26.1 Introduction

“Panta rhei”—“everything flows”—an aphorism leading back to the thoughts of Heraclitus provides a metaphor for the dynamic nature of physical existence. Humans have been conceiving sets of actions ever since to deal with and survive in a dynamic environment. Sets of consecutive actions that are directed to some end are commonly referred to as a process. In fact, human societies are organized around such generic processes as crafting tools, producing energy, producing food, producing goods, building structures, making astronomical observations, communicating messages, or celebrating rituals. Process can also be perceived on the level of an individual like preparing a meal, buying a car, or simply completing a to-do list. For processes that require the participation of multiple individuals and that are intended to be executed on a continuing basis are usually embedded within an organization.

Consequently, an organization can be best described in terms of what it does and can do (cf. Rescher, 1996) and processes—in a very simple sense—are what companies do (cf. vom Brocke and Rosemann, 2013). The capability and potential performance of an organization is thus built into its processes. Processes do happen, regardless whether we manage them or not. Managing processes, therefore, intends to make sure that the organization meets the requirements in operating processes that support the strategic corporate objectives. Over the past decades, business process management (BPM) emerged
as an academic discipline that intersects with both information systems and management research. BPM represents an integrated, process-oriented management approach utilizing tools and methods for business process analysis and improvement (cf. Rosemann and vom Brocke, 2010).

The BPM discipline, as it is understood today, evolved around two main strands (Hammer, 2010):

- **Continuous change**: Earlier studies focused on continuous or incremental improvement of existing processes. Examples of this approach have been total quality management (TQM), lean management, or kaizen. Basic methodological principles have been contributed by Deming, for example, on statistical process control, dealing with a systematic analysis of processes by means of both quantitative and qualitative criteria (Deming, 1986).

- **Radical change**: By introducing the concept of business process reengineering, Hammer and Champy (1993) advocated an approach, which is based on the fundamental rational to challenge all existing processes in an organization. Here, the goal is to rethink and, if deemed appropriate, redesign business processes from scratch (so-called green field approach). Information and communication technology is considered to be the main facilitator and driver of such radical process redesigns (Davenport, 1993). The focus is less on fine grained analytical methods but more on discussing the design of “large-scale, truly end-to-end processes” (Hammer, 2010, p. 4).

Today, BPM aims to combine both approaches (Hammer, 2010), recommending that there should be an alternation of activities of continuous improvement and radical change (vom Brocke and Rosemann, 2013).

### 26.2 Core Elements of BPM

In order to enable the management of processes in an effective and efficient way, companies need to develop a wide range of capabilities. From research on maturity models (de Bruin et al., 2005), six central core elements for BPM were derived (Rosemann and vom Brocke, 2010). They are shown in Table 26.1 and are briefly explained in the following.

- **Strategy**: Process management needs to be guided by strategic organizational goals. The main question to answer: How can each process be assessed in order to be able to determine its specific value?

- **Governance**: Process management needs to be rooted in the organizational structure. The main question to answer: Who is supposed to be responsible for which process management task?

- **Methods**: Process management needs methods for process design supporting different phases of process development. The main question to answer: What methods are being used by the organization?

- **Technology**: Process management needs to be supported by technology, particularly information technology (IT), as the basis for process design. The main question to answer: What systems are available in the organization in order to be able to execute different process management tasks?

- **People**: Process management needs certain capabilities and skills on the part of an organization’s staff. The main question to answer: What capabilities and skills are available and what must be done to develop missing capabilities?

- **Culture**: Process management needs a common value system in order to effect certain changes required. The main question to answer: Is the culture of an organization and the basic attitude of its people supportive of change and innovation?

Research on BPM has shown that no core element should be neglected when BPM is to be implemented successfully in an organization (e.g., Bandara et al., 2007; Kettinger and Grover, 1995; Trkman, 2010). The challenge is to develop each core element adequately while at the same time matching all core elements with each other in a reasonable way. The context an organization is embedded in plays a crucial role here. There is no “out-of-the-BPM approach” that can be readily applied to any organization whatsoever.
Every organization has to analyze and decide very precisely what their particular BPM approach with regard to each of the six core elements should look like.

The particular BPM approach taken is dependent not only on the actual maturity level attained within the six core elements but also on the organization-specific perception of the business process concept. There are several views about the nature of business processes that require BPM activities to be approached differently depending on the view adopted. The subsequent section briefly sketches four possible views on business processes and outlines how BPM activities—in particular process analysis tasks—can be organized within a BPM life cycle. It will also be discussed how different views both enable and limit specific approaches to business process analysis.

