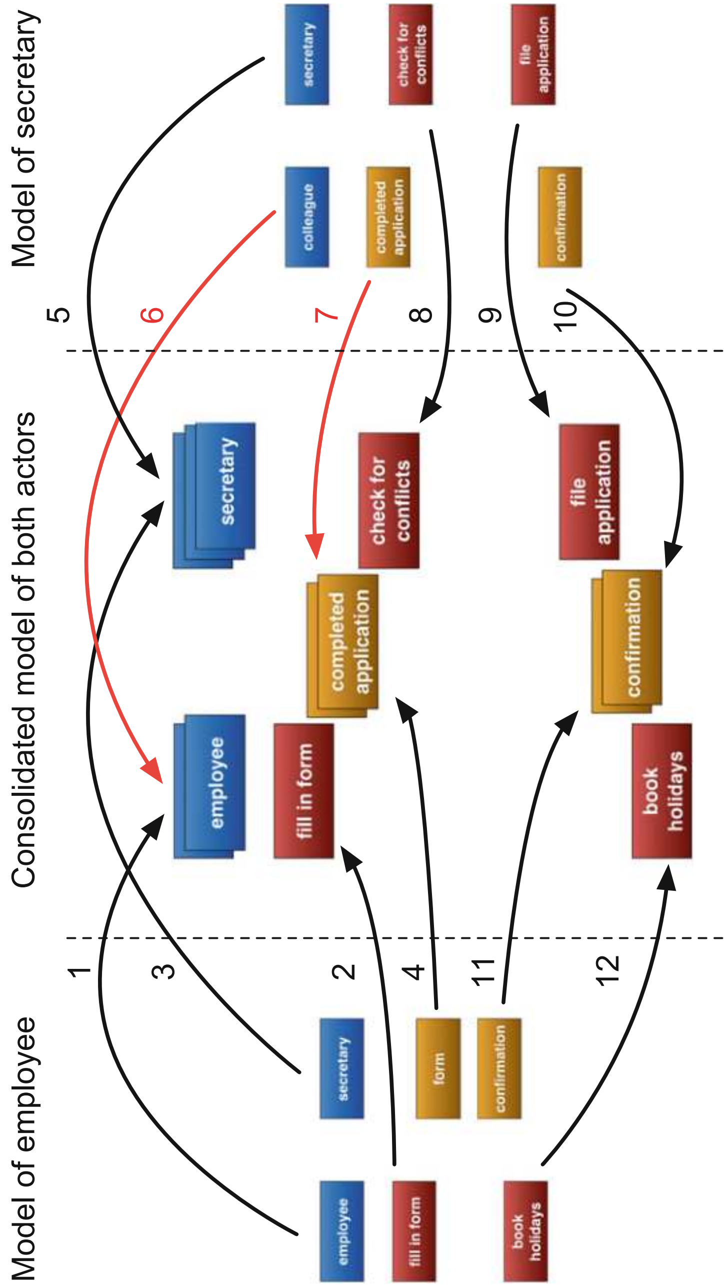


Figure 6.5: Collaborative alignment

semantically limited modeling language was again made from a didactic point of view, since empirical evidence shows that inexperienced modelers can initially describe their views on a work process more simply narratively on the basis of a concrete case. Decisions regarding the concrete implementation of the work process have already been made in the case-based description. Thus, an explicit representation of the same is not necessary in the scope of the modeling. A complete description of the work process therefore requires a multiple execution of the method or its extension by further refinement steps, which however, will not be considered here in detail.

In the sense of the formation of modeling competence, the focus in component 1 is on the introduction to the abstraction and conceptualization of perceptions of the real world necessary for modeling. In component 2, the representation and reflection of one's own perception of work, guided by structural aids, is captured in conceptual models and their description by means of given structural elements. Component 3 subsequently focuses on model understanding (of the other individual models), interpretation (with regard to their effects on one's own model), and negotiation (of the jointly justifiable view) of model content, which ultimately conveys the competence to influence work processes in a self-empowering way.

6.1.3 Raising Awareness of Process-Relevant Change Potential

In the following, the Value Network Analysis [2] is first discussed, as it was introduced in Knowledge Management for the processing of performance relationships between networked actors, before its potential for process design (analysis and modeling) is detailed.

Value Network Analysis

If we consider, as previously mentioned, the added value of organizations and thus the level of performance-related exchange relationships between actors, tangible exchange relationships can be distinguished from intangible ones in the network of actors within the framework of work processes. Tangible exchange is determined by energy and material flows. Intangible exchange, such as knowledge, refers to cognitive processes and action-guiding information. If now participants and exchange relations are described, the structure of an organization or a network can be captured.

Exchange relationships represent the molecular level of economic activities. Thus, value creation does not only consist of tangible transactions, but also of intangible transactions. These relate above all to cognitive exchange, since the sustainable success of an organization is based on the exchange of information, knowledge sharing and open cognitive paths that enable appropriate decisions to be made (and thus the successful existence of an organization). However, knowledge and intangible elements behave differently than physical resources in business life, so they cannot be considered tangibles. Due to their proximity to living systems, they constitute a separate category of exchanges and differ from tangibles related to goods, services and revenues.

Tangible exchange of knowledge is defined as transactions involving goods, services or revenues, e.g., physical goods, contracts, invoices, delivery and receipt confirmations, inquiries, requests, invitations to bid, and payments. It is essential here that knowledge-intensive products and services that generate income and for which payments are made as part of a service or product or on the basis of a contractual obligation are also regarded as tangible transactions.

Intangible exchange of knowledge and performance: Intangible exchange of knowledge and information supports core processes and thus the classic value chain but is not subject to any contractual obligation. Intangibles are 'extras' or (small) courtesies that people exchange in order to build relationships and allow processes to proceed pleasantly or without disturbances. Intangible transactions include the exchange of strategic information, planning knowledge, business-operational knowledge, joint planning activities, collaborative design, and policy development. Intangible transactions are therefore not contractually agreed services for the benefit or support of organizations or their members. They can be extended from one person or group to another, for example, when an organizational unit requests an expert to work temporarily for them in a prestigious position. Recognition often helps in relationship work, so that intangible benefits constitute genuine motivation factors for active participation and engagement in group activities.

Intangibles represent the core of all human action, and thus also determine socio-economic action. Intangible transactions are deliberately seeded. They can be brought about and recognized. If one wants to understand how intangibles generate value, one must first understand how they become visible and work as negotiables in economic exchange relationships. They are often not immediately visible, but rather 'packaged' in services or products. A typical example is to build an understanding for a customer situation (intangible) before offering a service (tangible). For a joint practice, the smooth running of processes is of immediate importance. Thus, those transactions are essential which (also) guarantee by means of intangibles that a common purpose of action is ensured. This must now be methodically taken into account.

In a Value Network, tangible and intangible values are generated by means of complex dynamic exchange processes between two or more individuals, groups or organizations, which represent the object of reflective design.

Holomapping

The view of organizational value generation based on networking brings with it a new form of organizational modeling; every exchange requires a mechanism or medium as enabler for transactions. These can be tools such as e-mail or face-to-face interactions in communities of practice. As already mentioned above, typical intangibles pertain to knowledge to gain information from customers and feedback on (product) developments.

The representation of tangible and intangible exchange processes in a diagram with flow elements allows the mapping of the dynamics of living systems. First the participants or roles (also groups, teams or organizational units, but not technical aids) are documented - they form the nodes of the network and are visualized through ovals. The participants send or supplement so-called deliverables to other participants. Arrows, which are labeled as the respective deliverable, indicate the direction deliverables take in the course of a particular transaction.

Transactions or activities are displayed as directed edges (arrows), which must originate with one participant and end with another. The arrow indicates movement and the direction in which something is happening between two participants. In contrast to participants, who are time-stable, transactions are time-limited and volatile. They have a starting point, a duration, and a conclusion.

Deliverables, on the other hand, are real 'things' that move from one participant to another. A deliverable can be material (tangible), like a document or a table, or immaterial (not tied to matter), like a message or a request that is only verbally delivered. Deliverables can also be intangible, for example, when referring to knowledge about a certain fact (cognitive) or in the case of a favor (social/emotional). Arrows are only allowed in one direction - they cover a single transaction between participants. Bilaterally directed arrows are meaningless, in fact, they make it impossible to analyze the processes and exchange relationships.

An exchange occurs as soon as a transaction results in a deliverable that is returned. It does not necessarily have to be present in the practical world of action in organizations. However, if it occurs, a Value Network can establish itself, with transactions as molecular elements of value generation.

In the context of change processes, it is essential to empower those involved and thus to have the affected role holders create the communication map with tangibles and intangibles (holomap), as well as to process the data collected by them within the framework of the Value Network Analyses.

Within the scope of knowledge generation or knowledge collection at the beginning of the work on the network structure, each individual participant considers his/her role, which he/she then communicates to the other participants. In this way, relationships and interdependencies between the individual roles, which are often unknown, become more explicit and clearer. The roles are symbolized as nodes, the exchange of material or immaterial values is represented in the form of lines connecting the roles. The modeling forms the basis for the subsequent analyses for knowledge evaluation and processing.

Exchange Analysis

The holomap shows how people use their work as a starting point for exchange analysis. Tangibles (material value flows) in the network refer to the material exchange between persons (typically goods, services and sales revenue). They represent transactions based on contracts. Intangibles (immaterial, ideational value flows), in contrast to tangibles, are based on knowledge or a certain additional benefit. They are not contractually fixed or subject to a charge. Intangibles often collected are strategic information, process or planning knowledge, as well as existing emotional components such as mutual trust, common interest, need for knowledge, security, etc. - see also Figure 6.6.

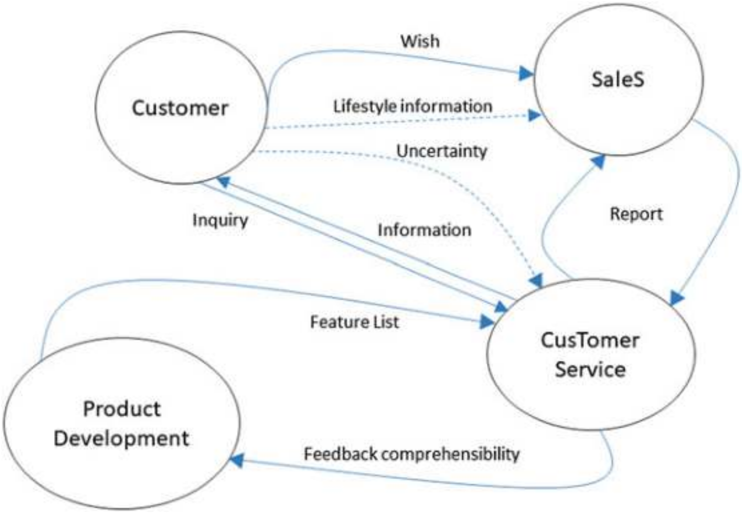


Fig. 6.6 Extract of a holomap for customer service

The exchange analysis examines a Value Network for its conclusiveness, robustness and sustainability. It provides insight into the current structure and dynamics of the network. The following questions should support the exchange analysis: How do the values flow through the organization? Does a certain logic emerge? Is the relationship between the exchange of material and immaterial values balanced or does a certain type of exchange predominate? Does the pattern in the Value Network show reciprocal value flows or are there participants who receive more value flows than they provide? Are there ineffective connections in the network that do not pass on value flows?

These questions are intended to check whether the network fulfills its purpose, whether missing end nodes or links can be detected, and how the structure of the network can be optimized. They ensure a general overview of added value and loss of value. The exchange analysis should serve as a stimulus for dialogue, understanding complex systems, and promoting systemic thinking (cf., [3]). The exchange analysis on customer service, under the assumption that the organization is facebook, shows several findings: Customer service is tangible from the point of view of product development as a sink - it only receives feature list. Sales receives lifestyle information, but no trust-related information. The transmission of uncertainty encumbers the relationship between customer service and sales as well as between customer service and product development. This first evaluation may be an indication of the information management shown in Figure 6.6, where features indeed allow for a certain form of feedback, but where these may be acknowledged by users with requests reflecting uncertainty.

Impact Analysis

Impact analysis examines the impact of each individual value input on the participants and thus focuses on the recipients of value inputs. This analysis thus shows which input triggers which reactions and activities and how this affects the material and immaterial assets of the recipients concerned. The costs and benefits of value inputs are then assessed as low, medium or high.

In order to gain a better overview of these questions, the answers for each individual recipient of value inputs are entered into a table and the current situation analyzed. The table in Figure 6.7 shows the impact analysis based on the insights gained from the exchange analysis of a customer service employee. The table shows who provides input for which activities and what effects are perceived in the form of material or immaterial value flows. The column entries for the general costs and risks as well as for the benefit of the input addressed are essential for the estimation of change potential.

The data from the initial evaluation (exchange analysis) thus form the basis for the two further analyses, whereby value-based detailed evaluations of transactions are carried out from the point of view of input received (impact analysis) and output transmitted (value creation analysis), and in this way provide insights for change processes.

For example, as can be seen in the table, it is explained when features enable a successful form of feedback, for example, to avoid requests that reflect uncertainty on the part of users. This is the case even if the current benefit of the presentation of features by customer service is estimated to be low due to a lack of comprehensibility.

The entries in the feature list table also show the reference point relevant to value creation, i.e., the quality of information provided to customers by customer service employees.

Based on the as-is analysis, strategic perspectives can then be derived and the table can be filled in again and serve as a comparison with respect to its planned, strategic activities (target analysis). In the present case, for example, a customer-oriented information service is of increased importance.

Value Creation Analysis

Value creation analysis analyses how values can best be created, increased and used. Like the impact analysis, the value creation analysis also considers the individual role in relation to the entire system. The difference to impact analysis is that, unlike with input, this time the sender or producer of output is considered in his role and with his related activities.

Each individual sender of value outputs is analyzed to determine how added value and value accumulation are realized in relation to the existing value output. A cost-benefit analysis will also be carried out for this purpose.

In the value creation analysis shown in Table 6.1, the results of the work were used to analyze how values can best be created, increased and used. The table shows the analysis based on data from a customer consultant's exchange analysis.

Customer service		Which activities are triggered by the input?	Effect on the costs and tangible assets of the recipient	Effect on the costs and intangibles assets	What are the general costs / risks of the input?	What are the general benefits of the input?
What do we get (Deliverables)	From which role?	Activity	Value-based effect	Intangible impact	Costs/ Risk	Benefit
Inquiry	Customer	Information	High effort, because translation required from inquiry → feature	Provision of associated feedback for the purpose of understandability to gain knowledge for product development	H in the case of unclear concerns / H for new requirements	M, because not always high, if no constructive feedback is possible
Uncertainty (Intangible)	Customer	Report to sales	Need to act	Negative relation to customer	H / L	H, because essential need for action evident
Report	Sales	Processing	Knowledge generation on customers (Customer Knowledge Management)	Achieving good customer relationships through knowledge of customers	M, depending on content / M	H, as an essential part of information flow
Feature list	Product development	Processing	Enabling customer-oriented information services	Knowledge gain about product	M, because required for product in any case / M, as comprehensibility possibly low when related to technology	H for comprehensible information, otherwise L

L=Low, M=Medium, H=High

Fig. 6.7 Extract from an impact analysis for customer service

Table 6.1 Part of the case-related (customer service) value creation analysis

Output of the sender	Output Recipient	Added or increased value of the activity	Costs / Risks	Benefits
Feedback - comprehensibility	Product development	Customer-oriented access Consideration of feasibility	H / H	H
Information	Customer	Dealing with customer needs Would also be interesting for potential customers	H / H	H
Report	Sales	Acquisition of customer data Would also be interesting for Training Department	H / H	H

For costs / risks and benefits: L=low M=medium H=high

Performance indicators can be used to filter out the requirements placed on sales and product development from the analyses. The strategy developed from this can be outlined with the statement 'availability or transparency of information'. As a change measure, it could be decided to increase the identified value creation potential in the Value Network by expanding the tangibles of information flows between all functional units.

After the presentation of the actual situation, the value creation analysis also makes it possible to derive strategic perspectives. Here the question should be asked 'Which possibilities should still be used in the future to generate value optimization in value output?' With regard to the concrete question 'What should be done to increase, expand or optimize the value of output?', the table is filled out in a comparative manner to analyze the target situation.

The entries in the 'costs/risks' and 'benefits' columns are essential for decision support. Here, the assessment can be put into perspective, especially with regard to costs and risks in the target situation, if, for example, content should be included in training courses, since there is know-how for the creation of corresponding training materials and for the provision of effective mediation formats in the network (e.g., in the context 'Report' - third table entry in the table). A target list usually takes into account knowledge about the feasibility or availability of resources and the associated costs for implementing measures, without anticipating a decision in this regard (see evaluation).

Evaluation

For the evaluation, tangibles and intangibles are evaluated in tabular form (impact analysis, exchange analysis, value creation analysis) according to their significance for the respective role(s). These evaluations reveal the effects on relationships and allow targeted measures to be taken.

So far, the exchange analysis has provided insight into the current structure and dynamics of the network. The implementation of the impact analysis allows all participants to deduce their own roles in the network in a context-sensitive way. The impact analysis also provides an overview of the effects each individual value transaction has on the participants. The value creation analysis allows decisions to be made on how values can best be created, increased and used, and how they possibly affect other roles or should include them in the consideration. Based on the derived performance indicators, a strategy can now be developed to increase the identified value creation potential in the Value Network.

Typical results of an evaluation to increase added value include the consideration of missing exchange relationships, such as the ones related to the case in question:

– *Concrete inquiries to customers in* order to better understand their concerns (especially those that lead to customer uncertainty) from the point of view of product development and sales - customer service plays a mediating role here, whereby the tangible report can also be upgraded and the feedback can contain concrete suggestions for the improvement or creation of features.
– *Feedback from customers* concerning both the future handling and the previous handling of features, i.e., those issues which affect product development in particular. As soon as the comprehensibility of the presentation is explicitly addressed, an associated flow of information (back) to product development (and also to sales) can be ensured - another measure that facilitates customer-oriented access to the product.

Both measures show the necessity of networked representation of exchange relations. It is not necessarily the immediacy of exchange relationships, but rather the concatenation of exchange relationships that can bring about added value. If, for example, product development (e.g., software designers) were to start communicating directly with customers, a certain basis for discussion would first have to be established. Such measures could even be counterproductive to the strategic goal to be achieved and increase the existing uncertainty on the part of customers, thus adversely affecting the desired objective.

The additions to the network thus allow a context-sensitive system view of value creation through material and immaterial services that should flow between the actors involved. The completed Value Network map, as shown in Figure 6.8, forms the basis for further development planning. For the first time, it shows the inquiry to customers as well as the feedback for the purpose of comprehensibility as tangible deliverables, which are intended to increase the knowledge of those involved through the transparency and availability of information. In the long term, as an intangible exchange between customers and customer service, it is important to strive for a mutually secure relationship that can be expressed through customer loyalty to the organization. Only the open exchange of information enables sustainable customer knowledge management.

Value Networks view systems in their entirety and consider their complexity. They enable the holistic identification of both material and immaterial values. In any case, the latter indirectly determine the quality of material exchange relationships and must therefore be taken into account in the development of socio-technical systems. By means of Value Networks the focus on the individuals involved can be captured, which in turn gives them a sense of purpose and fosters their motivation to participate in reflection and actively take part in codesign. The surveys and analyses

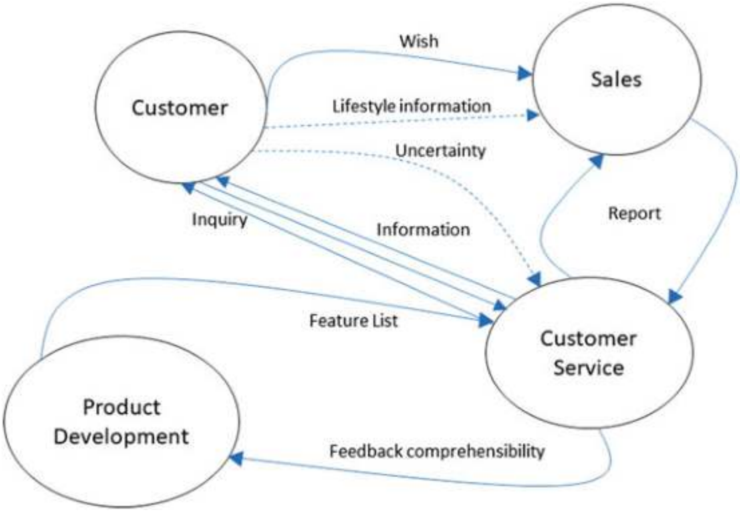


Fig. 6.8 Adapted network

presented here allow the explication of individual roles and their directly or indirectly perceived contribution to the value creation of an organizational system by actors. This facilitates the clarification of roles and the understanding of correlations, since these are visualized graphically in holomaps by means of value exchange relationships, and are thus fed back role-specifically. As such, they represent a viable starting point for the participatory design of socio-technical systems.

Potentials for Process Analysis and Modeling

For the design of processes and their modeling, multiple potentials can be drawn from the Value Network Analysis:

– The Value Network Analysis allows the representation of a situation as it is perceived by actors. The representation is based on roles, which are illustrated by nodes of a network of actors. Since each role holder can perform this analysis, individually perceived interaction patterns can be compared with the perception of other role holders. Thus, the Value Network Analysis allows the structured identification of differently perceived interaction patterns between actors from their individual point of view, whereby the different interaction patterns can also be presented cumulatively.
– The recording and presentation of the actual situation represents a reference point from which change potentials can be tapped. This makes it possible for all participants to discuss changes in interactions on the basis of a common starting position in a way that is comprehensible.
– The Value Network Analysis allows the determination of roles that are not necessarily functional in an organization, for example in the organization chart. The definition of roles is based on the communication and interaction patterns

which are considered relevant within the framework of the fulfillment of tasks and the perceived organizational events. Thus, the focus is placed on the way of working and the interaction level.

- Within the framework of the development of proposals for change, instead of demands from individual role holders, proposals are made to others in the sense of individual offers to the collective. Such an approach gives the addressee the choice of whether or not to exploit this potential. All offers are presented in the context of their origin and evaluated with regard to their organizational effectiveness by the respective proposing actors. They can then be coordinated collectively on this basis.
- Interaction relationships are differentiated according to contractually binding services (tangibles) and not only from the respective role according to contractually regulated services (intangibles). This already makes it evident how tasks are handled, whether more so according to formal principles or interactions that promote successful process fulfillment, or according to informal principles. The same applies to proposed changes, which may also be in formal structures (as part of functional role descriptions), or at informal level (as voluntary contributions).
- There are several ways to change an actual situation into a target state: (i) an informal (intangible) interaction becomes a formal part of a role (tangible); (ii) a formal (tangible) interaction is omitted or becomes an informal part of a role (intangible); (iii) a tangible or intangible relationship is newly introduced and complements existing interaction patterns.
- Interaction relationships can be directly transferred into concrete process steps, as they represent the fulfillment of tasks by the role holders in chronological order. Thus, a process model, which focuses on the exchange of services between actors (in contrast to linearized functional steps), can be derived from a holomap.

Overall, Value Network Analysis is a diagrammatic / tabular technique for organizational development with the goal of process definitions or executable processes, which directly supports the mapping of interaction structures to communication-oriented approaches such as S-BPM ([4]). All information is generated from the point of view of the involved or responsible role holders (see also [5–7]).

6.1.4 Structured Asset Records

In the following a procedure is described, which leads from a natural linguistic description to a formal behavior model. This approach is based on active sentences of natural languages. The process will be introduced on the basis of the Poly Energy Net (NET) project, an approach funded by the German Federal Ministry for Economic Affairs and Energy. The aim of this project was to develop a solution for a self-organizing distributed energy supply system. The following sections show how the associated software was developed, from a general description to a precise model, which was then converted in a first stage into a program based on process specifications.

General Information

The following steps, which lead from an informal description of a process to a formal model of the process flow, do not have to be performed in the order given. The steps rather serve the goal of a precise process description. If it turns out in one step that something had been forgotten in a previous step, or if it turns out that it is more advantageous to design subprocesses differently, this change is included in the step currently being worked on. The change is not reflected in the previous descriptions. To create a first draft of a process description, the procedures from Design Thinking described in the previous chapter can be used.

All documents of the previous descriptions are obsolete by default and no longer valid after completion of a more detailed description, except when a description explicitly states that a previous description is still valid (e.g., as an overview document). Experience has shown that it is not possible to keep several documents consistent. There should therefore only be one valid document, so that a formal behavioral model is available after the preparation has been completed. As a rule, changes should only be made to this model.

When creating a process model, one can also start with any step. For example, only active sentences can be used for a natural language description. Thus, the steps which transform a description in arbitrary form into an active description form are omitted. It can also begin with the identification of the actors and continue with the detailing of each actor, including communication with other actors. The concrete procedure depends on the circumstances and preferences of the parties involved.

Natural Linguistic Description of Processes

A process is described in a more or less structured way in natural language. There is no specification as to the structure of the document to be created or the vocabulary to be used. The creators of a process description can follow their preferences. Figure 6.9 shows an excerpt from the rough description of requirements for the energy management system.

The initial free use of natural language does not require any special methodological knowledge on the part of the participants. The need for such knowledge could be a major obstacle to the involvement of stakeholders from different departments.

Textual descriptions can be supplemented by suitable images. Figure 6.10 shows a functional structure of the system to be created. Such a functional structure is a first approximation to the specification of a process system.

Based on these more structural descriptions, a first process-oriented specification can be created. Figure 6.11 shows an excerpt of a process description.

This process description is supplemented by an illustration of the effects on a holonic energy network. Figure 6.11 refers to some elements (switches) in Figure 6.12.

The tools used at the beginning for an initial requirement definition and process specification are not structured. Basically, texts and supplementary drawings of any

The holonic model

- The holonic model is a logical system that can be mapped to a physical system in the real world.
- The entire holonic supply system consists of holons.
- Within each holon, the same amount of energy is provided ("generated") as is consumed at any time.
- Holons consist of holonic (energetic) elements. These are:
    - holonic production and consumption elements
    - holonic connectors
    - holonic conversion elements
    - holonic storage elements
    - holonic guardians
- All holonic elements have a physical instantiation.
- Holonic elements have the ability to communicate with at least one holon manager. They are connected to at most one holon manager (to a holonic object).
- Holon managers can influence the behavior (e.g., generation, consumption) of holonic elements via control signals. All holonic elements that can communicate with the same holon manager form one holonic object.
- In each holon, there is an instance holon coordinator that can communicate with holon managers in its own holon and holon coordinators in other holons.
- Holonic objects can exist without being connected to a holon.
- Holonic elements can exist without being assigned to a holon manager ("free flying holonic elements").
- All holon managers and holon coordinators have a physical instantiation.
- Holons can dynamically merge into new holons or disintegrate into smaller holons according to certain holon rules. The rules are implemented by holon coordinators in interaction with holon managers.
- The rules for the holon formation are based on logical requirements and determining factors, which are defined by the physical instantiation of the holonic elements.
- Holons can dissolve. This is the case when there is no holon coordinator.
- There are also "non-holonic" elements, i.e. those that can neither be accessed nor are connected to the energetic system.

Fig. 6.9 Requirements for a holonic energy management system

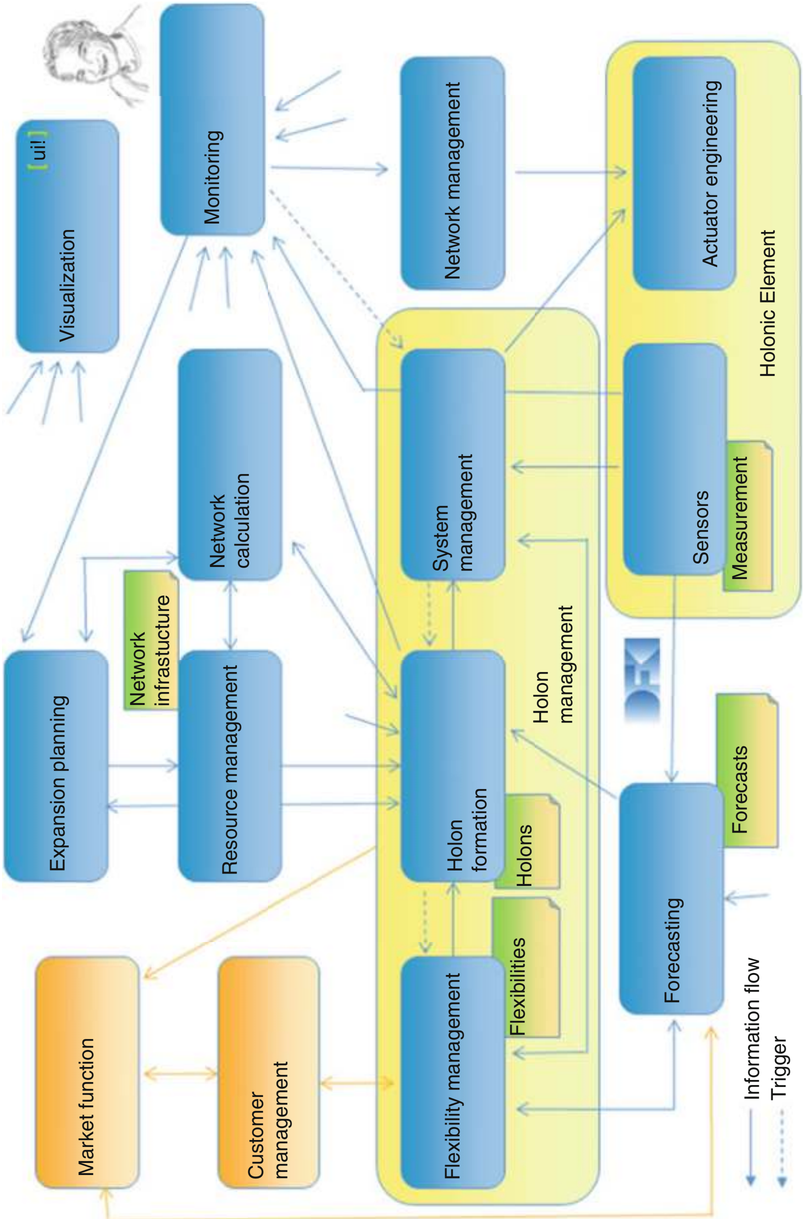


Fig. 6.10 Function blocks of a holonic energy management system

	in the holonic system
Trigger	Due to the failure of the local network transformer (in example A), a single network section in the low-voltage range can no longer be supplied by the higher-level medium-voltage network. This is detected by measuring and reporting a voltage drop to near zero in the affected network section (cf. PMU in Fig. 2.). It is assumed that the distribution bus bar in the local network stations is intact and that only the transformer as such has failed. The control center or network supervisor automatically reports the "voltage drop" result to the control station, which triggers the fastest possible supply (minimum possible supply of a consumer) with a process largely characterized by automated measurement and remote effects.
Procedure	By automatically querying measured values (at the PMUs in the network) and other fault information (e.g. from the involved local network stations involved), the control center identifies the location and, as far as possible, the cause of the fault. In the current case, it is determined that the transformer at A has failed. The control center informs one or more Holon Managers (HM) or Holon Coordinators in the affected network area. The Holon Coordinators and Holon Managers concerned communicate with each other and negotiate (also with neighboring Holon Coordinators) a new formation of holons which will enable the supply of as many consumers as possible even without the defective local transformer. The aim is to maintain such a supply (even without an emergency power supply system) until defective parts of the local network stations have been replaced. In the situation outlined in Fig. 2, it can be assumed that the holon management of the control center will propose the following new formation (cf. Fig. 3): • The holonic lines are to be switched as follows: o Switch 4 is to be closed and thus connects the distribution bus bar in the local network station A, which was previously supplied by the now defective transformer, to be supplied. o Opening or closing switches 5, 6 and 7 creates an island separated from the rest of the system. o Opening or closing switches 8 - 11 creates an island separated from the rest of the system. • The following holons are formed: o The island separated by switches 6 and 7 forms holon C and supplies itself. Apart from a domestic PV system, the only source of supply is the CHP, which is provided as an "emergency power generator" for a very critical consumer. The holon is to be operated in such a way that the latter is always fully supplied, further consumers can expect only a minimum supply. (Note: If this supply situation cannot be maintained, Holon C would probably soon disintegrate into smaller, sometimes very poorly supplied holons, or some of the consumers would be added to holon A). o The island separated by switches 8 and 11 forms holon B and supplies itself. Here, an extensive full supply is to be expected, since there are efficient branches in the holon (large BHKW, open-area-PV-plants, quarter storage). o All other holonic elements form holon A. Apart from a smaller BHKW and few domestic PV plants and associated domestic storage the local network station B secures the supply. If the control station accepts the holon formation, it initiates the corresponding switching processes and settings. The operating personnel is informed, drives to the local network station and disconnects the defective transformer from the distribution bus bar. The situation caused by the holon formation allows an optimal, asfar as possible full supply of all consumers with sufficient energy source (sun, gas) until a replacement transformer can be installed or the existing one repaired. Note: If a supply as enabled by the new holon formation is not possible, this is recognized as an error situation and a new holon formation is triggered. As ultima ratio also here a net-replacement plant would also be used here.
Result	After the fault has been rectified, the operating personnel on site, including the network control center, restores the normal state (transformer supplies the distribution bus bar in local network station A). The measurement in the control center recognizes that a new, strong supplier is available, which triggers a new formation of holons. Note: This will likely lead to the situation involving holon A and B like before the occurrence of the use case.

Fig. 6.11 Dynamics of a holonic management system

kind can be used. In the following steps, this non-technical description of a process is transformed step by step into a precise description of the process flow.

Process Descriptions in Active Form

Informal process descriptions in natural language very often contain passive clauses (see also the above description). However, passive sentences do not contain a direct assertion about the performer of an action. Passive sentences are used when it is not important who the performer of an action is. However, this is not the case with processes. Process descriptions must include the performer of an action. All passive sentences must now be converted into active sentences. To do this, the active elements must first be identified. Active elements can be humans, software systems that run automatically and perform certain activities, physical systems, or any combination of these basic elements. Therefore, in our example, the control center can be a combination of software, people and electrical systems. The software prompts an operator to read a specific measured value and enter it. Depending on the entered measured value, the software initiates the closing of a switch.

In order to avoid a process description being too dependent on the organizational and technical environment, abstract actors are introduced. Such abstract actors are entities that send messages, receive messages, or perform internal tasks. Figure 6.13

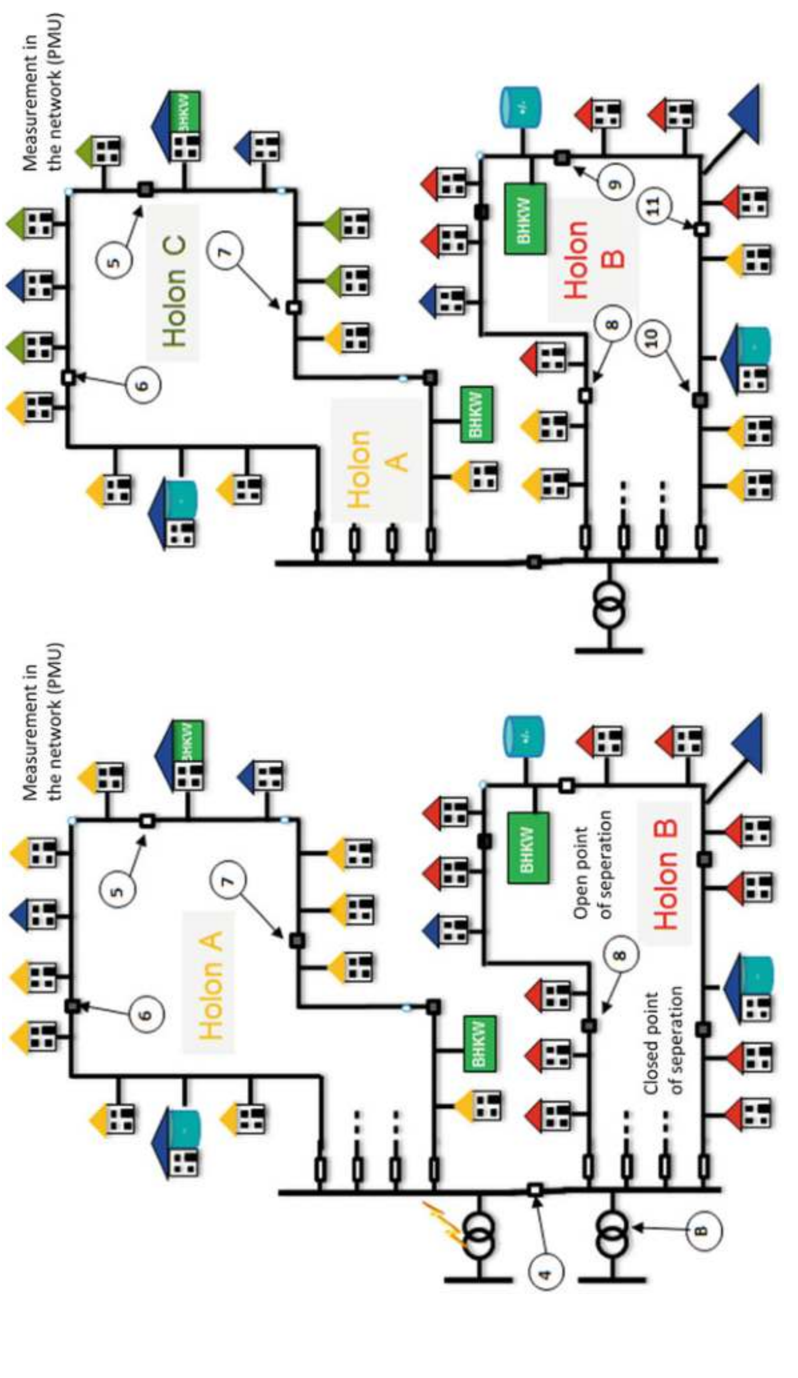


Fig. 6.12 Example of a holonic energy management system. Source: B.A.U.M

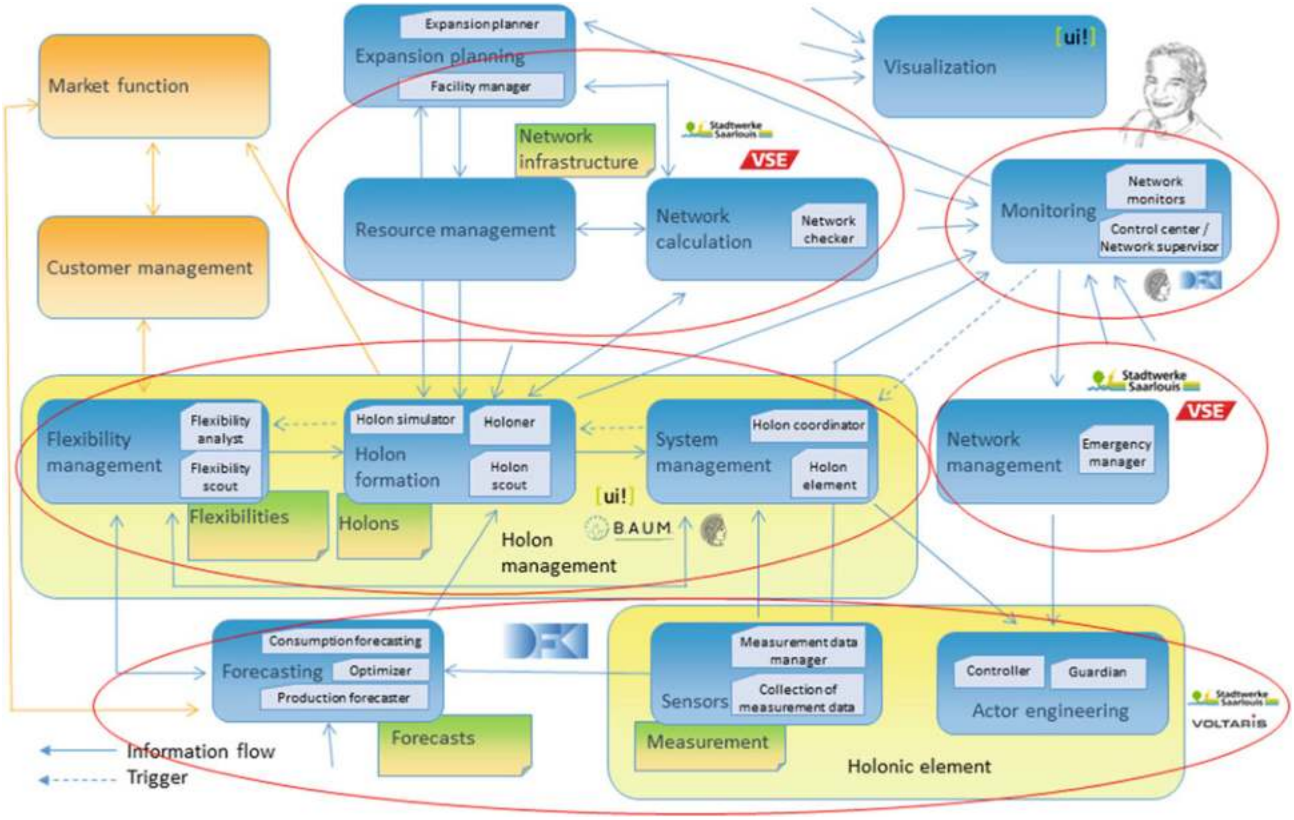


Fig. 6.13 Subject-oriented consideration of a holonic energy management system

shows the function diagram with assigned active elements. These abstract active elements are called subjects, following the subjects in active sentences. Subjects are the roles in processes.

The tasks of the identified subjects can be briefly described for a better understanding as shown in Table 6.2.

With the introduction of actors in the form of subjects, all passive sentences can now be replaced by corresponding active sentences. This makes the process description more complete. Table 6.3 shows a process description with active sentences in tabular form. The numbering of the entries already reflects the control flow. The number of the follow-up action is specified in the associated column. If the follow-up action depends on certain results of the action, this condition is described in the column follow-up action. Depending on the valid condition, there may be another follow-up action.

A table with the control flow of a process can be the starting point for a control flow-oriented process model. The individual actors are the swim lanes in BPMN or in a swim-lane-oriented EPK or in UML state diagrams. However, these modeling methods should only be used if no asynchronous events, such as the possibility of changing purchase orders, occur in a process and the parallelism of the agents is not to be modeled. In addition, the number of actors should not be too high. Swim lane representations are usually flat, i.e., there are no hierarchies of swim lanes. More than five swim lanes lead to confusing representations. Thus, a service process contains as a minimum the swim lanes customer, call center, first level support, billing and, where applicable, customer feedback. Experience shows that in real processes there are usually about 10 actors or more involved. Swim lane diagrams even become confusing if the control flow just has to cross several swim lanes in order to switch to another swim lane.

Control flows are not a manageable representation for cross-organizational or cross-company processes in a distributed environment. In a distributed environment, messages are the more vivid way to model the collaboration of individual actors.

Tabular Role-Oriented Description

In the last step before the actual process modeling, the process description is structured according to the actors. All sentences with the same actor as subject are summarized in a table. For this purpose, it may be necessary to supplement the process description with interactions between subjects. Phrases such as "informs subject xy" or "engages with" etc. are replaced by send and receive actions (see Table 6.4). Sentence no. 2 "Network monitor reports the situation to the control center" in Table 6.3 is converted into a send and a receive action. In Table 6.4 this corresponds to sentence no. 2 “Network monitor sends status black to control center" in the table section for the network monitor. The counterpart to this is sentence no. 1 in the table section for the control center.

After creating the behavior tables, a formal model can be derived in a suitable modeling language. A language should be used in which the aspects deemed important for the process under consideration can be clearly and precisely expressed. In our example we have selected S-BPM. Figure 6.14 shows in the left half the

Table 6.2 Tasks of identified subjects

Name	Type (person, organization, component, system, subsystem, application)	Description	Specifics in this Use Case
Holon manager (HM)	software service	The role of a HM describes a set of functionalities that are provided by one or more software components.	Examples of use-case-relevant functionalities are: – Identifying a holon coordinator – Acting in the role of holon coordinator use case relevant functionalities
Holon element (HE)	component	Holon elements are production or consumption elements, conversion elements, storage elements or line elements. A HE always has a connection to the energy system, but a connection to the ICT system is optional, although the rule. A HE has at least one (usually IT-technical) control element, usually also the ability to provide information.	Examples of use-case-relevant functionalities are: – HE can send status information – HE can receive control commands
Holon coordinator (HC)	software service	Holon coordinators perform the basic functions for operating a holon. They control holonic elements in their own holon using the functions of a holon manager.	Examples of use-case-relevant functionalities are: – Determination of a permissible holon (if necessary, taking other goals into account). Criteria are still to be defined – Determine switching commands for establishing a holon – Initiate switching commands in a holon and monitor their implementation.