### 26.3 On the Nature of Business Processes and Process Analysis

#### 26.3.1 Common Business Process Definitions

A typical definition of a business process as presented in a large number of BPM literature regards business processes as a *sequence of activities generating a specific value for a customer* (e.g., Davenport, 1993; Hammer and Champy, 1993). The activities contained in a business process are assumed to transform some *input* of suppliers into *outputs* for customers, whereby the transformation activities can be hierarchically decomposed into sub-processes (vom Brocke and Rosemann, 2013). *Activities* in a process are not always arranged in a strict sequence but they are rather linked though network structures.
These network structures determine the order in which entities may flow through a process. The network structure of activities is also frequently referred to as control flow, and the entities flowing through a process are referred to as flow units (cf. Anupindi et al., 2012). To evaluate process performance, process managers are often interested in analyzing the paths of individual flow units to calculate process performance measures such as flow time, inventory (total number of flow units present within a process), quality, cost, or value-added (determined in terms of added or changed flow unit attributes that customers consider important). In this regard, some process definitions account for the significance of flow units for distinguishing between processes. For example, Becker and Kahn (2011) define a process as a “timely and logical sequence of activities necessary to handle a business relevant object” (cf. Becker and Kahn, 2011). Resources and information structures are needed to enable and coordinate the transformation of inputs into outputs. Resources (like raw materials, manufacturing facilities, equipment, or employees) are required by individual activities within control flow to process flow units. An appropriate information structure supports managerial decision making within a process on an operational and strategic level. In summary, five aspects are central for managing processes (cf. Anupindi et al., 2012):

1. Inputs and outputs
2. Flow units
3. Activities and control flow structures
4. Resources
5. Information structure

Figure 26.1 illustrates how these aspects relate to the concept of a process. Figure 26.2 extends on this conceptualization by illustrating the central characteristics of a business process.

FIGURE 26.1  A process as set of activities linked through control flow relationships.
A business process is commonly regarded as a particular process type which is characterized as having interfaces to a customer or a supplier and which contains the activities that represent the core business of an organization (cf. Becker and Kahn, 2011). Business processes are thus designed and perceived as end-to-end processes (see Figure 26.2). An example of a typical end-to-end business processes is an “Order-to-Cash” process, which comprises all activities for generating and fulfilling a customer order and to collect the payments for fulfilling an order. These activities are usually divided into sub-business processes like “Sales,” “Purchasing,” “Production,” “Shipment,” and “Delivery.”

Ultimately, the application of BPM in organizations requires a “thinking in processes.” Process thinking emphasizes the significance of inter-departmental, cross-functional coordination of business activities with the aim to better serve customer needs (cf. Hammer, 2010). Traditional “functional thinking” on the contrary would rather emphasize the “local optimization” of activities within a particular function by exploiting resource efficiency gains through specialization. However, this local optimization does not necessarily yield an overall optimal coordination of all the cross-functional activities involved in a business process. "Process thinking," as mandated by BPM, stimulates optimization across an organization along defined process structures and to effectively and efficiently coordinate activities in the interest of customer needs.

26.3.2 Views on Business Processes

While the general understanding of the business process concept as outlined earlier is a useful common denominator for BPM, real-world processes are multifaceted and show great diversity according to their specific nature. BPM thus has to account for this diversity by allowing business processes to be approached from different perspectives. Melão and Pidd (2000) present four different views on business processes. According to them, processes can be viewed as (1) a deterministic machine, (2) a complex dynamic system, (3) an interacting feedback loop, or (4) a social construct. These views emphasize different important features of a business process and also resemble different assumptions about organizational life (cf. Walsham, 1991). In the following paragraphs, these views are briefly outlined along with their implications on BPM practice and research.

26.3.2.1 Business Process as a Deterministic Machine

According to this view, a process is regarded as a “fixed sequence of well-defined activities or tasks performed by ‘human machines’ that convert inputs into outputs in order to accomplish clear objectives” (see discussion earlier) (Melão and Pidd, 2000). This view holds that a process can be objectively...
perceived and described in rational and technical terms. Process descriptions focus on the process structure (routing of flow units), procedures (logical dependencies between activities, constraints, and required resources), and goals. Goodness of a process is evaluated in terms of efficiency measures like time, costs, or productivity. This view has its roots in scientific management (Taylor, 1911) and assumes the presence of stable and structured processes as can be observed in manufacturing processes or bureaucratic processes.