(continued)

Table 6.2 (continued)

Name	Type (person, organization, component, system, subsystem, application)	Description	Specifics in this Use Case
control box	system	A control box can receive and implement switching commands for holon elements.	
Control center	person/Organization	Monitors networks and initiates measures to ensure optimal supply	Examples of use-case-relevant functionalities are: – Detection of network states – Communication of network states
Network monitor	software service/ system	Monitors networks by means of specific services	Examples of use-case-relevant functionalities are: – Detection of network states – Communication of network states

network structure of the considered process. The rectangles with rounded corners represent the actors involved. The arrows between the actors are labeled with the exchanged messages. The numbers on the message names correspond to the sentence numbers in the table above. The diagram to the right of the process structure shows the behavior of the holon coordinator subject. The circles with the letters E and S are states of communication. Transitions from circles with an E are labeled with the messages expected in this state. The transitions to S states are labeled with the messages sent in this state. All other transitions define local operations on local data.

6.1.5 Process Modeling

Selection of the Modeling Language

In the following, some factors which should facilitate the selection of a suitable modeling notation or language are listed as examples. These include the determining factors of a BPM project, the manageability of the language and the support of downstream activities such as validation (see also [8–13]).

With regard to *determining factors,* the question 'What are the properties of the subject matter to be modeled?' is of particular interest. The following properties should be decisive for the selection of a modeling language:

Table 6.3 Sequence in the Holon System

No.	Task	Indication	Follow-up action
1	Network monitor detects voltage drop in the low-voltage network	Uses functions from the function group "Measurement" for this purpose.	2
2.	Network monitor reports the situation to the control center		3
3.	Control center identifies and localizes the error	Uses functions from the function groups "Monitor" and "Measurement" (Remark: What happens if the control center cannot identify the error?)	4
4.	Control center informs holon managers (HM) or holon coordinators in the affected network area	Attention: a single-point-of-failure or bottleneck may be present here; if necessary, it should be replaced by a decentralized mechanism.	5
5.	(where appropriate) Holon coordinators contact holon managers in the affected holons	Uses functions from the function group (holon management) for this purpose	6
6.	(if no holon coordinator takes action) Holon managers identify a HM to take over the coordination task		7
7.	Holon coordinator and affected holon manager identify a plausible new holon constellation	Uses function "Holon formation" of the function group "Holon management"	8
8.	Holon coordinator reports the calculated proposal to the control center		9
9.	Holon coordinator goes into "alert" state and waits for feedback from control center		10

– Asynchronous events: If such events exist, then a corresponding modeling language should offer the possibility to map the associated parallelization of processes or process steps in such a way that, if necessary, an execution reflects the parallelism. This means that the modeling language contains constructs for representing asynchronous events that explicitly contain this temporal quality. This allows a runtime environment to be configured accordingly at execution time.
– Cross-organizational and cross-company: If an issue is relevant not only for a particular organization or one of its subsidiaries, but also for networked partners outside the immediate areas of activity, such as other subsidiaries or the supplier industry, then proven modeling mechanisms or language constructs should be available that enable the encapsulation of external issues. This makes it possible to embed external actors in an organization without having to know and represent their processes in detail.

Table 6.4 Sequence in the holon system (more precise)

No.	Subject (actor)	Verb	Object	Indirect Object	Outcome	Continue with
Network monitor						
1	Network monitor	measures voltage in the	low voltage network		voltage drop	2
					voltage rise	3
2	Network monitor	sends	(1) status black	to control center	sent	1
3	Network monitor	sends	(10) status green	to control center	sent	1
Control center						
1	Control center	receives	(1)status message	from network monitor	received	2
2	Control center	identifies and localizes	the error		error found	3
					error not found	?
3	Control center	sends	(2) status	to holon coordinators	sent	4
4	Control center	receives	(5) proposal	from holon coordinator		5
5	Control center	checks	proposal		accepted	6
6	Control center	sends	(6) V-accepted	to holon coordinators		7

– Number of actors: At first glance, this parameter does not seem to be necessarily relevant for the selection of a modeling language. However, it becomes more important when it comes to the complexity of processes on the one hand, and the comprehensibility of the models on the other. After all, the instance-model relationship also plays into this factor. If there is a large number of actors involved, a notation should also have modeling mechanisms or language constructs that make them visible, as well as accessible aggregately or through other perspectives (e.g., functional view).

With regard to *manageability,* the question 'How should the subject matter be described?' is of particular importance. The following properties or quality of a modeling language should be decisive for its selection:

– Number of symbols: On the one hand, a limited number of symbols can make powerful models easily manageable, but on the other hand it can lead to an undesirable abridgement of facts.

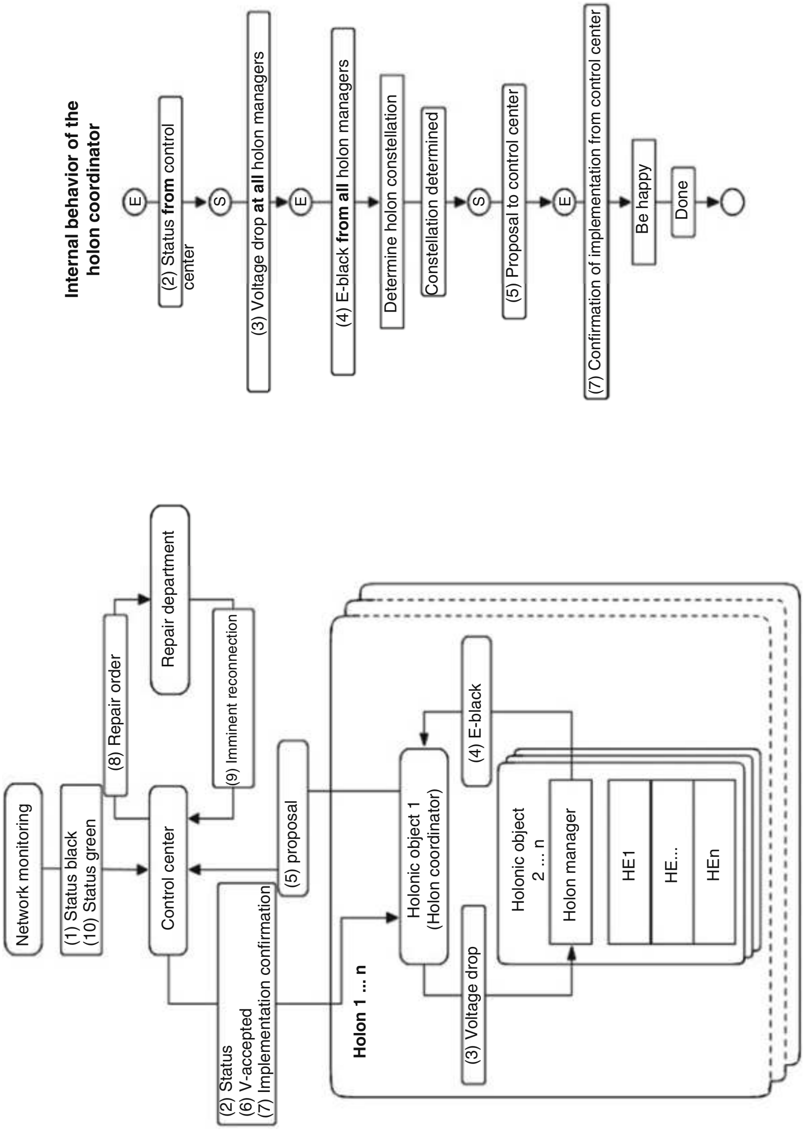


Fig. 6.14 Behavioral description of the holon coordinator subject

– Definition of symbols: The use of the symbols should be clear, i.e., objectively comprehensible for modelers. This facilitates effectiveness and efficiency in the creation of models.
– Availability of tools for model description: The availability of digital tools for the simple and correct representation of subject matter determines the usability of a modeling language. A syntax editor for diagrammatic languages helps, for example, to create syntactically correct models and to structure complex situations in a usable way.
– Possibilities of structuring models along a hierarchy: This feature allows networked issues to be viewed from a top-down perspective. This usually facilitates the legibility of models and thus increases the comprehensibility of illustrated facts for those not directly involved.

Regarding the *support of further activities,* the question 'How is the support of the model for next steps?' is of particular interest. When selecting a modeling language, the following features of a modeling language should be considered in this context:

– Validation tools: Can a model be validated? This means that a tool can be used to determine whether the notation and thus all of the language used has been utilized correctly in the sense of the syntax of the language and in the sense of the intention of the subject matter depicted.
– Optimization tools: Can a process be optimized with the help of its model? Optimization follows initial modeling and validation and aims at the optimal distribution of tasks and the optimal use of resources. For this purpose, a tool should be used to determine whether the notation and thus all of the language used allows, or actively supports, the optimization of modeled processes through corresponding constructs of the language, or through special mechanisms (e.g., through suggestions or reference models).
– Tools for organizational embedding: Can the integration of a modeled process into an organization be realized with the help of a tool? This step is the first one to implement processes and requires that a tool can be used to determine which task or role holders or which organizational units should be able to carry out the illustrated activities in practice. On this basis, the corresponding assignments for the organizational implementation of process models can then be made (and changed again as required).
– Tools for technical embedding: Can the integration of a modeled process into an IT infrastructure or information system architecture be realized with the help of a tool? This step is the second necessary step for the implementation of processes and requires the determination by means of a tool which technical system can carry out the depicted activities in practice. On this basis, the corresponding assignments for the technical implementation of process models can then be made (and adapted again as required).
– Tools for commissioning and operation: Can the commissioning and operation of a modeled process in an organization be supported or ensured with the help of a tool? This step is necessary as soon as a process is switched to 'productive',

i.e., after it has been embedded in the organizational and technical infrastructure and has been transferred into the operational environment or has become operationally effective. In this context, a tool should help to support the introduction phase of a modeled process, i.e., to determine which process steps are transferred to operations and in which order. An appropriate tool should also support monitoring of implemented processes and thus help to ensure the operations of an organization and effectively support its further development. The latter can be done by means of annotations in process models, which mark obstacles to execution.

Modeling by Construction

When constructing a process model, modeling begins with a "blank sheet of paper". The information from the process analysis is used to describe the process step by step. The required activities include several different tasks, depending on the approach chosen:

- Description of the processes and their relationships (process network)
- Identification of the process to be described
- Identification of the actors or systems involved in the process
- Specification of the information exchanged between the actors or systems within the control, data and message flow for processing business cases. This also includes the implementation of business rules, as they directly influence the behavior of actors and systems.
- Description of the behavior of the individual actors or systems, in particular through functional steps and their temporal-causal relationship according to the modeled subject matter.
- Definition of business objects or data and their use.

These activities are set in a certain order according to the selected language and lead to differently detailed representations of subject matter.

Modeling by Restriction

In addition to modeling by construction, there is also the possibility of modeling by restriction. This is based on general process models. A typical example is the communication-oriented approach as pursued with S-BPM. In the universal process model, each actor or system involved in a process can send a message to, or receive a message from, any other actor or system involved at any time. This message has the general name 'message' and can transmit any media as a business object. The result is a universal process that is characterized by the number of its subjects. These are marked as boxes with subject 1... n. Their mutual interaction possibilities are marked by arrows between the subject boxes. This results in a similar initial behavior for each subject.

Within the framework of modeling by restriction, the following steps are taken to a detailed factual situation: (i) determine number of subjects and subject identifiers,

(ii) reduce communication paths, (iii) specify message types, (iv) customize subjects' behavior, (iv) specify and refine business objects.

If other, for instance, function-oriented approaches are chosen, then basic structures can be generalized, for example in the form of reference models. In doing so, behavioral patterns will also be used, such as previous events for functions or conditions that determine the further course of business processes after executing a function.

Combining Approaches

While in the construction of process models the modeling begins with a "blank sheet of paper" and is extended step by step with information from the process analysis, in the modeling by restriction a generalized structure of process models and their components is assumed. A typical example of a combination of both approaches is the case corresponding to the middle-out approach. One instance of this case is to start with a construction and, as soon as a (recurring) pattern occurs, to use a reference model and reduce or concretize it.

Further application instances would be a pattern comparison and the start with recurring (routine) processes. The latter reverses the above-mentioned case and allows the embedding of special characteristics of process flows into generalized process architectures. The pattern comparison, on the other hand, represents a kind of control process in which the completeness or correctness of a model can be checked by means of a generalized pattern.

6.2 Quality Control: Validation and Optimization

Validation is closely related to modeling. In modeling the process flow is described according to the objective. This means that during the modeling activity the following question resonates: 'Does the model correspond to the set qualitative and quantitative objectives?' The examination, whether a process model corresponds to the set goals, is called validation and/or optimization. This means that validation and optimization are constantly performed in the course of modeling. After the decision to complete the modeling process, a final check is made as to whether the overall model meets the set objectives.

Validation shows whether the process meets all requirements and achieves the intended results. It is also essential for a process whether the desired results are achieved with the least possible effort. Quality control in business processes therefore has two main tasks. It is intended to test the effectiveness and efficiency of processes. Effectiveness means that the process meets the requirements placed on it, i.e., delivers the desired result (output). The process is efficient if it can be carried out with as little financial and time resources as possible in order to deliver the desired result. This is to be achieved through optimization.

Both quality controls must be implemented as early as possible, before IT systems are developed at great expense and later users are trained. Figure 6.15 summarizes the individual aspects of validation and optimization. The verification of a process

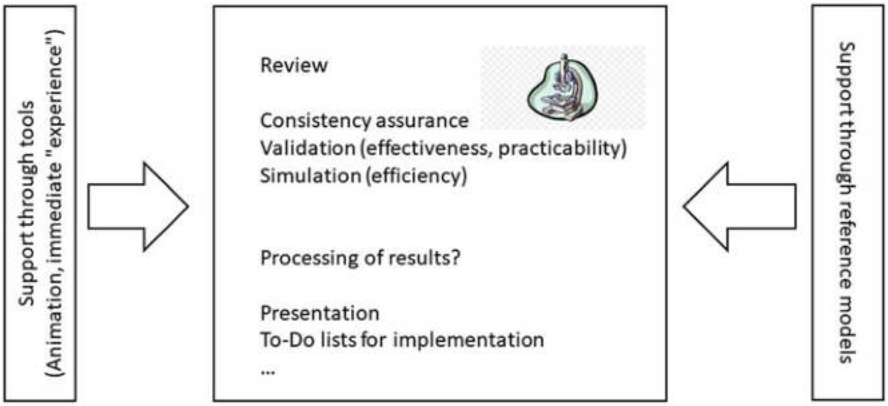


Fig. 6.15 Structure of the quality assurance of a process model

model is supported by appropriate tools and reference models. For the validation there are manual tools such as checklists or role-plays, while for the optimization simulation software must be used. The results of the check are prepared and entered into 'ToDo' lists for processing, i.e., modeling activities are started again. This cycle is repeated until the verification leads to a result considered to be good enough.

6.2.1 Validation

The prerequisite for validation is a model that reflects the subject matter to be represented. It is checked whether the model delivers the expected result according to the specified quality characteristics and whether the process contributes to the goals of the company. This aspect is called semantic correctness. This results from the consensus of the managers as well as the technical and methodological experts who consider the model to be appropriate.

The semantic validity is to be distinguished from the syntactic correctness, which concerns the adherence to the fixed description rules, i.e., the description means are used according to the defaults of the modeling language.

Manual Process Validation

Figure 6.16 shows a general procedure for manual validation. Here the process documentation is checked with the help of checklists. The process documentation includes the description of the goals, inputs, results, triggering events, and of course, the model of the process. The process documentation should be checked by all parties involved in a process on the basis of the checklists.

The findings of the individual parties are consolidated and clarified in a joint workshop and the necessary revisions are jointly determined. This cycle is repeated until it is jointly decided that no further revisions are necessary.

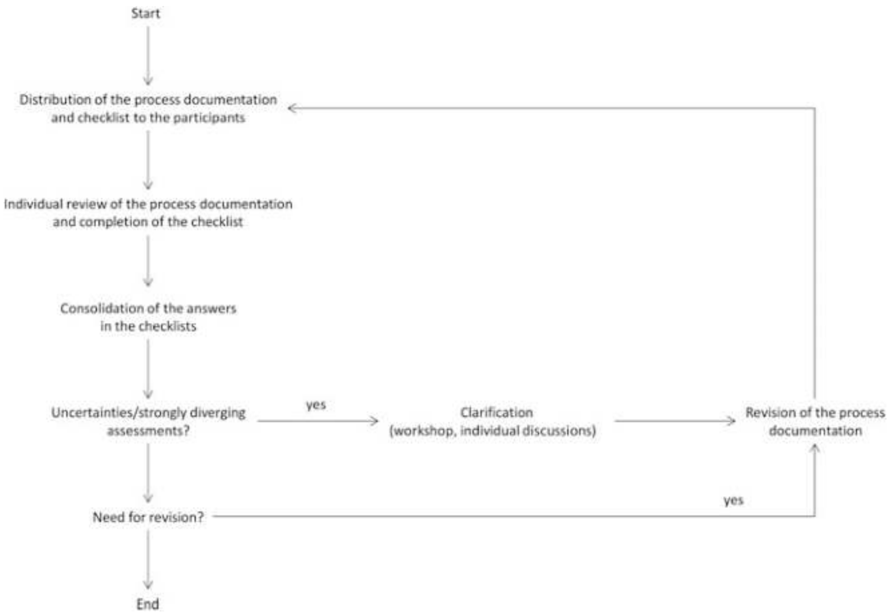


Fig. 6.16 Sequence of a manual process check

To prepare a review, the process description and a **checklist,** according to which the process description is to be verified, are distributed. This checklist contains questions to be answered by the evaluators regarding the process.

Examples of such questions are:

– Does the process support the company's goals?
– Are the objectives of the process defined?
– Is the benefit of the process clearly described in the objective and is it clear what added value it delivers and for whom?
– What risks does the process entail?
– Is a process owner assigned?
– Have the authorities of the process owner been defined and are these sufficient?
– Are there any performance indicators with which the achievement of objectives can be evaluated?
– Are the measurement procedures for the performance indicators clearly defined?
– Are the target values for the performance indicators of the process systematically defined and do they provide an assertion about the value contribution of the process?
– Does the process support the policy and strategy of the company or IT organization?
– Is the process flow described?
– Are the inputs and results of the process described?

- Is it clear who (organizations, roles, people) provides which inputs and who receives which results?
- Are the description conventions for processes adhered to?
- Is it defined who is responsible for the individual steps of the process (organizations, roles or persons)?
- Is the procedure in the process aligned to the interest groups (e.g., customers)?
- Is the procedure in the process clearly justified?
- Are there sufficient tools for executing the process (checklists, work instructions, etc.), in addition to the process description?
- Is the scope of the process clearly defined?
- Are the relationships of the process to other processes described or defined?

The above list of questions is only exemplary and not complete. Companies often use lists with up to 100 questions.

Reading extensive process documentation and comparing it with long checklists is very tedious. Experience shows that the intensity of the check decreases with an increasing number of pages. The first pages are still read in detail. Then the accuracy decreases continually. In order to compensate for the weaknesses of a visual assessment, a more formalized version of the review, the so-called "walk-through", was developed, whereby the walk-through refers predominantly to the process model.

Walk-throughs

Similar to code inspection in programming, in a walk-through a process is discussed step by step with selected process participants. In order to make the step-by-step process more engaging, a formal process description can be run through with the help of a practical example. A process participant goes through the business process description step by step using a concrete example. For each process step, an expert asks specific questions in order to question the effectiveness of the process description.

For example, the understanding of technical terms, the technical necessity as well as the completeness of the process description are questioned. In this way, the process description is evaluated. A walk-through is performed with about two to four process participants representing different user groups.

The 'authors' of the process description (e.g., process managers) should remain in the background so that criticism can be formulated openly. All points of criticism and suggestions are collected, documented, and then evaluated with the process participants. This evaluation leads to a revision of the process.

The step-by-step analysis of a process can be supported by appropriate tools. The tool used shows the process model on the screen and the current process step is highlighted in color.

Role-Plays

The next level for a tangible review of process models are role-plays. These are particularly useful when communication-oriented modeling languages are used. The actors are then already identified and the roles are subsequently assigned to suitable persons. A game leader triggers the process and provides the necessary input. These process instances are then executed by the individual role holders according to the process descriptions. These 'process flows' are observed by other affected parties and the anomalies identified are noted. After a number of process instances has been executed, the findings are evaluated and necessary adjustments identified.

The execution of role-plays can be supported by suitable IT tools. The role owners of a role-play do not receive their role descriptions on paper, they are guided through the process by software that, in particular, implements the flow logic. This software is generated directly and automatically from the process model. The prerequisite for such an approach is that the semantics of the process modeling language used are clearly defined. This prerequisite is, for example, only partly met by BPMN and not at all in the case of EPCs. However, S-BPM entirely fulfills this requirement due to its clearly defined formal semantics.

Figure 6.12 shows what an IT-supported validation can look like. The advantage of an IT-supported role-play is that the preparation time is very limited and the process experience is very close to the subsequent productive process execution (Figure 6.17).

6.2.2 Optimization

After checking the effectiveness (do the processes deliver the desired result at all?), it must be checked whether the result is achieved with the least possible use of resources. Optimization through manual testing, walk-throughs or role-plays is only possible to a very limited extent. The knowledge gained in this way only provides an approximation for determining the resource requirements for an assumed number of process runs. A systematic determination of the resource and time requirements is only possible through simulation. However, the prerequisite for a simulation is that the process model can be executed.

When business processes are simulated, the business cases handled by a process are generated randomly according to an assumed probability distribution. As a rule, this is the exponential distribution with a presumed or observed expected value. The individual work steps are assigned the corresponding resources with the required working time. The required working time usually follows a normal distribution with expected values and standard deviations determined from observations.

Within the framework of simulation instances, information is provided on the execution capability of processes, on process weak points, and on resource bottlenecks. On the basis of the simulated Process Performance Indicators, various alternatives can be evaluated and a realistic benchmarking carried out in advance of cost-intensive process changes within a company.

Modern tools and simulation methods enable the analysis and optimization of processes with regard to costs, lead times, capacity utilization or bottlenecks. In addition, the simulation of business processes forms a starting point for introducing activity-based accounting instead of the relatively inaccurate cost-plus pricing. The

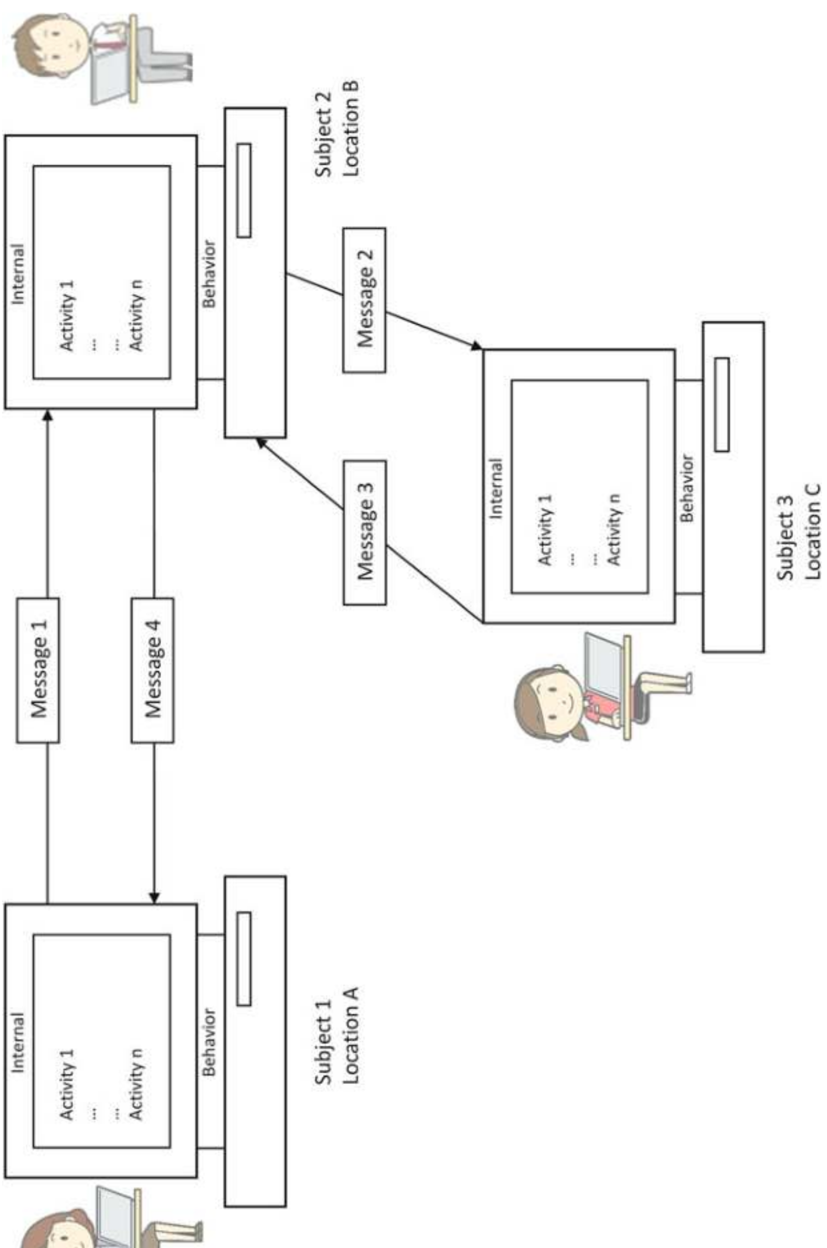


Fig. 6.17 Setting for an IT-supported role-play

profits and losses of the individual departments thus become transparent at an early stage.

As already mentioned, conducting a simulation study requires a precise description of the process under consideration. This means that a formal method must be used to define the process flow. In addition, the most precise knowledge possible of the probability distributions, their parameters and the examined performance indicators is required. In practice, simulations are not often used due to the high effort involved, although the findings can be compelling.

References

1. Seel, N. M. (2003). Model-centered learning and instruction. *Technology, Instruction, Cognition and Learning, 1*(1), 59–85.
2. Allee, V. (2003). *The future of knowledge. Increasing prosperity through value networks*. Amsterdam: Butterworth-Heinemann.
3. Senge, P. (2008). The necessary revolution: How individuals and organisations are working together to create a sustainable world. *Management Today, 24*(10), 54–57.
4. Fleischmann, A., & Stary, C. (2012). Whom to talk to? A stakeholder perspective on business process development. *Universal Access in the Information Society, 11*(2), 125–150.
5. Augl, M., & Stary, C. (2015). Communication-and value-based organizational development at the university clinic for radiotherapy-radiation oncology. In *S-BPM in the Wild* (pp. 35–53). Springer International Publishing.
6. Augl, M., & Stary, C. (2017). Stakeholders as mindful designers: Adjusting capabilities rather than needs in computer-supported daily workforce planning. In *Designing healthcare that works* (pp. 95–112). Elsevier.
7. Fleischmann, A., Schmidt, W., Stary, C., & Augl, M. (2013). Agile process management through subject orientation. *HMD Practice of Business Informatics, 50*(2), 64–76.
8. Giaglis, G. M. (2001). A taxonomy of business process modeling and information systems modeling techniques. *International Journal of Flexible Manufacturing Systems, 13*(2), 209–228.
9. Lin, F. R., Yang, M. C., & Pai, Y. H. (2002). A generic structure for business process modeling. *Business Process Management Journal, 8*(1), 19–41.
10. Pinggera, J., Soffer, P., Fahland, D., Weidlich, M., Zugal, S., Weber, B., et al. (2015). Styles in business process modeling: an exploration and a model. *Software & Systems Modeling, 14*(3), 1055–1080.
11. Recker, J., Rosemann, M., Indulska, M., & Green, P. (2009). Business process modeling-a comparative analysis. *Journal of the Association for Information Systems, 10*(4), 1.
12. Zur Muehlen, M., & Recker, J. (2013). How much language is enough? Theoretical and practical use of the business process modeling notation. In *Seminal contributions to information systems engineering* (pp. 429–443). Berlin: Springer.
13. Stary, C. (2014). Non-disruptive knowledge and business processing in knowledge life cycles–aligning value network analysis to process management. *Journal of Knowledge Management, 18*(4), 651–686.

7 Realization

With the specification of effective and efficient processes, the foundation for their implementation has been laid. Following these preparatory activities, we now deal with the implementation of process specifications in an execution environment and the handling of process instances in live operation. The implementation of a process specification as an executable process comprises the activity bundles organizational implementation, IT implementation and operation including monitoring. Figure 7.1 shows the classification of these activities in the process management model.

Based on reflections on the documentation of elaborated process specifications, the following sections present selected methods for the activity bundles organizational and IT implementation, as well as for operating and monitoring of processes.

The activities of the process management model provide the conceptual framework for the implementation of business processes in a working system. As already discussed in previous chapters, the activity bundles represent a certain standard classification criterion, but they can be performed in any order. Each of the phases contains a bundle of activities that are typical for accomplishing the respective tasks. Every business process is in a certain phase at any particular point in time - or in other words, in a certain state: either being modeled, put into effect, executed, or analyzed. The life cycle model thus defines a phase or state space for business processes.

7.1 Process Documentation

Since an organization’s or company's business processes define how products and services are developed, manufactured and delivered, it is useful to document these processes in a structured way. Documentation should be made centrally available to all employees. It must be available at the beginning of the implementation activities at the latest. Business process documentation is the result of activity bundles during preparation measures.

A. Fleischmann et al., *Contextual Process Digitalization*,
https://doi.org/10.1007/978-3-030-38300-8_7

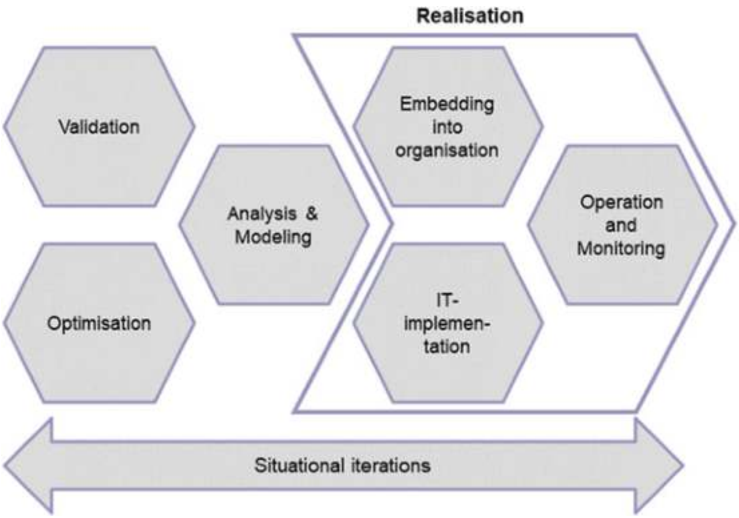


Figure 7.1: Classification of implementation activities in the process management model

Nowadays, digital documents serve as the standard for documentation. Ideally, the provision of these documents takes place by means of a generally available intranet, with which a basic access control can also be designed. The documents themselves can be, for example, PDF files that cannot be easily changed (possibly also digitally signed), or HTML files that can be viewed via a browser.

For smaller organizations and companies, it is usually sufficient to use existing office software for creating and maintaining process documentation. For more extensive collections, the use of specific software, so-called *Business Process Management Systems* (BPMS), appears to be worth considering. In any case, the chosen form needs to support its users and the management systems based on it in the best possible way. The documents should also contain a unique identifier, a version number and a date. In case a word processor is used, a style sheet is recommended in order to achieve a uniform appearance and uniformly defined structures. In addition, a list with all documents including their history is recommended to provide a general overview.

The process owner or process coordinator is responsible for ensuring that process documentation is up-to-date and complete. The process office specifies what the process documentation should contain and provides appropriate templates and explanations. In case a dedicated IT system (BPMS) is to be used, respective training of all employees - depending on their role - is strongly recommended. In the sense of change management, the employees should also be involved in the selection or development procedure in an adequate form.

Roughly speaking, the following content for a process documentation has proved to be useful, although not all items are relevant for every business process and additional items might be added where needed:

Aim of the process - a short descriptive text explaining the relationship of the process to overall business objectives (strategic contribution).

Trigger of the process - specifies which event starts a process instance and who can generate this event.

Input - a list with descriptions of the information, documents, and physical artifacts needed as input, including who provides it.

Output - a list with descriptions of the information, documents and physical artifacts generated by the process; possibly also quality criteria per customer.

Area of validity and organizational determining factors - if necessary, delimitation to other business processes, or organizational restriction (for example, only valid for the business area key accounts).

Definition of terms and abbreviations - abbreviations used in the document; Tip: If these are also consolidated in a company-wide directory, a glossary of the most important terms in a company, as well as uniform definitions in all process documentation, is obtained.

Overview description of the process model – a textual representation of the actors/roles and the process steps.

Process model - a (visual) representation created in an agreed modeling language; there should be uniform rules for the creation of process models to avoid uncontrolled growth (e.g., regarding colors and labelling of notation elements).

Technical determining factors - a list and description of technical tools required in the process; representation of IT support or dependencies; e.g., references to certain modules of an ERP system.

Feedback mechanisms – a description of how process participants can articulate problems during execution, e.g., incorrect or insufficiently modeled logic, or make suggestions for improvement. This should be a standard procedure implemented in Business Process Management.

Exceptions - possible exceptions from the process model, that is, activities that are not taken into account in the process model, since associated cases only occur rarely.

Interfaces to other processes - a representation of which other business processes require the output of a specific process, thus defining the customer of the process or his expectations.

Process Performance Indicators (PPIs) – a list, definition and explanation of the planned Process Performance Indicators.

Performance measurements - a representation of how and when the performance indicators are to be measured and calculated, including references to other associated available documents.

Reporting – a representation of how and at what point in time Process Performance Indicators are reported and to whom, including references to other associated available documents.

Escalations - a definition of procedures to be started as soon as the tolerance range of performance indicators is violated.

Audits and compliance – a reference to appropriate guidelines. If individual activities are included in a business process for reasons of compliance, they should be documented in order to ensure that they are not eliminated as part of an efficiency optimization. It is advisable to highlight such activities in color in a visual representation. Furthermore, it has proven helpful in practice to document (in tabular form) in which business processes compliance with specific standards and legal regulations is necessary. Similar considerations apply to external requirements of compliance, e.g., quality standards, which have an impact on internal compliance issues, such as risk analyses and internal audits.

7.2 Linking Elements of the Enterprise Architecture

7.2.1 Overview

As mentioned in chapters 1 and 2, business processes are part of the enterprise architecture. Enterprise architectures address the internal aspects and structures of a company. They are essentially models of the internal structure of a company and cover not only organizational but also technical aspects, in particular the deployed IT infrastructure. When implementing the process models, it is now necessary to establish the relationship between the process model and the available resources. Figure 7.2 shows the individual steps from a process model to the executable process instance.

In a process model, the actors, the actions, their sequences and the objects manipulated by the actions are described. Actions (activities) can be performed by humans, software systems, physical systems or a combination of these basic types of actors. We call them the task holders. For example, a software system can automatically perform the "tax rate calculation" action, while a person uses a software program to perform the "order entry" activity. The person enters the order data via a screen mask. The software checks the entered data for plausibility and saves it. However, activities can also be carried out purely manually, for example when a warehouse worker receives a picking order on paper, executes it, marks it as executed on the order form and returns it to the warehouse manager.

When creating a process model, it is often not yet known which types of actors execute which actions. Therefore, it can be useful to abstract from said model when starting to describe processes by introducing abstract actors. A modeling language should allow the use of such abstractions. This means that when defining the process logic, no assertion should have to be made about what type of actor is realized. In S-BPM, the subjects represent abstract actors. In BPMN, pools or swim lanes can be interpreted as abstract actors, while in EPCs roles can be used for this purpose.

In the description of the control logic of a process, the individual activities are also described independently of their implementation. For example, for the action "create a picking order" it is not specified whether a human actor fills in a paper form

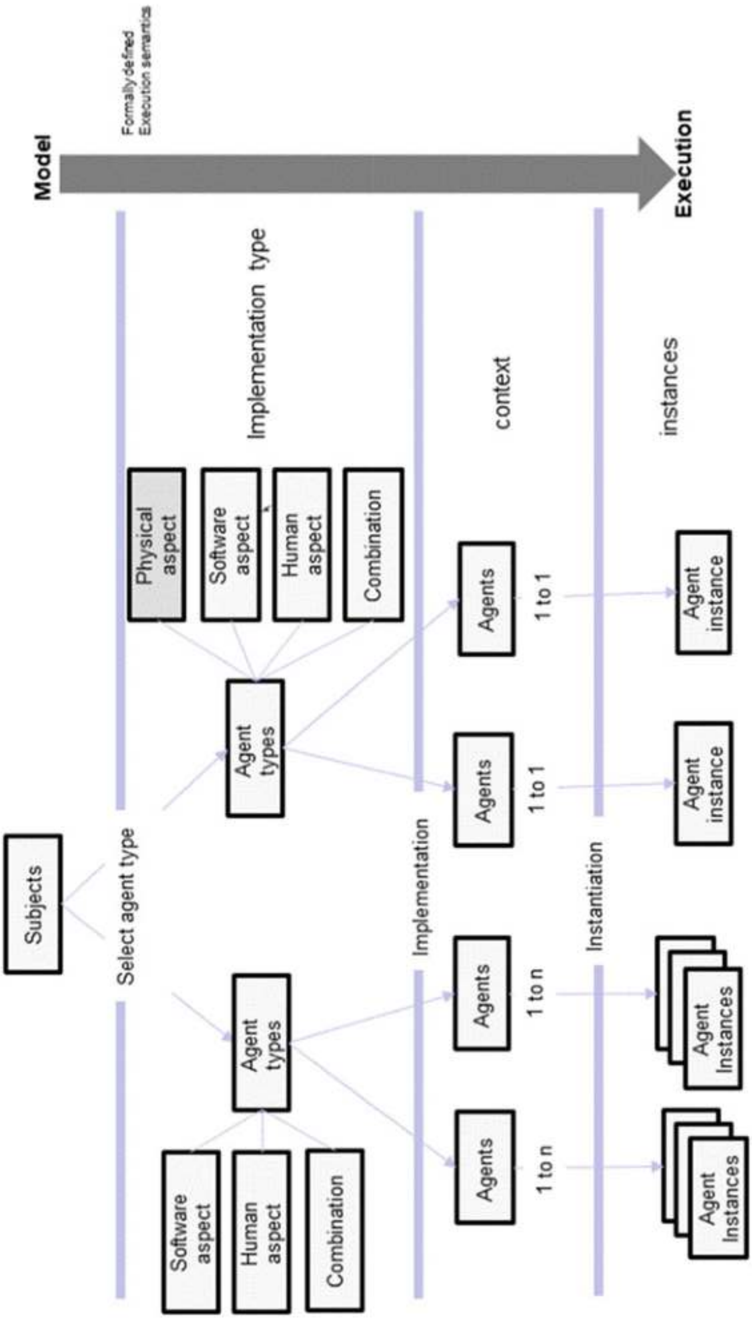


Figure 7.2: From process model to process execution

or a screen mask, or whether a software system generates this form automatically. Thus, with activities the means by which something happens is not described, but rather only what happens.

The means are of course related to the implementation type of the actor. As soon as it has been defined which types of actors are assigned to the individual actions, the manner of realization of an activity has also been defined. In addition, the logical or physical object on which an action is executed also needs to be determined. Logical objects are data structures whose data is manipulated by activities. Paper forms represent a mixture between logical and physical objects, while a workpiece on which the 'deburring' action takes place is a purely physical object. Therefore, there is a close relationship between the type of task holder, the actions and the associated objects actors manipulate or use when performing actions.

A process model can be used in different areas of an organization. The process logic is applied unchanged in the respective areas. However, it may be necessary to implement the individual actors and actions differently. Thus, in one environment certain actions could be performed by humans and in another the same actions could be performed by software systems. In the following, we refer to such different environments of use for a process model as context. Hence, for a process model, varying contexts can exist, in which there are different realization types for actors and actions.

In BPMN, the modeler can define for each task separately whether it is a so-called human task, service task or user task. User tasks are performed by humans together with software systems, such as filling out a picking order on the screen. This means that the description of the implementation type in BPMN is part of the process logic. Since in BPMN pools and swim lanes can be interpreted as actors, care must be taken that no contradictions arise with the implementation type of tasks within, for example, a pool. For instance, in case the designer specifies that a pool is only executed by humans, this pool cannot contain any service or user tasks. Since the definition of the implementation type is part of the process logic in BPMN, it may be necessary to create a separate process model for each context.

In S-BPM, actors are not assigned to individual activities, but rather the actor type is assigned to an entire subject. In S-BPM, this assignment is not part of the process logic, but is done instead for each process in a separate two-column table. The left column contains the subject name and the right column the implementation type. If there are several contexts for a process model, a separate assignment table is created for each of them.

The assignment of the implementation type forms the transition between the process logic and its implementation. Subsequently, it has to be defined which persons, software systems and physical systems represent the actors and how the individual actions are concretely realized. These aspects are described in detail in the following subsections.

7.2.2 People and Organizations

For each context in which people are involved it is necessary to determine the respective action (activity) holders, and thus the concrete persons or organizational units that carry out the actions.

In companies and administrations there are people with different educations, qualifications, preferences, and interests. There are merchants, developers, craftsmen, etc. who will take care of the arising tasks. Organizations can therefore also be described as *structured resource pools*. Depending on the type and scope of the tasks, an organization forms units in which the respective specialists are combined. There are purchasing departments with procurement experts, or development departments that are made up of several development engineers and other specialists. The relationship between the persons and organizations in the organizational structure and the abstract actors in the processes can be established statically or dynamically.

Static Assignment

In the simplest case, the two-column table already mentioned provides information on which actor defined in the process (column 1) is assigned to which person (column 2). The second column can also contain an organization or organizational unit. Then, all its members are assigned to the abstract actor. Such a specification ensures that in case of illness or vacation any person from the organization can take over the arising tasks of the process. In addition, the workload can also be distributed dynamically if actions from several process flows (process instances) have to be handled.

With BPMN, pools, swim lanes and individual actions can be assigned separately to an actor. It is important to ensure that there are no inconsistencies when using all assignment options. If a software system executes a swim lane as an actor and there is a human task in this swim lane, it is not clear what this means. Actors were therefore introduced in Bonitasoft's BPMN-based tool (see [1]). They are placeholders for task holders, to whom, similar to the subjects in subject-orientation, concrete actors are assigned.

In S-BPM, a complete subject is embedded into the organization, i.e., the assignment then applies to all actions of this subject.

Dynamic Assignment

In many processes, the assignment of the actors of a process to persons and organizational units cannot be determined statically. For a business trip application, the person handling the request can depend on whether the trip is domestic or international. While the process logic may be the same in both cases, the executor may differ, for example because an employee has special expertise in international travel with visa issues.

In such cases, it makes sense to first determine and assign the persons or organizational units involved during execution of the process logic, for example, depending on data values in business objects (in this case, the travel request).

For a flexible dynamic assignment of physical actors, tables are usually no longer sufficient, but instead programming language constructs are necessary. [2] shows how such a language can be used to embed subject-oriented Process Models into organizations. Since in BPMN the assignment of actors is possible via pools, lanes and tasks, the description of the assignment is very complex here. Various BPMN-based tools such as Bonita [1] or Activiti [3, 4] integrate processes into the organization by programming this in Java or XML.

7.2.3 Physical Infrastructure

Particularly in manufacturing processes, physical systems are involved as actors. In this way, blanks can be delivered to a machine via a transport system, the machine processes the blanks, and the processed parts are then transported to the next processing step. If such a machine is regarded as a subject or pool, the delivered parts are modeled as messages to be received and the dispatched processed parts as messages to be sent. The modeler represents the processing step as a single task in a subject or pool.

7.2.4 IT Infrastructure

Digitalization in an economic system means implementing processes with the most comprehensive support possible through software systems. In corresponding scenarios, computer systems or machines carry out the activities for the most part; human intervention is reduced as far as possible and sensible. Essential aspects thereby are:

- The control of the process flow:
  The sequence of actions described in the process (control flow) are automatically controlled by computer programs.
- The execution of actions
  Actions on data objects can often be performed fully automatically by appropriate computer programs.

Software systems are the means to merge the different types of actors. For example, during the control of the process flow, a process engine integrates human and machine operators at runtime in the processing of instances, according to the individual situation.

The following explanations deal with the control of the process flow and the manipulation of the associated business objects by Information Technology, although without going into detail about the integration of humans or physical systems as task holders. This will be covered in section 7.2.5.

Control Flow

The process logic corresponds to the control flow logic of a computer program. The exclusive automated execution of the process logic by a computer program is only possible for highly structured processes. All conceivable process possibilities must be covered. No human intervention is planned. To avoid that the computer program's execution logic differs from the described process logic, it is useful if the computer program can be automatically derived from the description of the process logic. The following prerequisites must be met for such cases:

- The syntax and semantics of the language in which the process logic is described must be precisely defined.
- The description of the process flow must be available in electronic form.
- An implementation program must be available that reads the electronic form of the description of the process logic and generates a computer program in a suitable programming language, based on the precise semantics of the process description language.

The importance of distributed systems, which are already widely used today, continues to grow with the development of cloud and edge computing. This can mean that parts of a fully automated process are executed on different computer nodes of a distributed system. Since these can be based on different technologies, appropriate variants may have to be available for the automatic conversion of a process description into a computer program.