For a long time, this view dominated the BPM practice and research since it is mostly concerned with capturing process knowledge within semi-formal process descriptions and graphical process models (flow charts) of all kinds. This “machine view” assumes that any process type, technical or human driven, structured, and unstructured, can be unambiguously mapped to a process model by means of a dedicated modeling formalism. Exemplary modeling formalisms are discussed further in the subsequent sections and are summarized in Table 26.2. Process modeling under a machine view is applied in a wide variety of domains for various purposes, for example, in software engineering (e.g., the unified modeling language (UML) with its activity and sequence diagrams is heavily used here),

<table>
<thead>
<tr>
<th>Main assumptions</th>
<th>Deterministic Machine</th>
<th>Complex Dynamic System</th>
<th>Interacting Feedback Loop</th>
<th>Social Construction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processes can be described and designed in a rational way</td>
<td>Processes can be described and designed in a rational way</td>
<td>Magnitude of impact of policies on process performance is easily quantifiable through equations</td>
<td>Process change or process design can be negotiated in presence of conflicting interests</td>
<td></td>
</tr>
<tr>
<td>Consideration of dynamics</td>
<td>Static snapshot (of as-is or to-be situation)</td>
<td>Interaction between process components</td>
<td>Interactions between process structure and policies</td>
<td>Dynamics are considered implicitly as part of the subjective constructions of a business process</td>
</tr>
<tr>
<td>A process design is good if...</td>
<td>...the parts work efficient</td>
<td>...the process as a whole is effective</td>
<td>...the process is effective given a set of policies to be obeyed</td>
<td>...the perceived process exhibits “cultural feasibility”</td>
</tr>
<tr>
<td>BPM techniques</td>
<td>Flow charting and workflow modeling</td>
<td>Process simulation, quality management (statistical process control), business activity monitoring, and process mining</td>
<td>Process simulation and system dynamics modeling</td>
<td>Soft systems methodology</td>
</tr>
<tr>
<td>Related BPM activities</td>
<td>Process documentation, application development, certification, and benchmarking</td>
<td>“What if” analysis, discovery of previously unknown process behavior, and process control</td>
<td>“What if” analysis and gaining qualitative insight about process behavior</td>
<td>Learning about processes, engagement in unstructured process design problems, and management of “creative” business processes</td>
</tr>
<tr>
<td>Main limitations</td>
<td>Lack of a time dimension</td>
<td>Neglect of interaction between process structures and policies. Neglect of sociopolitical issues</td>
<td>Difficult to apply since the magnitude of impact of policies is hard to quantify</td>
<td>Focus on “cultural feasibility” may impede radical process designs</td>
</tr>
<tr>
<td></td>
<td>Neglect of sociopolitical issues</td>
<td>Neglect of sociopolitical issues</td>
<td>Neglect of sociopolitical issues</td>
<td>Not accessible to objective and quantitative assessment of process performance</td>
</tr>
</tbody>
</table>

TABLE 26.2 Summary of Different Perspectives on Business Processes
the modeling of automated processes in the domain of workflow management, or for certification (e.g., quality management according to the ISO 9000 requires a documentation of business processes). Moreover, process models could be used to customize enterprise systems according to individual needs of an organization.

Process modeling is the main focus in the machine view, and research in this area increasingly focuses on the understandability and usability of particular modeling notations (see, e.g., zur Muehlen and Recker, 2008). Other research activities additionally study capabilities of modeling notations to map existing business realities onto a process model (e.g., see work on modeling grammars, as in Recker et al., 2011).

The main limitations of this deterministic machine view on a business processes are its assumption that business processes can be objectively perceived and modeled in a rational way (like engineering machines). It therefore neglects the sociopolitical dimension of BPM initiatives. Moreover, business processes as seen from a machine-view represent static snapshots of an as-is or a to-be situation and thus neglect the dynamic nature of processes, which are likely to change over time. However, as long as the main goal is to understand and clarify structural features of a process, then this view can still be useful.

### 26.3.2.2 Business Process as a Complex Dynamic System

In this view, a process can be understood as a system that dynamically adapts to a changing environment. From a systems perspective, a process has inputs, transformational capabilities, outputs, and a boundary to its environment. A process can also be hierarchically decomposed into subsystems (people, tasks, organizational structure, and technology) and elements (e.g., individual entities of each subsystem). Subsystems and elements interact with each other (internal relationships) and with the system's environment (external relationships) to achieve some superordinate objective (i.e., the system's objective). As opposed to the mechanistic perspective, this view emphasizes dynamic behavior of processes and interactions with an external environment (cf. Melão and Pidd, 2000). In particular, instead of analyzing the efficiency of its parts, this view encourages process managers to analyze process behavior holistically. Consequently, goodness of a process is more likely measured in terms of effectiveness (e.g., by employing quality and service level measures).