Thus, individual subjects of an S-BPM description of a process can run on different computer nodes of a distributed system. This can mean that the program code might need to be generated separately for each subject. This can, however, be avoided if the generated target code is available on preferably all the node types of the distributed system, and the target system has a framework through which the program parts running on different nodes can work together. The software components usually synchronize their collaboration by exchanging messages. Such an enabling programming language is, e.g., Java with the AKKA framework. It allows messages to be exchanged across computer boundaries. [5] has investigated this possibility on the basis of S-BPM models.

In BPMN, individual pools can be executed theoretically on different computer nodes. This requires that communication between the pools is possible across computer boundaries. A search for possibilities of a corresponding code generation tool for BPMN models in spring 2018 brought no results.

Activities and Data

In fully digitalized processes, computer systems also perform the activities contained in the process flow. This can be done using functions or services that are already available in existing application programs and can be inserted or called at the appropriate places in the program that implements the process control flow.

For the integration of newly developed software for activities, technologies such as web services, REST interfaces, or simple APIs are often used. Database access for direct manipulation of data can be integrated analogously depending on the interfaces of the database systems used.

When using S-BPM, data belongs to a subject. Multiple subjects are not allowed to access shared data. If several subjects want to use the same data, e.g., stored in a database, it has to be 'packed' into a subject. This subject receives the corresponding requests from other subjects, performs the desired action, and sends the result back to the requesting subject.

In software technology, activities together with their data are called classes. Several such classes can be combined to form services. In IT, this is then called a Service-oriented Architecture. The services are invoked according to the control flow of the process. Tasks in BPMN can be realized through services. Similarly, functions can be implemented within a subject in this way.

Microservices [6] are a current concept for structuring software systems. Microservices describe an architecture in which individual small functional building blocks are independently programmed, tested, approved and made available on different technical platforms on demand. Through their interaction by means of interfaces which are independent of the programming language, complex application software can be realized. In doing so, microservices usually communicate by exchanging asynchronous messages. This also corresponds to the concept of Reactive Programming [7], whose main characteristic is the coupling of program blocks by asynchronous messages.

In BPMN, pools can be implemented as microservices if all tasks in a pool are executed by the same actor.

In S-BPM, subjects are always assigned to a task holder and communicate via the asynchronous exchange of messages. Therefore, a process described in a subject-oriented format corresponds, in view of IT architecture, to a microservice-oriented structure that meets the requirements of Reactive Programming.

7.2.5 Combinations of Task Holders

Tasks are often not performed by a single type of task holder. With the increase of digitalization, the number of tasks performed exclusively by humans is decreasing continuously. Today, a warehouse worker often receives his picking orders via a tablet connected to an IT system. After he has assembled the goods, he also confirms task completion via the tablet. This action closes the commissioning task. The IT system updates the data and automatically triggers the subsequent task, such as preparing the shipping documents. The picking task is therefore carried out by humans and IT as types of task holders, a combination that is very common in business processes. The IT controls at least the process logic and manages the data while people enter data or operate machines.

Machines controlled by small computers with corresponding software are increasingly replacing machines as purely mechanical actors. These embedded systems control the mechanics and handle communication with other machines or higher-level business applications. Communication with other intelligent machines is referred to as horizontal communication, and with business applications as vertical communication. The latter links production systems with business processes and is an integral part of the Industry 4.0 initiative.

The following sections describe in more detail how the various task holders are integrated when implementing a process logic.

Combination of Humans with IT

The combination of people and IT in the execution of business processes is the core of their digitalization. The IT at least takes over the control of the process logic, i.e., it arranges the handling of the intended tasks by the respective task holders according to the process logic. In addition, tasks can be carried out directly by IT as task holders without the need to involve people. If a task can only be completed with human support, the software responsible for the control flow prompts the responsible human task holder via an appropriate user interface. This may concern not only the input of data, but also the confirmation that the user has executed a manual action.

IT systems as control software and as actors for activities usually also store the data generated during process execution, both the content of business objects, as well as metadata, such as time stamps for the start and end of instances and execution steps.

A comprehensive platform for implementing business processes executed by people and IT solutions is called a Workflow Management System (WFMS). The Workflow Management Coalition (WfMC) has developed a reference model for this purpose. Figure 7.3 shows the components and interfaces of the reference model [8].

The interfaces between the individual components are defined as follows:

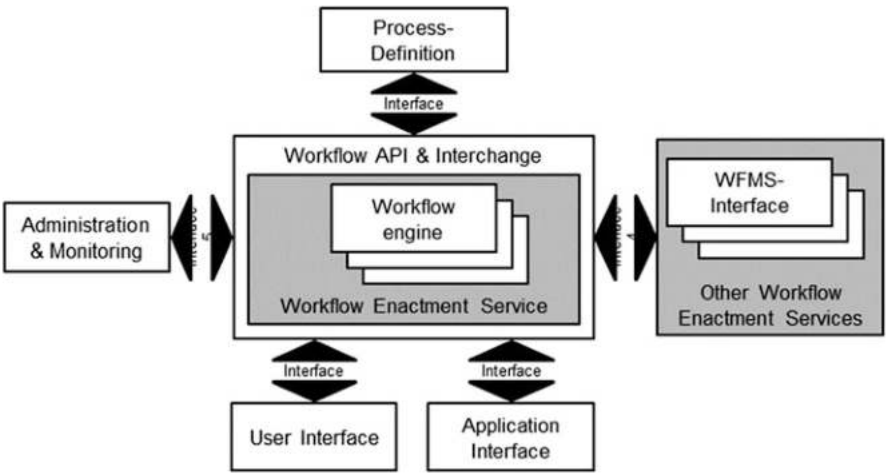


Figure 7.3: Reference model for Workflow Management Systems (WFMS) [8]

- Process definition (Interface 1)
  Interface between process definition, modeling tools, and the workflow engine (execution environment).
- User interface (Interface 2)
  APIs for clients to request services from the workflow engine so that process progress and actions can be controlled.
- Application interface (Interface 3)
  APIs that allow the workflow engine to call and use applications.
- Workflow Management System interface (Interface 4)
  Standard interface for exchanging data with other workflow systems.
- Administration and monitoring (Interface 5)
  Interface for tools for process control and monitoring.

The essential components in the reference system of the WfMC are the workflow enactment service and the workflow engine. The actions and their sequence are defined in the process definition and controlled by the workflow engine.

In a company usually several instances of a business process run simultaneously. A process instance is created when the execution of a business process is triggered by an event defined for this purpose. Process instances follow uniform process descriptions but are executed independently of each other.

For example, two different customers can place an order. Each of these individual customer orders creates an independent instance of the business process "*order processing*".

The workflow enactment system uses workflow engines to manage the individual instances. A workflow engine provides the execution environment for a workflow instance.

Its main tasks are:

- Integration of the instance into the organizational environment
- Interpretation of the process definition
- Control of the process instances
- Navigation between sequential or parallel process activities
- Interpretation of process data
- Identification of user interfaces
- Linking of the system to other programs/application systems
- Linking of the system to other workflow systems
- Superordinate control function

A workflow enactment service is a service that starts, manages and executes one or more workflow engines. Workflow systems use application interfaces to access the functions of application systems that perform tasks defined in a process.

A worklist handler is a component that involves users in a process. It can be part of a Workflow Management System or defined and programmed by a workflow expert. Through the worklist handler the involved actors know which tasks they have to perform in which process instance.

Workflow functions can be embedded in other common applications, such as an e-mail program, so that users have a uniform user interface when handling process instances. For such an integration, there must be a communication mechanism between the workflow enactment service and the other applications.

The reference architecture also provides an interface for administration and monitoring. The software connected there serves to monitor the simultaneous execution of different instances of several business processes. On the one hand, this monitoring refers to the functional level of the processing of business cases, for example with the recording and evaluation of response and processing times (see section 7.3.3). On the other hand, the IT systems used must be monitored at the technical level, for example, with regard to load and malfunctioning behavior.

Most workflow systems currently available on the market only support orchestration, i.e., a single control flow of tasks. For BPMN, this means a workflow system can only execute the functions in a single pool. If several pools are used in a BPMN-based process description, a separate workflow engine is required for each pool. These workflow engines must then as a rule be connected by programming via Interface 4. Especially if a process should be executed on a distributed infrastructure, such structures can become complex and cost-intensive, as many software vendors charge license fees for each installed instance of the workflow engine. This increases not only technical complexity but also the costs.

Subject-oriented process descriptions are process choreographies, i.e., the subjects are independently executed in parallel, without central control for all subjects. To coordinate their individual activities, the subjects exchange messages asynchronously. An execution environment for subject-oriented defined processes is therefore, strictly speaking, a so-called multi-workflow system. Each subject corresponds to an orchestration performed by its own workflow system. The asynchronous exchange of messages between several workflow systems is realized in Business Subject-oriented Process Management using the input pool concept. A number of multi-workflow systems have been developed for S-BPM (see [9]).

Combination of Physical Devices and IT: Cyber Physical Systems

In a process, actors can also be machines that perform tasks in a purely mechanically or electrically controlled way, and fully automated without human intervention.

Machine control is carried out mechanically, electromechanically or, in the case of the embedded systems already mentioned, increasingly by computer and software systems. Sensors monitor the physical condition of the machine and ambient conditions, and actuators such as servomotors, switches, controllers, etc. intervene with the operation of the machine. The control software largely determines what happens on and at intelligent machines and defines their vertical and horizontal communication with other systems.

Machines communicate with each other not only by exchanging data, but also by forwarding and receiving physical artifacts. The message "workpiece to be deburred" can be realized by automatically transporting a workpiece from a metalcutting production machine to a machine that deburrs edges. At the same time, the deburring machine can receive additional messages that contain a more detailed description of the workpiece and specify the type of deburring.

Combinations of physical and Information Technology components are referred to as Cyber Physical Systems (CPS). Since they bring together production and business processes, they are of particular importance for Industry 4.0 applications.

In BPMN, the modeler can model a task executed by a combination of humans and IT as a service task and describe an intelligent machine as a separate process in a pool. If there is a corresponding number of machines working together in a complex manufacturing situation, the use of numerous pools in BPMN can lead to representation problems. The reason lies in the horizontal arrangement of the pools: In case of a larger number, the message edges inevitably have to cross pools, which can be confusing. In S-BPM, on the other hand, a machine is always modeled as a subject, since the decision as to which type of task holder will execute the subject is first made in the implementation stage.

If a physical task holder is used in a process, there cannot be an arbitrary number of parallel instances of this process. A physical unit cannot be split logically, as is possible with IT or human actors.

Combination of Humans, Physical Devices, and IT

There are also scenarios in which people carry out certain tasks when intelligent machines are used. For example, a machine receives the workpiece via a transport system together with the associated information as a message, e.g., contained in a RFID device. Together the machine and an operator carry out the intended tasks in accordance with the information contained in the RFID device. The operator confirms the completion of his work via a user interface. The workpiece is then dispatched and the updated accompanying information is transmitted to the other parties involved.

In such situations, the aim is to reduce the share of human labor and replace it with more sophisticated mechanics or improved embedded systems. This does not necessarily mean that changes to the process logic are required.

7.3 Execution and Monitoring

7.3.1 Putting the Process Into Operation

After the definition of task holders and the implementation of the activities, the process has to be put into operation.

To this end, the task holders must be prepared by setting up the physical and IT infrastructure and by training the human task holders. IT-based task holders are loaded with the respective programs and the mechanics of machines are configured accordingly.

During the go-live phase, it is important to establish the link with other processes. In a company, processes are integrated into a coherent network of processes. There should be no isolated processes. If, for example, a new process for shipping preparation is introduced, it needs to be linked to the order acceptance process. With communication-oriented process descriptions, this is done simply by exchanging messages. Another possibility is shared data, i.e., processes write data required by other processes into a shared database.

After the preparatory work, the process should be tested as an overall construct. An advantage would be a test environment that represents a realistic image of the operational environment. However, for cost reasons this is often not possible.

Irrespective of that, it is advisable to introduce the process step by step. It is an advantage if a process is described and implemented as a loosely coupled system of communicating task holders. The individual strands of action for the actors can be put into operation step by step. Since they are only loosely coupled to each other through asynchronous message exchange, other task holders can be initially simulated easily. With BPMN, pool by pool is thus put into operation. However, if the pools are very complex and include several swim lanes, the go-live phase is also complex. When using S-BPM, a system can be built by gradually adding the individual subjects and their behavior.

If modelers do not use several pools in BPMN, i.e., they define and implement a process exclusively in a control flow-oriented way, it can only be put into operation as a whole.

In the case of choreographies, i.e., when using subjects or several pools, the process can be executed, even though, for example, one pool contains software which is still faulty.

The messages sent to this pool can alternatively be received and worked on by people, and the result sent back by them as a message. The other pools recognize the described logic and flow (i.e., the observable behavior), although it is not yet implemented in its final form.

7.3.2 Process Instances

The actual execution of a process is referred to as a process instance. A process instance is created when the start event occurs. This can be a call from a customer who wants to order a product. In this case, a telephone salesperson explicitly creates an instance by starting the digital ordering process and entering the necessary data. When ordering in an online shop, the customer enters the order data himself and creates an instance as soon as he confirms the purchase. In both cases, an instance then runs through the processing steps and positions defined in the process logic.

Since a company normally has several purchase orders at any particular point in time, several independent process instances, which are in different processing stages, exist at the same time. Employees of a company are usually involved in several process instances and carry out the tasks assigned to them in these instances. They allocate, so to speak, a time slot of their working time to each process instance. This happens analogously with IT systems. The capacity of both the human resources and the IT systems can thus be divided into time slots in order to process several instances of the same process, or different processes.

This cannot be readily done with physical or cyber-physical systems. Such a system can only be involved in one instance at a time. A machine cannot be instantiated several times, it is only available once. Only after a machine has completely finished a task or task sequence in a process instance, can it work for another process instance. A physical system is therefore assigned to exactly one process instance at a particular point in time. This fact is important when processes that can be easily instantiated because they do not contain physical task holders are linked to processes that contain physical components.

A manufacturing system almost entirely consists of physical or cyber-physical task holders. A process running in such a system is instantiated just once and exists as an instance until the whole production environment is switched off. A physical task holder cannot work in a time slot for a process instance 1, then do something for a process 2, and then continue working on process instance 1. The actions of a physical task holder for different process instances would not be independent of each other. If, for example, a valve for process instance 1 is to be half opened and then fully opened for process instance 2, the valve is of course also fully opened for process instance 1, which should not happen.

If a machine is assigned to a task in BPMN and the assigned pool can be instantiated several times, the machine must always be assigned to a single instance at runtime. The machine must always know which instance it is working for, in order to retrieve the correct data from the memory common to all instances, and then take the work piece to be processed out of the work piece container.

When using S-BPM, the machine is assigned to the corresponding subject. The latter receives the work piece and the accompanying data as a message. The data also contains an identification of the associated higher-level process instance, usually the order number. With this, the machine knows for which process instance it is now working. The message by which a machine declares its work finished is transmitted to a subject instance of the triggering process instance.

7.3.3 Monitoring

In day-to-day business, process participants execute business processes according to the modeled design in the environment created during organizational and IT implementation, and consisting of personnel and technical resources.

Each business transaction runs in this execution environment as an instance. Transaction systems such as Enterprise Resource Planning (ERP) applications or workflow engines record the behavior of the instances in the form of entries in a log file (event log, audit log). A log data record contains, among other things, a unique instance identifier, a partial step identifier and time stamps for the start and end. This results in raw data for the calculation of Process Performance Indicators (PPIs).

Reliable management information based on such key indicators is the prerequisite for continuous adaptation of process design with respect to increasing the degree of target achievement. The periodic ex post facto evaluation of a large number of instances over longer periods of time such as weeks, months, quarters, etc. primarily serves to identify structural improvement potential, for example with regard to scheduled personnel deployment, process logic, or the degree of IT coverage. This traditional monitoring and reporting follows the concept of Business Intelligence with the principle 'store and analyze' and the methods of data and process mining. The resulting changes are primarily of a medium- and long-term nature.

In order to meet the increasing real-time requirements, traditional process monitoring is supplemented by Business Activity Monitoring (BAM), which evaluates event-driven data almost in real time, reports results promptly, and thus enables short-term, instance-related measures [10]. An example is the prioritized processing of an ordering instance of an “A” customer once the system recognizes and reports that it lags behind the usual processing progress at a measuring point, and therefore the promised delivery date cannot be met (predictive analysis). BAM uses the concept of Complex Event Processing (CEP) with the principle 'stream and analyze' and stream mining methods. This means that the system constantly searches for patterns of complex events in the stream of recorded individual events (e.g., set time stamps, passed measuring points), which only become relevant for certain purposes by linking the individual events.

A typical example is the recording of two transactions with the same credit card in Hamburg and New York. These simple single events (low level events) are registered in the event stream as normal events. The CEP system only combines them into a complex event if both transactions take place within a short period of time, in this example about 3 hours. In this case, an event pattern of geographical distance and time frame would lead to classification as a complex event of "assumed credit card fraud".

Business activity monitoring is intended to monitor a large number of data sources permanently and simultaneously. Event data generators include applications that execute process instances (e.g., ERP, CRM, workflow engines), and which provide other information from inside or outside the enterprise, such as surrounding conditions, weather and traffic data. Increasingly, this also includes (sensor) data produced by smart phones and devices that are part of the Internet of Things. BAM analyzes and aggregates the flood of data using defined rules and transmits the results to entitled and interested recipients.

Figure 7.4 juxtaposes traditional and Business Activity Monitoring. The activities listed in the left column serve as attributes for comparison [11].

The timing and type of exploitation of the recorded/registered data depend on the design of the monitoring and reporting according to the user requirements. Utilizing the pull principle, the user can retrieve the desired evaluation at any time. According to the push principle the system generates evaluations on a time-controlled basis, e.g., daily, weekly, monthly, quarterly at specified times, and informs the predefined recipient group accordingly. If limits or tolerance thresholds of Process Performance Indicators of individual instances are exceeded at runtime, alarm messages can also be transmitted by push to those responsible, or other processes can be started, e.g., an extensive escalation procedure with corrective measures for the case in question.

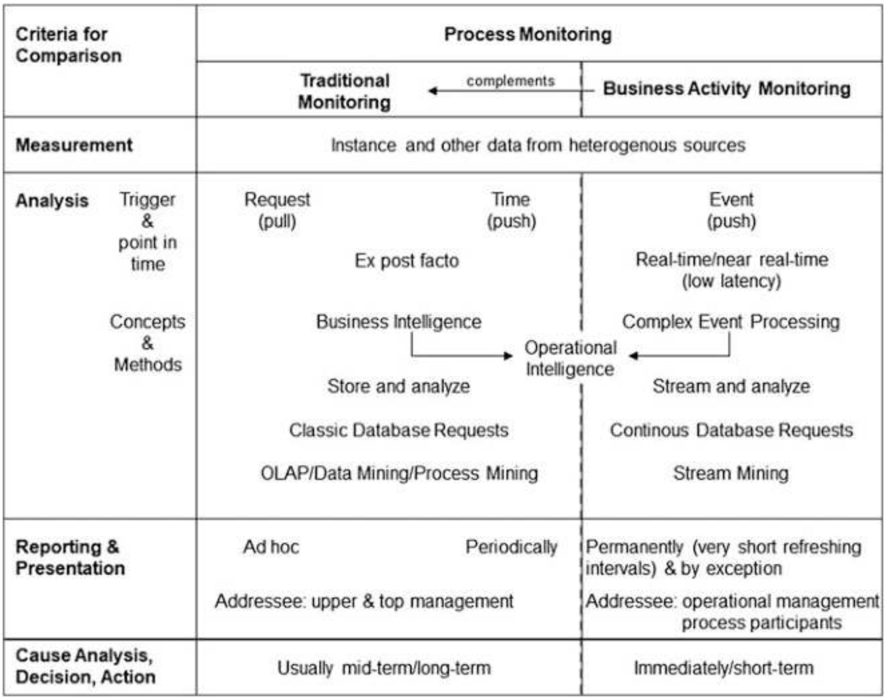
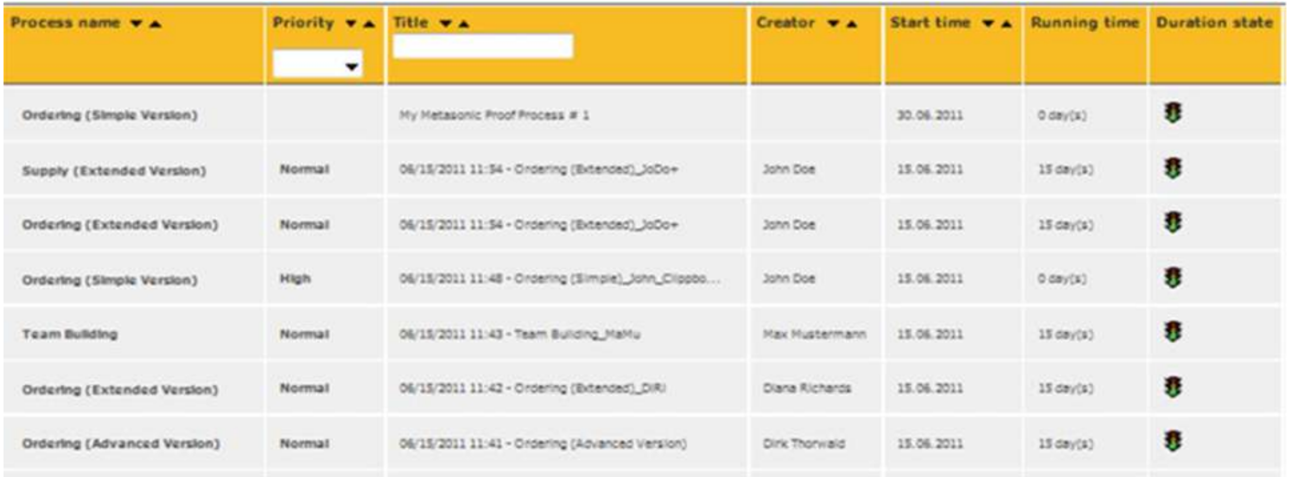


Figure 7.4: Properties of Traditional and Business Activity Monitoring

Management cockpits containing dashboards dominate for the presentation of evaluation results. These usually include representations in the form of tachometers, traffic lights and bar or pie charts. For space-saving displays with high information density, word graphics such as spark lines or bullet graphs are also used.

Examples of evaluations and their presentation are:

- Lead time of running and completed instances. The status is expressed in traffic light colors, depending on the deviation from defined target values or tolerance ranges (see Fig. 7.5).
- Lead time of a running instance (absolute value and comparison with average), chronological sequence and duration of individual steps for each process participant (cf., Fig. 7.6).
- Sequence of process steps of a running instance with time stamps for the completion of a step, i.e., for the transition from one state to the next (see Fig. 7.7).
- Number of instances of a process per time unit, minimum, average and maximum processing time per process participant and step (see Fig. 7.8).



Process name ▼▲	Priority ▼▲	Title ▼▲	Creator ▼▲	Start time ▼▲	Running time	Duration state
Ordering (Simple Version)		My Metasonic Proof Process # 1		30.06.2011	0 day(s)	
Supply (Extended Version)	Normal	06/15/2011 11:54 - Ordering (Extended)_JoDo+	John Doe	15.06.2011	15 day(s)	
Ordering (Extended Version)	Normal	06/15/2011 11:54 - Ordering (Extended)_JoDo+	John Doe	15.06.2011	15 day(s)	
Ordering (Simple Version)	High	06/15/2011 11:48 - Ordering (Simple)_John_Clippbo...	John Doe	15.06.2011	0 day(s)	
Team Building	Normal	06/15/2011 11:43 - Team Building_MaMu	Max Mustermann	15.06.2011	15 day(s)	
Ordering (Extended Version)	Normal	06/15/2011 11:42 - Ordering (Extended)_DiRi	Diana Richards	15.06.2011	15 day(s)	
Ordering (Advanced Version)	Normal	06/15/2011 11:41 - Ordering (Advanced Version)	Dirk Thorwald	15.06.2011	15 day(s)	

Figure 7.5: Instance report (overview)

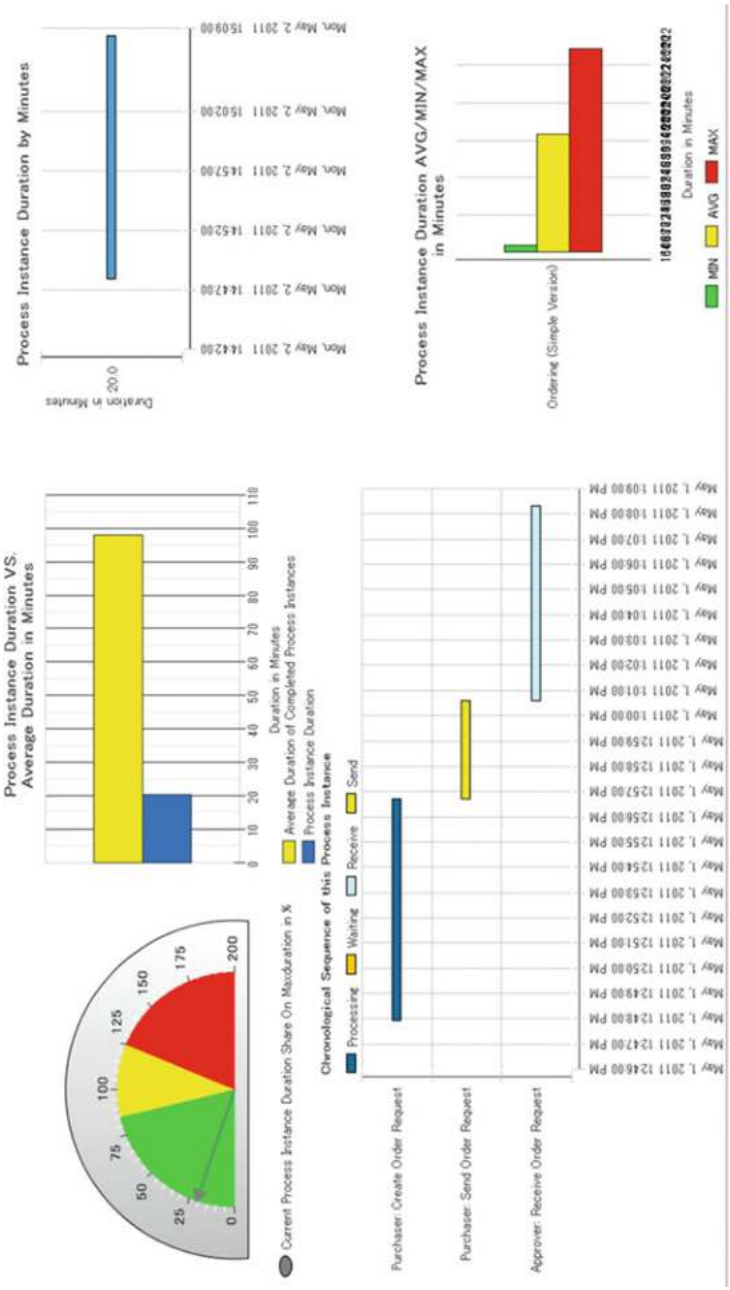


Figure 7.6: Instance report (single instance)

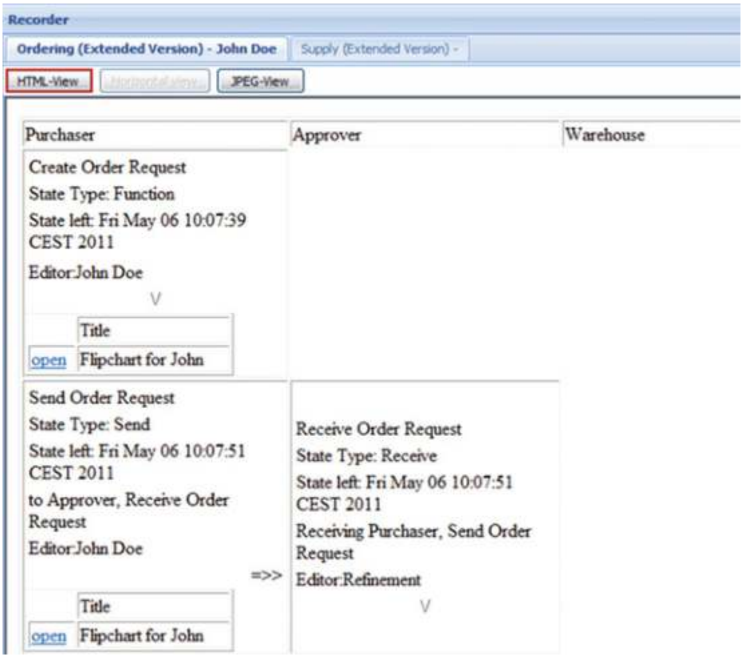


Figure 7.7: Instance report (current processing status)

7.3.4 Process Mining

Process mining is a special form of evaluating process data. Log files are thereby used to extract process-related information from the transaction data of central IT systems (e.g., ERP, CRM and SCM), and thus visually reconstruct and analyze the actual process flow. This allows insights to be gained into the actual behavior of executed process instances [12]. Hence, process mining concepts and tools deliver valuable information for analysis and continuous improvement of processes (see section 1.3.5). The most common and frequently analyzed business processes are purchasing, sales, accounts payable and accounts receivable, as well as ticketing systems (e.g., in IT service management).

An event log contains sequentially recorded events (trace) with attributes such as case ID, activity, timestamp, resource, and business objects (data elements). Depending on the process, additional information can be added, such as the name of a customer or vendor, order quantity and value, delivery quantity, etc. Figure 7.9 shows an extract of an example of an event log.

Combining such protocol data for process execution with process models allows identifying three essential types of process mining approaches [12]:

- *Discovery*: Algorithms reconstruct the actual process flow and its manifold variants from the log data (without additional information). This allows the

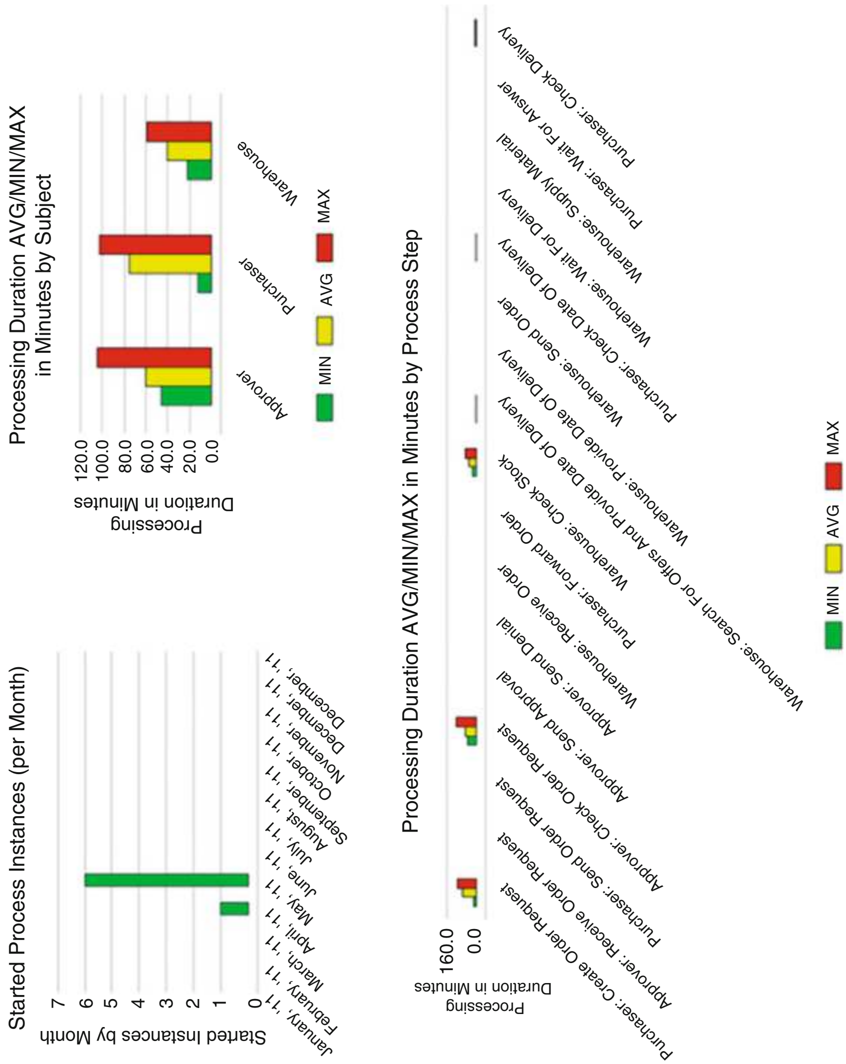


Figure 7.8: Process report

CaseID	Activity	Timestamp
10001	Create purchase order	01-01-2009, 8:35 am
10001	Print and send purchase order	03-01-2009, 12:13 am
10001	Goods receipt	07-01-2009, 07:01 am
10001	Scan invoice	09-01-2009, 2:00 pm
10001	Book invoice	10-01-2009, 10:30 am
10002	Create purchase requisition	02-02-2009, 1:17 pm
10002	Create purchase order	04-02-2009, 9:15 am
10002	Print and send purchase order	07-02-2009, 4:41 pm
10002	Goods receipt	27-02-2009, 6:53 am
10002	Scan invoice	28-02-2009, 1:00 pm
10002	Book invoice	13-03-2009, 11:59 am
10003	Scan invoice	13-04-2009, 10:00 am
10003	Create purchase order	17-04-2009, 3:47 pm
10003	Print and send purchase order	17-04-2009, 5:30 pm
10003	Goods receipt	27-04-2009, 4:23 pm
10003	Book invoice	30-04-2009, 8:50 am

Figure 7.9: Extract from event log (Image taken with permission of Celonis SE (www.celonis.com))

formal description of processes that are already in day-to-day operation ('lived') but not yet documented.

- *Conformance*: Algorithms compare an existing, valid process model with the actual process as it is derived from the log data. Any discrepancies that are detected can provide clues for optimization and uncover abuse or violations of compliance rules.
- *Enhancement*: In this case, existing models are adapted to the observed reality of process execution (repair) or further aspects, such as duration, are added to the model (extension).

In particular, the conformance check should be carried out as part of Business Activity Monitoring during the runtime of instances, in order to promptly detect violations and mitigate their consequences, e.g., stopping a bank transfer in the event of irregularities in the release of payments.

The following example is taken from the process mining tool provided by Celonis SE (www.celonis.com). It shows selected mining results for an ordering process (purchase to pay) that is handled by an ERP system. The analysis of the associated event log for a given period has produced the following information (see Figure 7.10):

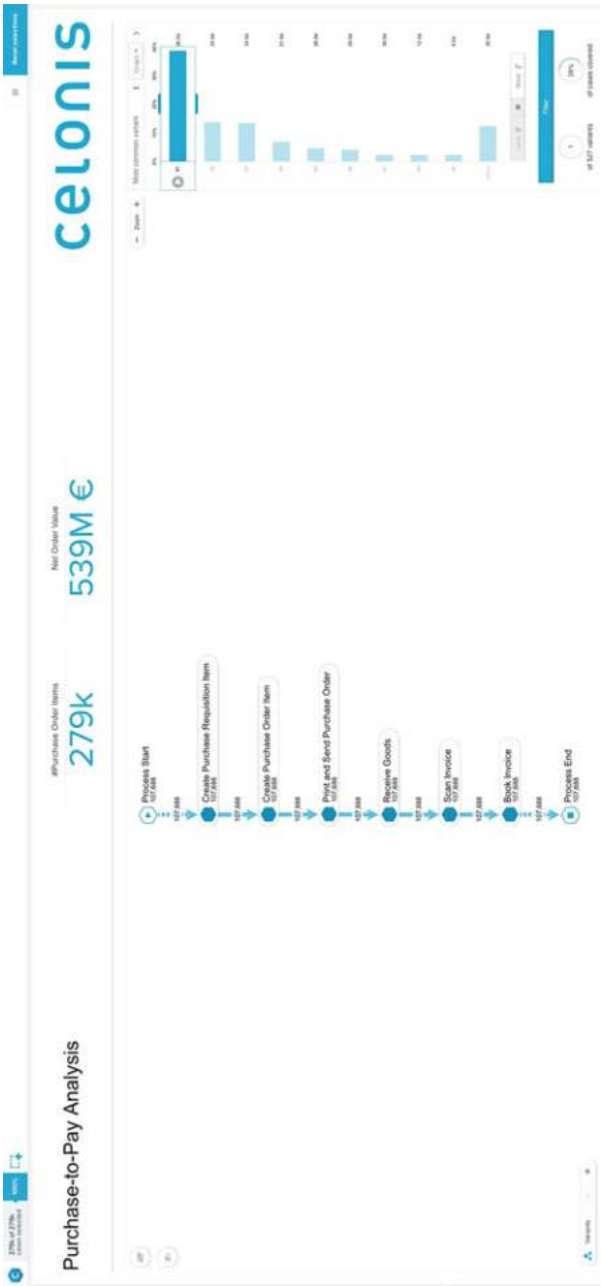


Figure 7.10: Overview of process variants (right) and happy path (left) (Screenshot of Celonis Viewer/Variant Explorer with permission of Celonis SE.)

- 279,020 instances of the process with a net order value of €539,180,072
- 107,688 instances followed the normal process flow (happy path), starting with the creation of a purchase requisition and ending with the posting of the invoice (see CaseID 10002 in Fig. 7.9). The average lead time is 27 days. Process variant 1 covers 39% of all instances; in addition there are 527 other variants.

If the scope is expanded to include the seven most frequently occurring process variants, 83% of all instances are covered. The process graph now also shows the seven paths with the corresponding frequencies (see Fig. 7.11). Alternatively, the average duration of paths and state transitions can be displayed. With regard to the process flow, it can be seen that a considerable number of instances (18,938) begins with the scan of the invoice and only then is an order created in the system (see CaseID 10003 in Fig. 7.9). This is an indication of so-called maverick buying, i.e., departments procure something without prior involvement of the purchasing department. In a next step, this usually undesirable process behavior can be specifically investigated in order to eliminate its root causes.

With the display of complementing information, processes can be examined in greater depth. Figure 7.12, for example, provides information on the number of order instances distributed over months with the respective values as well as their distribution among suppliers (Fig. 7.12).

The presented analyses cover only a small part of the extensive capabilities of the Celonis environment. In addition to the default evaluations provided by the system, users can define their own evaluation scheme on the basis of their individual data models by using a large number of existing analysis components, such as various diagram types, Process Performance Indicators, and OLAP tables. Furthermore, the software allows executing individualized process evaluations with the help of the Process Query Language (PQL).

Additional functions allow, for example, the comparison of the actual process flow with the one intended in the model (conformance check), or the parallel visualization and thus the comparison of different behaviors of the same process, e.g., in two branch offices (benchmarking).

The future of process mining lies in the combination of process analysis with intelligent algorithms. For example, the Celonis Proactive Insights Engine (Pi) integrates machine learning and Artificial Intelligence techniques, enabling automated identification of process weaknesses and their causes for users [13]. Building on this information, the user can retrieve intelligent recommendations for process improvements.

Given the described features, process mining plays an important role in the context of Robotic Process Automation (RPA), in particular when it comes to assessing RPA potential of processes and developing respective applications [14].

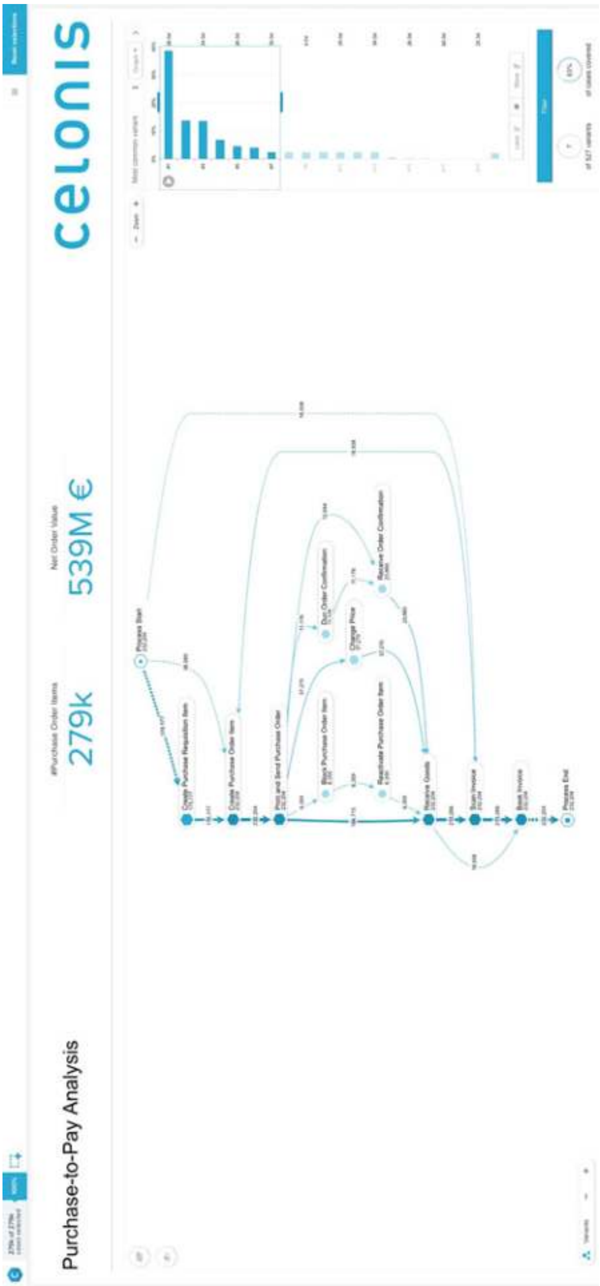


Figure 7.11: Display of the seven most frequent process variants (Screenshot of Celonis Viewer/Variant Explorer with permission of Celonis SE.)

Figure 7.12: Display of additional information (Screenshot of Celonis Viewer/Analysis with permission of Celonis SE.

7.3.5 Continuous Improvement

The aim of monitoring, evaluating and reporting process instances is to generate and provide management information on process behavior. The addressees can analyze it, investigate the causes of deviations from target values and derive short-, medium- and long-term needs for action and restructuring measures.

In the following we discuss typical changes in process design, the goal of which is to have positive effects on process behavior with respect to optimization. For illustration we refer to the loan application example in chapter 5.

Parallelism and overlapping execution. Logically and technically independent activities can be executed completely or partially simultaneously and by different task holders. Although the number of different activities can increase as a result of the splitting, executing these in parallel often accelerates the process. For example, a review step in loan application processing could be broken down into the creditworthiness check for the customer and the value check for the object to be financed, and these could be run in parallel.

Aggregating activities. Aggregating as the opposite of splitting means that activities that were originally carried out separately and by different task holders are now carried out by a single task holder. This reintegration of tasks reduces the division of labor and thus, for example, reduces interfaces. The number of activities in a business process and its model decreases as a result of the grouping and the sequence of steps in the relationships between the activities change. In the credit application process, the (re-) combination of the two checking steps could again be reasonable, in case the automation of the customer creditworthiness check previously described outweighs the time gained through parallelism.

Changing the sequence. The sequence of events in relationships between activities, or between groups or bundles of activities, can possibly be reversed, which may have advantages in terms of time, cost or capacity. In the loan application example, the creditworthiness check should be accomplished before the value check of the object to be financed, because the latter is not necessary, or the entire process ends, if a potential customer is not creditworthy. The sequencing of checks contrasts to running them in parallel. Therefore, in order to choose one of these variants, it would be helpful to know how often an application is rejected due to a lack of creditworthiness, and thus how many resources are wasted through a parallel object value check in these cases.

Elimination of activities. Verbal discussions, process mining, path analyses, and simulation experiments can reveal activities that are not needed (dead paths), activities that are very rarely carried out, and activities in which hardly any value is added or that are inefficient. The number of activities of a process decreases due to elimination and the structure of the relationships change. An example of eliminating low-value-adding activities could be the omission of an additional credit application approval by the department management for amounts over €200,000 as mentioned in section 5.2.

Elimination or reduction of cycles. When business transactions are iterated along cycles of activities, the lead times of processes are generally increased, which often leads to waiting times during execution. With path analysis, such cycles can be identified and localized, and possibly eliminated by changing the process design. During loan application processing, for example, the loan officers may repeatedly have to ask the prospective customer for information because there is a lack of sufficient details to assess the financing project. If the electronic application form requires input (mandatory form fields, annexes, etc.) for such information, the workflow system can prevent the submission as long as information is missing. Although this does not ensure that the applicant will provide the correct information in a complete and valid manner, the probability that he will do so increases, and the number of enquiry loops necessary due to missing details will most likely decrease.

Insourcing and outsourcing. Under certain circumstances, it may make sense to have entire business processes, or parts of them, carried out by specialized external service providers rather than by the own organization. External partners might accomplish tasks in a more cost-effective way and/or much faster, e.g., due to economies of scale. In this way, fixed costs can often be replaced with variable costs through the use of partner services. For example, instead of employing its own experts for property valuation, the bank could, on a case-by-case basis, engage an architectural firm. If the right conditions are in place, the reverse route can also bring advantages, for example in case previously outsourced activities can be organized more economically internally, e.g., because interfaces and transaction costs are no longer necessary.

Automating. Technical progress opens up possibilities to have manual work steps supported by IT applications and machines, e.g., robots, or to have them executed completely automatically. This is particularly relevant for time-consuming, less motivating and error-prone activities. In order to implement automation options, the development and market for corresponding technologies needs to be continuously monitored. In doing so, suitable solution modules can be identified and included in process adaptations. As an example, the credit bureau's web service can be used for obtaining customer information. Its integration not only fosters automation but is also an example of partial outsourcing.

Reduction of interfaces. Naturally, process execution based on the division of labor has interfaces on the organizational and technical level. It involves various organizational units and external partners, who often use heterogeneous IT systems and tools to generate and exchange intermediate results which are sometimes linked to different media. The consequences are media disruption and associated duplication of work effort, transmission errors, loss of time, costs, etc. Reducing the number of interfaces counteracts these deficiencies. It can be achieved through organizational changes such as reintegration or elimination of activities and changes in the allocation of tasks to task holders. On the technical level, integrated IT systems such as Enterprise Resource Planning software and workflow applications are helpful. In the loan application example, the partial transfer of signature authority to the processing level has changed the tasks assigned to the clerks and the department management. As a consequence, one branch of the process flow could be eliminated, and the corresponding set of interfaces reduced.

Since many of the mentioned optimization approaches are mutually interdependent, the effects of a measure on other design factors must always be taken into account. For example, outsourcing of process parts can increase the number of interfaces and increase efforts for service management. These effects counteract the advantages of outsourcing and always require an assessment of associated advantages and disadvantages.

References

1. Bonitasoft Documentation. Accessed February 2018, from https://documentation.bonitasoft.com/bonita/7.6/_application-and-process-designLast
2. Lawall, A., Schaller, T., & Reichelt, D. (2013). Integration of dynamic role resolution within the S-BPM approach; S-BPM ONE-Running processes. In *5th International Conference, S-BPM ONE 2013; CCIS*. Berlin: Springer.
3. Rademakers, T. (2012). *Activity in action*. Maning Publications.
4. Activity Documentation. Last accessed February 2018, from https://www.activiti.org/docs
5. AKKA Documentation. Last accessed February 2018, from https://akka.io/docs/
6. Newman, S. (2015). *Building microservices*. O'Reilly.
7. Bonér, J., Farley, D., Kuhn, R., & Thompson, M. *The ReactiveManifesto*. Last accessed February 2018, from https://www.reactivemanifesto.org/
8. Hollingsworth, D. The workflow reference model document number TC00-1003; *Workflow management coalition*. http://www.wfmc.org/standards/docs/tc003v11.pdf
9. Fleischmann, A., et al. (2017). An Overview to S-BPM oriented Tool Suites. In M. Mühlhäuser & C. Zehbold (Eds.), *S-BPM ONE - 9th International Conference*. ACM. ISBN 978-1-4503-4503-1.
10. Schmidt, W. (2013). Business activity monitoring. In P. Rausch, A. Sheta, & A. Ayesh (Eds.), *Business intelligence and performance management – Theory, systems and industrial applications* (pp. 229–242). London: Springer.
11. ibid., p. 231.
12. van der Aalst, W. (2016). *Process mining – Data science in action*. Berlin: Springer.
13. Veit, F., Geyer-Klingeberg, J., Madrzak, J., Haug, M., & Thomson, J. P. (2017). The proactive insights engine: Process mining meets machine learning and artificial intelligence. In *Business process management conference 2017*. Barcelona.
14. Geyer-Klingeberg, J., Nakladal, N., Baldauf, F., & Veit, F. (2018). Process mining and robotic process automation: A perfect match. In Casati, F. et al., *Proceedings of the dissertation award, demonstration, and industrial track at BPM 2018*. Last accessed on June 2, 2019, from http://ceur-ws.org/Vol-2196/BPM_2018_paper_28.pdf

Industrial Use Case 8

The following case study shows how some of the previously described BPM concepts, methods and languages were used to improve a core process of a company. On the basis of the individual aspects of the open BPM cycle their instantiation and intertwining will be demonstrated as occurring in a real industrial development project. In the following, the course of the project is described in a structured way. The respective activities of the bundle of activities in Figure 8.1 are noted in the margin of the use case description. The activities of each activity bundle are not only carried out with a high degree of overlap, there are also frequent shifts with respect to the considered focus throughout the project.

8.1 Background and Setting

ENGEL is a traditional Austrian manufacturer of injection moulding machines and was founded in 1945 by Ludwig Engel. Following the introduction of the first injection moulding machine in 1952, ENGEL had developed into the world market leader by 2016 with a total sales of 1.36 billion euros. The fully owner-managed company employs around 5,900 people worldwide in 9 production plants and over 85 branches [1].

ENGEL is a strongly customer-oriented company with a focus on flexibility and innovation.

> ***Analysis:*** *Determining the Key Performance Indicators based on the business model and strategy*

The strong orientation toward customer needs and the ongoing development toward shorter delivery times has led to the definition of a company-wide goal:

A. Fleischmann et al., *Contextual Process Digitalization*,
https://doi.org/10.1007/978-3-030-38300-8_8

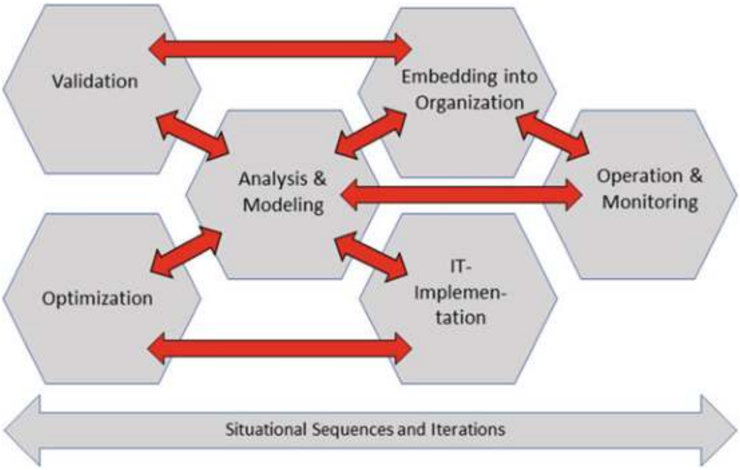


Figure 8.1: Shift between activities in the sample use case

reduction of the total process lead time for all variants of a standard component for injection moulding machines by 30%.

A project team was set up to survey and analyze the existing process and to implement the necessary improvements. The first step in any optimization process is to capture the actual status to gain more detailed knowledge of the material and information flows, and of any factors that may affect the process. At the start of the project there was hardly any explicit information about the overall process, the detailed process steps, or the process actors involved.

Initially, it was only known that the process involved two production sites in two different countries (Factory A and Factory B) and three relevant product groups:

> ***Analysis:*** *A look at the "world" as it is.*

– Product 1: The finished frame for the injection moulding machine is the initial component of each machine. Frames are assembled in Factory A and consist, among other things, of Product 2. The total lead time for Product 1 comprises order processing, production, and delivery of components (Product 2) and subcomponents (Product 3).
– Product 2: The so-called 'raw' frame with oil tank. This main component is still mechanically unmachined. It is assembled in Factory B from Product 3. There are several dozen variants of Product 2, depending on customer requirements.
– Product 3: A kind of building kit of saw blanks which are sawn off from bar stock in Factory A for each variant.

> ***Analysis&Modeling:*** *Outline the current process*

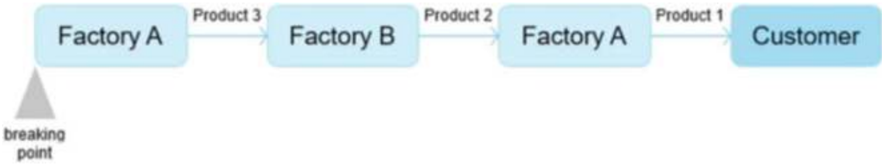


Figure 8.2 Supply chain between factories

Figure 8.2 shows a rough diagram of the supply chain. The triggering event is the order submitted to Factory A from the production department (as internal customer). Factory A saws the blanks and then delivers the necessary parts (Product 3) for Product 2 to Factory B for assembly. The intermediate Product 2 is delivered to Factory A for assembly and then subsequently delivered to the internal customer (production line for the injection moulding machine).

This process is controlled by orders that are exchanged between the factories. When an order arrives at a factory, it is entered into the factory's ERP system, and a production order is created with a corresponding delivery date. An order is not productively effective until this entry has been made. If the process up to the entry takes longer than 2 working days, the production order does not reach production on time because the order lead time is set to two working days.

In the initial situation, approximately 95% of all orders between the factories were entered too late, i.e., more than two working days passed between the arrival of the order in the factory and the actual entry. These orders then had to be processed manually with enormous additional effort, which led to an internal delivery reliability for Product 2 of only 39%. This in turn resulted in problems with production planning in Factory A and further delays in the production of Product 1.

Analysis: *First rough planning and diagnosis of enterprise architecture: No fundamental changes to the organization and the IT*

In order to achieve the specified goal of reducing the lead time by 30%, a time frame of only 10 weeks was specified. The two factories were located in two different countries with different languages. The tight timeframe led to further restrictions for the project; new software solutions or technologies could not be introduced on-the-fly. The introduction of such far-reaching changes is a strategic decision and requires a lot of manpower, budget, risk management, and time. This meant that changes to the existing processes had to be implemented within the existing organization and IT environment. Only after achieving the original project's objective it would be possible to implement further measures for additional improvements.

8.2 Implemented Measures

Analysis: *Process analysis and selection of a modeling language*

Apart from a superficial description of the material flow between the factories, no explicit process information was available to the project team. Since proper process documentation is essential for understanding the overall process and identifying optimization potential, the next step was to document and analyze the already established production process using Value Stream Analysis (VSA) [2, 3], i.e., the standard tool for documenting production processes in the company.

For an initial illustration of the process, it was necessary to select a suitable representative product which covered the basic production steps and most of the material flow. The project team decided to use a variant of Product 2 for a first as-is analysis. This selection was based on an ABC analysis of all product variants and the corresponding work plans. The variant selected for Product 2 accounts for 30% of total production. It had the most complex work schedules and the highest total lead time of the three defined product groups.

Analysis&Modeling: *Modeling of a part of the as-is process*

By tracking the material flow at both production sites, collecting relevant KPIs (inventory levels, production lead times, customer cycles, etc.), and interviewing the responsible employees, the project team was able to create a Value Stream Model (VSM) for Product 2. Figure 8.3 shows the VSM of the production process and its hierarchical structure.

Analysis&Modeling: *Identification of optimization potential*

The results of the Value Stream Analysis were as follows:

- The production process of Product 2 consists of two main steps: production of the base frame and production of the oil tank. Once the oil tank is finished, it is mounted to the frame to produce Product 2.
- The oil tank and the frame are manufactured separately. This means there is no coordination between the two production lines once a job has been started.
- Due to the lack of production planning, unfinished stocks are piled up with long waiting times. Consequently, only about 10% of the total lead time is production time.
- The lead time can be reduced by optimizing production and workload planning. However, the identified improvements are not sufficient to achieve the project goal.
- The project team has identified high potential in the processing of orders as part of the information flow.

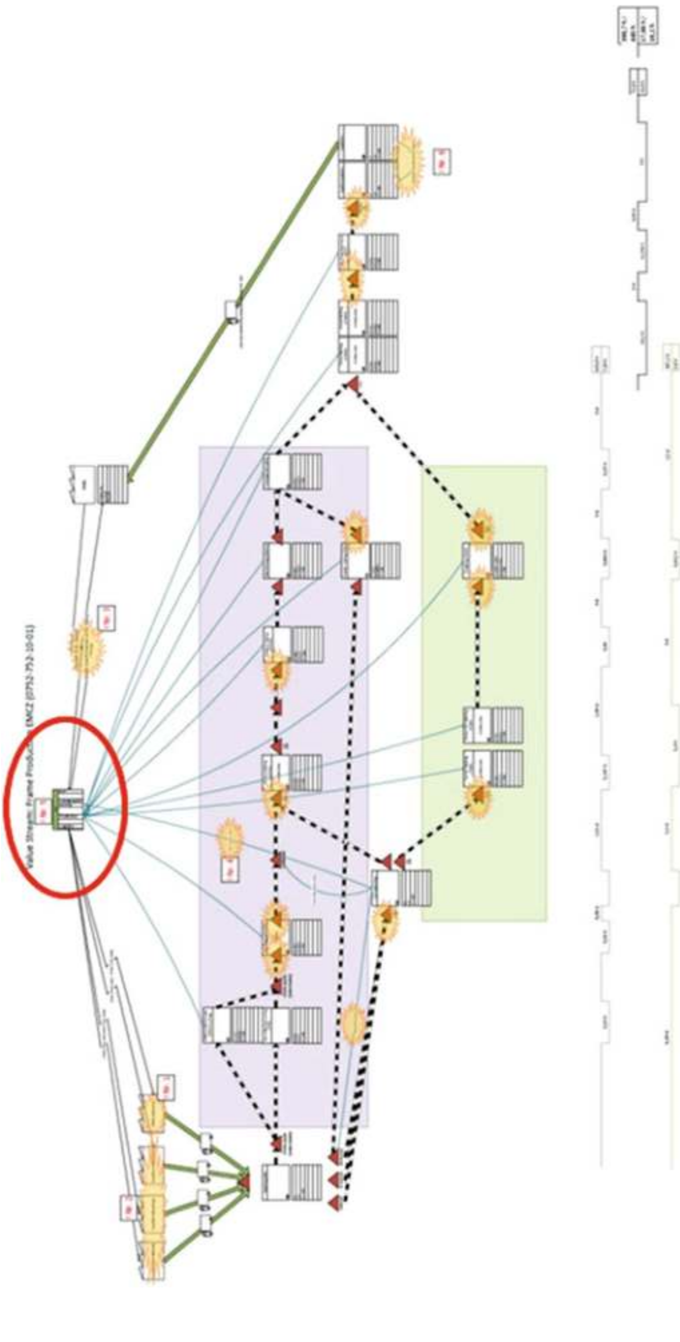


Figure 8.3 Value Stream Analysis of the production of "Product 2"

Analysis&Modeling: *Detailing the process model*

In addition, the project team could obtain more detailed information on the ordering process between the factories and extend the process description accordingly:

- The demand for Product 1 originates from Factory A, from where an order for the required variant of Product 2 is sent to Factory B.
- The order for Product 2 is processed and the procurement of the required components, including Product 3, begins.
- Factory B now orders Product 3 from Factory A.
- The order for Product 3 is recorded in Factory A, the parts are cut and sent to Factory B.
- As soon as Product 3 arrives at Factory B, the production of Product 2 starts and the finished Product 2 is then delivered to Factory A.
- Once Product 2 arrives at Factory A, the production of Product 1 begins.
- This process is the same for each machine, regardless of the variant of Product 1.

The Value Stream Analysis enabled the project team to identify two potentials: The lack of production synchronization and the non-optimized order processing. However, production synchronization is directly linked to the respective production planning process. During a short analysis of the planning process the project team came to the conclusion that long-term improvements would only be possible if the procedures in the planning process were completely reorganized and restructured, and the mindset in itself was changed. Although this would have been a necessary modification, the project team could not implement it within the timeframe of the project. The team members therefore focused on the ordering process and its optimization potential, as they identified it as a possible 'quick win' for their project.

Analysis&Modeling: *Modeling still missing aspects*

This was the starting point for a comprehensive process survey. However, the VSM still lacked relevant process information to describe the overall process and the corresponding information flow in detail, such as,

- information on the interactions between the parties involved
- information on which steps in the process are automated and which steps are performed manually.
- concrete information on the transactions used in the SAP ERP system (SAP / R3)
- information about the timelines of the information flow
- verification of the information provided

Analysis&Modeling: *selection of an additional modeling language*

Since the VSA focuses on the production processes and especially describes the material flow between the process steps, the project team quickly realized that it needed a different approach. They had to use an additional method to describe the flow of information in a level of detail that allowed an accurate analysis to be conducted. The next step was to create a process model of the people and SAP systems involved in the process and their respective interactions and information flows. To supplement the company's existing VSM, the project team introduced Subject-oriented Business Process Modeling (S-BPM) [4].

The project team chose S-BPM because of past experience and the problems they had had with flowcharts and swim lane diagrams, which are occasionally used alongside the VSA.

Analysis&Modeling: *Rationale for the addition of Subject-oriented Modeling*

In previous use of swim lanes, the process models either provided an overview of the process (and were not detailed enough for a thorough process analysis), or the process models were so detailed that it became very difficult to keep track of them.

Furthermore, in the team's experience, swim lane models are not suitable for visualizing the individuals involved and their interdependence in a transparent way, especially if there are more than five or six participants. Experience has also shown that the people involved in the processes, their individual approaches, their knowledge and their experience are a decisive driving force for the processes of a company and are indispensable for successful processes [5, 6].

Since the way in which the flow of information between process actors is organized has a significant impact on business process performance ([7], the project team focused on the S-BPM representation of processes. It concentrates specifically on the point of view of the involved process actors, so-called "subjects", and their interaction in the process environment. A subject can be a machine, an IT system or a person. Subjects are abstract agents without any indication of how they are finally implemented.

Analysis&Modeling: *Description of S-BPM and tools used*

The S-BPM modeling method uses two levels to describe processes, the Subject Interaction Diagram (SID) and the Subject Behavior Diagram (SBD). The interaction between the subjects is visualized in the SID, which describes the exchanged messages between the involved parties (subjects). The SID provides a process overview and helps to identify the role of a subject in the overall process.

The Subject Behavior Diagram describes the individual process steps of a participating subject's role in the process. The behavior of a subject is described by three different actions (the so-called "states" in S-BPM) it performs: "send", "receive", and execute an internal task ("function"). Since the relevant part of a process for a process actor is encapsulated within the respective subject, each SBD represents an independent actor within the process. It is not necessary to model the whole process at once or in a strict sequence, e.g., if information is missing or not available, or if part of the process is not relevant for the task at hand.

The S-BPM notation consists of five symbols which are defined by their meaning and not by their form, although recommendations exist in associated literature. Two symbols are used in the SID to represent a subject (e.g., a rectangular shape) and the interaction (e.g., an arrow), and three symbols (most often rectangular shapes in different colors) in the SBD to represent each state type (send, receive, function).

One advantage of the two levels and such a simple modeling notation is the possibility to model processes simultaneously in a top-down and/or bottom-up approach, whereby the traditionally separate areas of Business Process Management and Lean Production [8] are combined dependent on the various demands and requirements, and on the available depth of detail of the information. Although there are dedicated software solutions for modeling S-BPM, the project team used their own individual MS Visio template. This allowed them to immediately focus on process modeling and analysis without investing resources and time in the application of an external technical solution - a common mistake in many companies when process modeling is implemented [9]. Based on more detailed personal interviews, the project team created a model of the first level of the process, the SID. The resulting process model of the current situation showed, as expected, that the logistics and production process were very complex. Approximately 40 subjects were involved in the production, production control and logistics of all three products in both factories.

Analysis&Modeling: *Detailed modeling using S-BPM*

The final version of the resulting as-is SID and the notation used are shown in Figure 8.4. The SID shows the general communication structure of the process and which subjects exchange which messages with each other. The concrete names of the subjects or the content of the exchanged messages are not relevant for the further understanding of the implemented measures. The rectangular shapes represent the various subjects involved in the process, and the arrows represent the messages exchanged between the subjects. In order to distinguish between the two factories, the project team marked the corresponding subjects in green for Factory A and in orange for Factory B. In addition, they marked the subjects that represent SAP systems with hatching to highlight the already digital parts of the existing process.

Although both plants use the same ERP system, the project team dealt with it separately according to the respective departments for a more structured visualization (SAP System A, SAP System B, and SAP System A Dispatching).

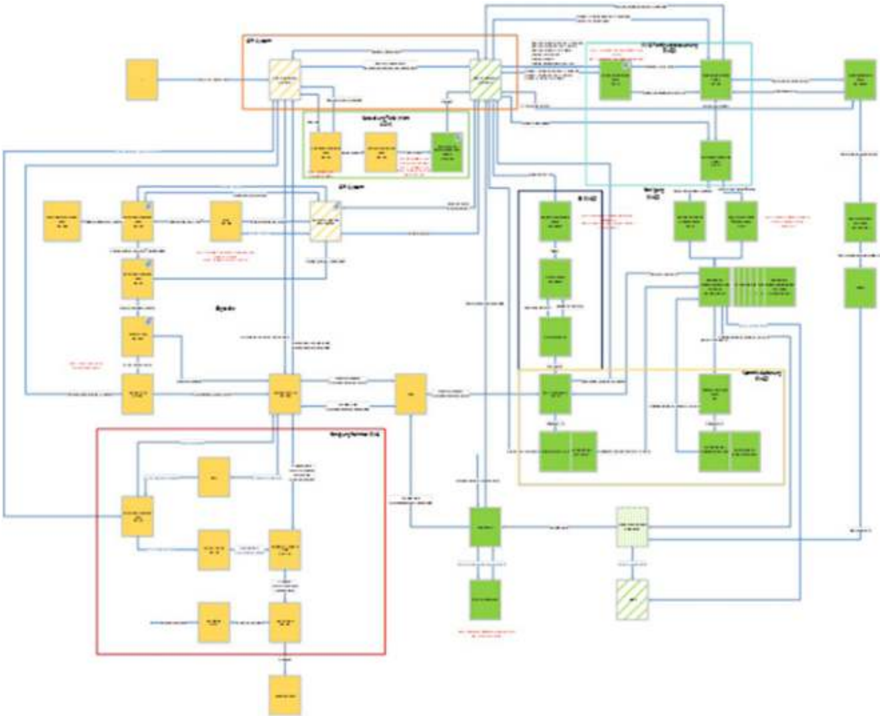


Figure 8.4: Communication structure (Subject Interaction Diagram) of production and the ordering process

Optimisation: *Identification of further optimization aspects*

Due to the number of subjects involved and the complexity of the entire process, it would not have been practicable to collect and model all subject behaviors for the next steps without a defined framework.

To define such a framework, the project team used the now existing SID to identify and analyze the main nodes and bottlenecks of the process with respect to the corresponding products (Figure 8.5).

The most striking part of the process was the order processing itself. The processing of the order for Product 1 by Factory A, the order processing and the procurement of Product 2 by Factory B, and the production of Product 3 in Factory A employed up to 12 subjects (3 SAP systems and 9 persons) and took up to 15 working days. In addition, only 65% of Product 2 was completed on time because order processing took too long and orders were received too late at the production center (approximately 95% of all orders). This had a direct impact on the production of Product 1 and on process stability. The delivery times could only be met with a lot of effort in production.

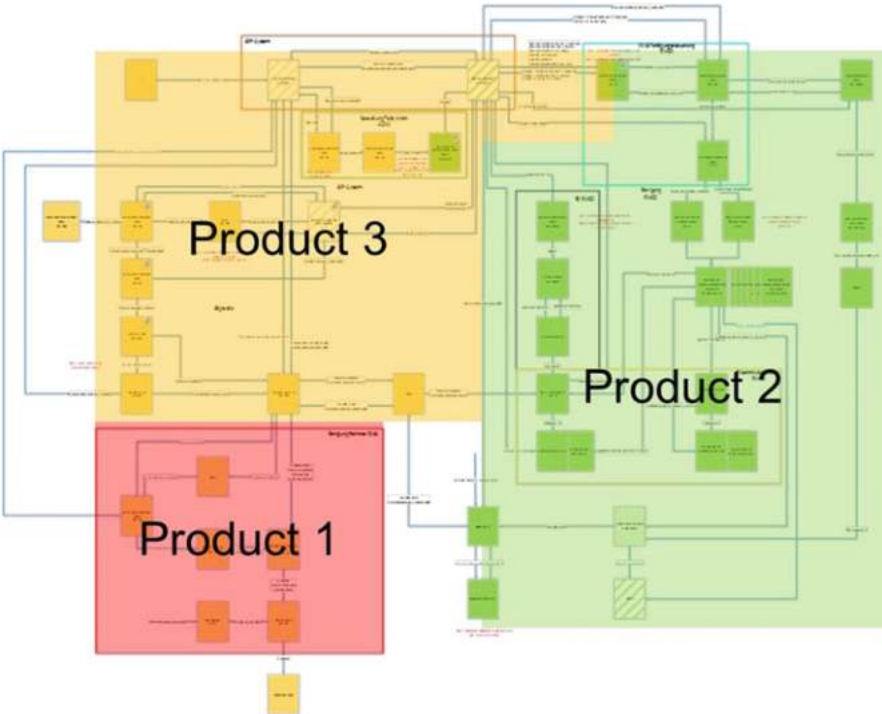


Figure 8.5: Relationship between the communication structure and the corresponding products

The project team decided to focus on this material procurement process because the process itself is very complex and time-consuming relative to the complexity of the components provided (Product 3). The project team defined the scope of the process survey as follows:

- The focus is on the logistics departments of Factory A and Factory B. This also includes the production of Product 3 in Factory A, as this is directly integrated in the logistics organization and is therefore part of the process and production of Product 2 in Factory B.
- The material procurement in Factory A and the actual assembly of Product 1 in Factory A are no longer part of the survey (see Figure 8.5 for a visualization of the process and the corresponding products).

***Validation:** Validation of the process model*

The project team examined the relevant process steps by interviewing the involved employees in individual interviews, accompanying the employees during the process, and at the same time modeling the Subject Behavior Diagrams (SBD) in the presence of the interview partners. This enabled the project team to describe the process flow for

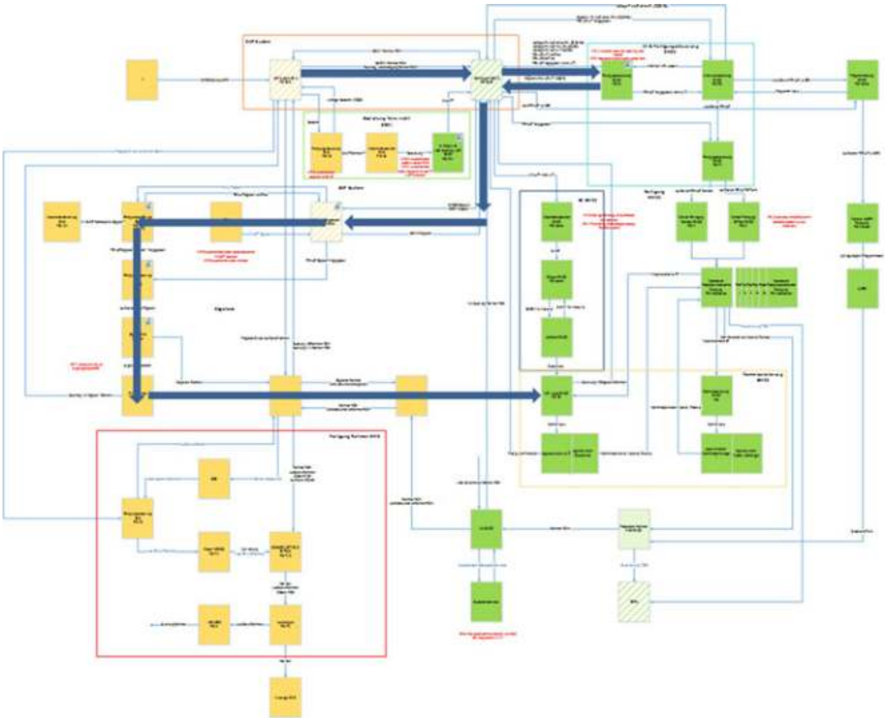


Figure 8.6: Communication structure (SID) of the process for the production of Product 3

Product 3 in detail (see arrows in Figure 8.6), document detailed information about the SAP ERP system and the transactions used, and enable the interview partners to directly accompany the process modeling procedure while verifying the model.

***Analysis**: Further process modeling and as-is analysis of process implementation*

After the SAP transactions were clearly described in the SID and SBD and a dummy request was tracked through the system, the project team specified the different steps of the SAP system. This allowed the team to distinguish between automated and manual steps, validate the verified process model, and document actual process lead times. For the SBD rectangular shapes and different colors were used to represent the three states: red for the "send" state, green for the "receive" state, and yellow for the "function" state (see also Figure 8.7).

Figure 8.7 visualizes the process behavior of an employee who processes production orders in Factory B. This employee checks whether production plans exist for planned production orders. Subsequently, all planned production orders with available production plans are combined according to a defined set of rules and

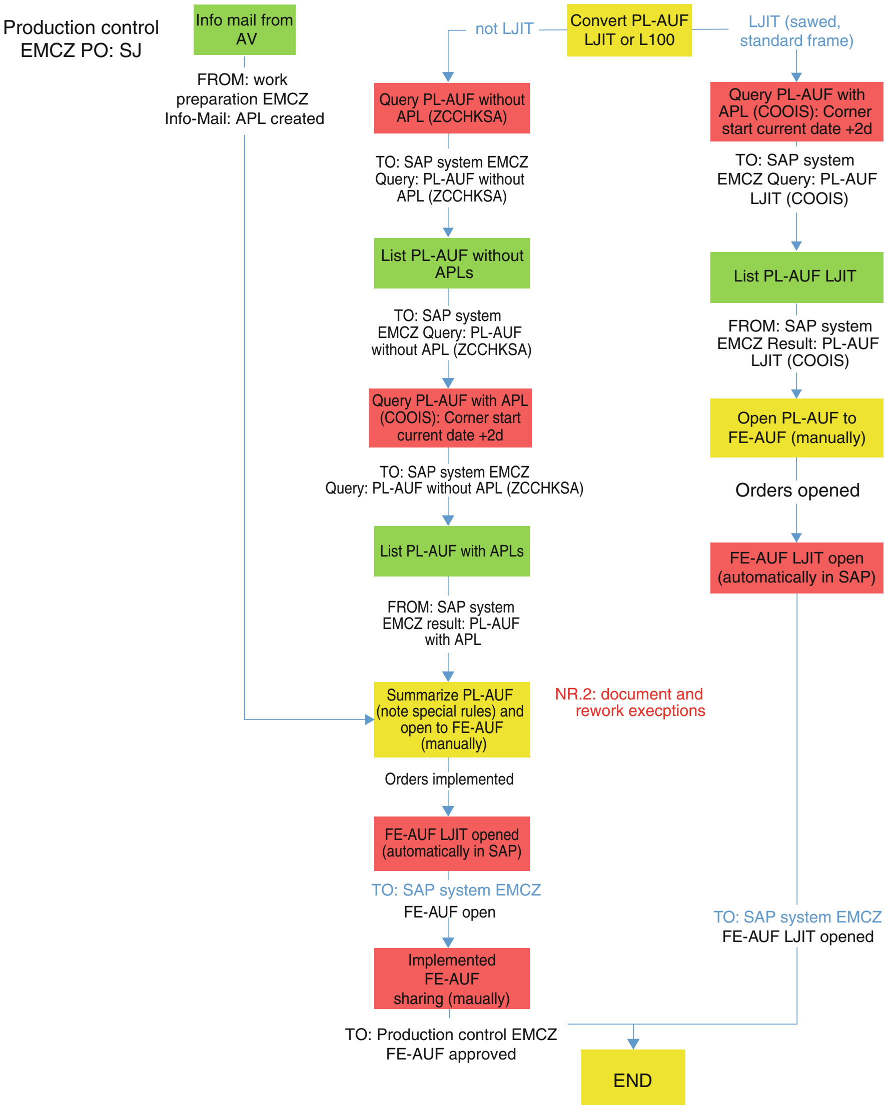


Figure 8.7: Example for a behavioral description of an employee

released for production. The employees do this manually for each production order, with several thousand orders per day. Product 3 alone causes a total workload of approximately 7 hours per day.

> ***Validation:*** *Identification of optimization potential on the basis of the as-is model*

The total hours worked for the entire survey, all interviews, and the time required to complete and review the process models was approximately 200 hours. This is a relatively low effort compared to other process optimization projects in view of the complexity of the process models and the depth of detail investigated. The now available detailed knowledge about the subjects involved enabled the project team to create a time schedule for the process based on the collected data and the data documented in the SAP system (order times, delivery times, etc.). This schedule includes all organizational and production steps and their respective lead times. For example, the lead time for one of the Product 1 variants, from order acceptance in Factory B to delivery of the finished Product 1 to the assembly line in Factory A, was approximately 30 working days (see Table 8.1).

Organisational and IT-Implementation:

In cooperation with the employees, the existing work plans were revised, updated and improved. These changes led to shorter lead times for the individual work steps, as well as to a reduced number of work steps as a result of merging existing steps. In this case, a reduced number of work steps means fewer subjects as well as fewer behavioral states. During the analysis, the project team identified several similar process steps that were carried out differently in Factory A and Factory B. In one factory, necessary process steps were executed manually, while the same steps were executed automatically by the SAP system in the other factory. In addition, existing automated SAP batch jobs were interrupted because the required manual input between jobs was missing. These jobs were scheduled at two defined times during the working day, and if manual input was missing at that time, the entire order would have to wait up to a full working day. This could happen several times with each order for different jobs, which could ultimately lead to a delay of several working days.

The subjects described provided precisely defined processes that describe all relevant process steps in the SAP system, all required SAP transactions, who executes these transactions, and the interaction between the system and employees. This detailed process documentation allowed the company's IT department to directly implement relevant subject behavior to create new standardized, digitalized, and automated processes, as well as enabling them to revise process steps and streamline the processing schedule of existing batch jobs for both factories. This included steps such as order acceptance, order entry, order opening, order release in both factories, and delivery of the production documents to production. The automated order processing enabled the company to process Product 3 in Factory A on an order-related and timely basis, which in turn allowed the introduction of KANBAN inventories with defined critical parts, the reduction of the inventory of non-critical parts, and the shipment of externally purchased parts directly to manufacturing.

Table 8.1: Time consumption for as-is process

AS-IS (standard process)						Possible Range for Order Reception 1 AT																																
WD	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Receiving Order in EMCZ											1 AT																											
Opening Order Frame EMCZ	1 AT																																					
Acquisition Sawing Parts for EMCZ																max 10 + 1 AT																						
Acquisition Flame Cut Parts																	max. 7 + 1 AT																					
Acquisition Components																max. 10 + 1 AT																						
Releasing FE-AUF Frame																			1 AT																			
Opening Order Oil Tank																1 AT																						
Releasing FE-AUF Oil Tank																1 AT																						
Acquisition Components Oil Tank									15 + 1 AT																													
Production Oil Tank																				4 + 1AT																		
Assembly Oil Tank																								3 + 1AT														
Cutting and Bending																						2 AT																
Production Frame EMCZ																											7 AT											
Frame Delivery to EMS																															1 AT							
Ware Entry EMS																																1 AT						
Production Frame EMS																																			3 + 2 AT			
Requirement Date Frame EMS																																						1 AT

8.3 Achieved Results

Operation and monitoring: *Measured Key Performance Indicators*

One of the achievements was a new warehouse strategy and a reassessment of the inventory, which made it possible to revise the entire inventory and implement a KANBAN system for critical parts of Product 3. The newly created KANBAN inventory and the higher value of the affected parts led to an overall increase in fixed capital of approx. 209%. However, this had only a minimal influence on the existing stock value of approx. €10,000 in total. This new strategy increased availability and reduced delivery times for all parts purchased from external suppliers. The interruptions in the production of Product 2 due to missing parts could originally last up to 15 working days. After the changes, all required components were available within one working day either directly on site or via the supplier's safety stock - an enormous improvement in process stability and a reduction in rotating stock compared to a relatively small increase in inventory.

Value Stream Analysis is an established tool in the company for the analysis of production processes. Although associated literature ([3, 10]) and external consultants have often highlighted VSA's ability to describe not only material flows but also information flows, the company's expectations were not met when trying to document and visualize these. When most relevant information is available, the administrative processes and information flow can be described with a Value Stream Model. Based on the project team's experience, however, a VSA is not suitable for a representation of the flow of information with partially abstract information. S-BPM provided the project team with an easy-to-learn modeling notation that can still provide very accurate and detailed process models. The employees involved were able to independently understand, read and correctly interpret the S-BPM notation and began to verify their own process models (subject behavior) without the input of the method specialists. This led to a high acceptance of the process survey and the subsequent changes in the process, as the employees were directly involved in the documentation and the optimization steps.

The restructuring and digitalization of previously manual process steps has led to a standardized process and a reduction of the subjects involved from 12 to 8 (see Figure 8.8). Fewer subjects mean fewer interfaces in the process, which in turn reduces process complexity and increases process stability and transparency. In addition, employees were freed from time-consuming and repetitive tasks.

The increased degree of digitalization and the newly planned process has led to a new process lead time of 2 days for order processing (originally 5-10 days). Thanks to the detailed and clearly defined process, the company's IT department was able to implement the process changes in the existing system environment within just 3 working days. The production and shipping process for Product 3 was reduced to 3 days, from 5-6 days originally. This means that the project team was able to reduce the total lead time for Product 3 from 11-15 working days by 87% to just

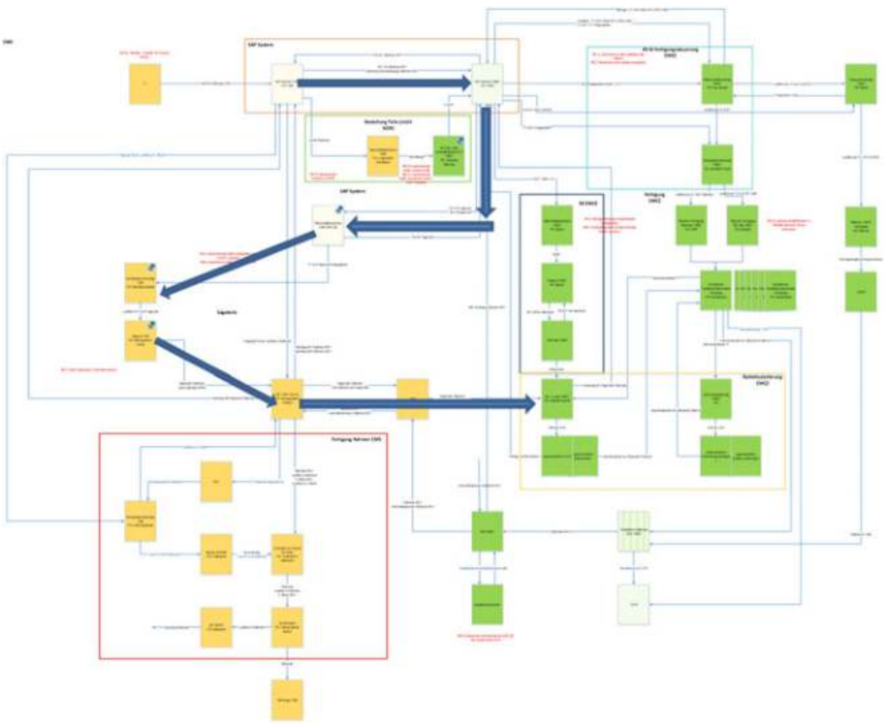


Figure 8.8: Communication structure of the updated process for Product 3

2 working days. These changes led to an increase in on-time delivery for Product 3: delivery reliability rose to 89% just four weeks after implementation and to 97% after one year.

The relatively long period of time needed to process orders for Product 3 in the initial phase meant that most orders arrived at Factory A too late or at very short notice. The newly created automated SAP processes resulted in faster processing of Factory A orders within the associated Factory B departments. This in turn led to a shorter ordering time for Product 3 and an earlier start of production for other components required for Product 2. The result was a reduction from an initial 95% of orders registered too late to only 12%, which again significantly increased process stability and quality, and reduced the need for troubleshooting in both factories. The total lead time of the production and ordering process for Product 2 could be shortened by 7 working days (approx. 38%), from 19-23 days to only 12-14 days.

The conversion of manual work into automated, digitalized processes running in the SAP system has led to a reduced workload of the employees involved from 5-6 hours per day to as little as one hour per day. The employees now only have to manually process the purchase orders for very specific components or special cases that could not be covered by the SAP system. The effects of these changes add up to a calculated process cost reduction of around €65,000 per year. The implemented

Table 8.2 Time required in the original process

AS-IS (standard process)						Possible Range for Order Reception 1 AT																																
WD	**-10**	**-9**	**-8**	**-7**	**-6**	**-5**	**-4**	**-3**	**-2**	**-1**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**
Receiving Order in EMCZ											1 AT																											
Opening Order Frame EMCZ	1 AT																																					
Acquisition Sawing Parts for EMCZ											max 10 + 1 AT																											
Acquisition Flame Cut Parts														max. 7 + 1 AT																								
Acquisition Components											max. 10 + 1 AT																											
Releasing FE-AUF Frame																			1 AT																			
Opening Order Oil Tank																1 AT																						
Releasing FE-AUF Oil Tank																1 AT																						
Acquisition Components Oil Tank		15 + 1 AT																																				
Production Oil Tank																		4 + 1AT																				
Assembly Oil Tank																							3 + 1AT															
Cutting and Bending																						2 AT																
Production Frame EMCZ																								7 AT														
Frame Delivery to EMS																															1 AT							
Ware Entry EMS																																1 AT						
Production Frame EMS																																	3 + 2 AT					
Requirement Date Frame EMS																																						1 AT

Table 8.3 Time required in the revised process

TO-BE

WD	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
Receiving Order in EMCZ											1 WD	Reserve																										
Opening Order Frame EMCZ											1 WD																											
Acquisition Sawing Parts for EMCZ													1 WD																									
Acquisition Flame Cut Parts													0 WD																									
Acquisition Components													0 WD																									
Releasing FE-AUF Frame											1 WD																											
Opening Order Oil Tank											1 WD																											
Releasing FE-AUF Oil Tank											1 WD																											
Acquisition Components Oil Tank													0 WD																									
Production Oil Tank													4 + 1 WD																									
Assembly Oil Tank																		1 + 1 WD																				
Cutting and Bending															2 WD																							
Production Frame EMCZ																	5 WD																					
Frame Delivery to EMS																						1 WD																
Ware Entry EMS																							1 WD															
Production Frame EMS																								4 WD														
Requirement Date Frame EMS																												1 WD										

improvements and corresponding changes at the process level reduced the lead times of Products 2 and 3 and led to a shortened overall lead time for Product 1: from initially 26 to 33 working days to 18 to 20 working days. This is a total reduction of approximately 60% for the entire ordering and production process (see Table 8.2 and Table 8.3).

Not only was the project team able to achieve the original target of a 30% reduction in lead time, but rather to more than double this reduction by digitalizing and automating the process steps and the corresponding information flow. In addition, this has led to a reduction in rotating stock with a total value of several hundred thousand euros over the entire process.

These results show that it is possible to significantly reduce lead time and manual workload by optimizing and digitalizing the flow of information. The increased degree of digitalization and the associated process transparency can help to achieve further improvements and to better understand the processes in future analyses [11]. Although the implementation of specialized S-BPM tools was deliberately avoided in the case at hand, their introduction, thanks to the S-BPM methodology, could provide the basis for an even more comprehensive digitalization of existing processes. The S-BPM method and supporting modeling tools enable a direct transformation of process models into running processes [4], which could significantly reduce the effort for the future digitalization of processes.

References

1. ENGEL AUSTRIA GmbH. *Facts & figures* [Online]. Last accessed February 2018, from http://www.engelglobal.com/en/uk/company/facts-figures.html
2. Rother, M., & Shook, J. (2004). Learning to see – Value-stream mapping to create value and eliminate muda. Lean Enterprise Institute.
3. Erlach, K. (2010). *Value stream design. The lean way to the factory*. Berlin: Springer.
4. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., & Börger, E. (2012). *Subject oriented business process management*. Heidelberg: Springer.
5. Liappas, L. (2006/2007). AGILITY by ARIS business process management. In *Yearbook business process excellence* (pp. 43–56). Berlin: Springer.
6. Riempp, G. (2004). *Integrated knowledge management systems, architecture and practical application*. Berlin: Springer.
7. Kock, N., Danesh-Pajou, A., & de Luca, D. (2009). Communication flow orientation in business process modeling and its effect on redesign success: Results from a field study. *Decision Support Systems, 46*, 562–575.
8. Kannengiesser, U. (2014). Supporting value stream design using S-BPM. In *Proceedings of the 6th International Conference on Subject-oriented Business Process Management (S-BPM ONE 2014), Eichstätt, Germany, April 22–23*. Heidelberg: Springer.
9. Schmelzer, H., & Sesselmann, H. (2013). *Business process management in practice – Satisfying customers, increasing productivity, increasing value*. Munich: Carl Hanser Publishing House.
10. Wiegand, B., & Franck, P. (2004). *Lean Administration I – How business processes become transparent*. Aachen: Lean Management Institutes.
11. Parviainen, P., Kääriäinen, J., Tihinen, M., & Teppola, S. (2017). Tackling the digitalization challenge: How to benefit from digitalization in practice. *International Journal of Information Systems and Project Management, 5*(1), 63–77.

Index

A. Fleischmann et al., *Contextual Process Digitalization*,
https://doi.org/10.1007/978-3-030-38300-8