Besides the mechanistic view, this view on business processes as dynamic complex system is the predominant view today in both BPM research and practice. Many techniques and tools have been developed within the past decades to analyze and control complex process behavior. For example, TQM and techniques of statistical process control like Six Sigma have been developed and are applied under this particular view. Moreover, complex dynamic interactions within a process can be modeled and analyzed by means of discrete-event simulation. Many information systems (so-called process-aware information systems, cf. Dumas et al., 2005) keep track of user activities in so-called event logs enabling the analysis of process executions (PEs) through business activity monitoring (zur Muehlen and Shapiro, 2010) and also the discovery of “hidden” or formerly unknown process structures through process mining (van der Aalst, 2011). Furthermore, execution semantics of process modeling languages are defined in this view (see detailed discussion in this chapter on the business process modeling notation (BPMN) language in the subsequent sections). By means of such semantics, it can be analyzed if a given process model is “sound,” that is, if it is able to be executed and if there is a danger of deadlocks in a process. Formalisms like Petri nets allow for mathematically proving process soundness. Many tools that support process modeling have a built-in functionality to animate the flow of flow units through a process and to automatically check process soundness.

In the past years, BPM research has widened its focus from solely analyzing process structures toward understanding interrelationships between process structures and other subsystems. For example, a research stream on value-oriented process modeling (see discussion in the subsequent sections and vom Brocke et al., 2010) explores the relations of process structures and the value system of an organization. Within this stream, important research contributions have been made with regard to goal-oriented process modeling linking the goal-systems of an organization with process
designs (e.g., Neiger et al., 2008). Goal-oriented process modeling can be applied to facilitate a strategic alignment of business processes with organizational goals or to support risk management activities in BPM. Another contribution in the value-oriented research stream has been made with regard to linking process modeling with the financial system of an organization (vom Brocke and Grob, 2011; vom Brocke et al., 2011) in order to analyze the impact of process design decisions on economic performance measures like the total cost of ownership, the return on investment (ROI), or the net present value (NPV).

While this complex dynamic systems view facilitates analyzing and understanding dynamic process behavior, it still shares critical assumptions with the mechanistic view. It still bears the risk to neglect the sociopolitical dimensions of a business process and it still implicitly assumes that process design and analysis can be approached in logical and rational terms (cf. Melão and Pidd, 2000).

Moreover, business processes as viewed from a complex dynamic systems perspective are assumed to have no intrinsic control, that is, the behavior of the system’s elements is controlled by exogenous input variables and not by internal system states. In particular, policies that might “regulate” a system’s behavior are not considered. Such systems are called open loop systems. Figure 26.3 illustrates an example of a complex dynamic system. The system under study is an airport, which might consist of subsystems like groups of passengers, IT resources, employees, and of course, a process system that determines the behavior of the “airport system.” In particular, Figure 26.3 considers the process of “boarding a plane” from the perspective of the airport management. Airport management is interested in analyzing process performance in terms of passenger throughput, passenger queue length, or passenger waiting times (i.e., the relevant flow unit here are passengers). Since policies for conducting a passenger boarding are not considered, the process behavior at time \( t \) is determined by the arrival rates of passengers at time \( t \) (exogenous variable), the passenger queue length at time \( t \), and the process structure (which in this case might be a sequence of a “check boarding pass” and a “complete boarding” activity). In the extreme event that all passengers arrive at the same time in front of the gate and are allowed to board the plane in the order they have queued (implicit first-in-first-out policy) process performance is likely to decline as time evolves since seat numbers among passengers are arbitrarily distributed. This might cause congestions within the airplane since passengers who have seats in the front of the plane temporarily block passengers seated in the back of the plane when trying to occupy a seat (anyone who has ever boarded a plane might have experienced such a situation). Figure 26.3 depicts the situation at three points in time. Initially, 150 passengers are waiting in front of the gate. After 3 min, 10 passengers have already found a seat while 20 passengers are waiting in front of an airplane. Additionally, 10 passengers arrive at the gate. After 10 min, already 50 passengers have found their seats. As is indicated in Figure 26.3, the number of passengers having found their seats in a particular row is randomly distributed and follows no particular pattern.

However, the existence of a policy, that is, the ability of the process to exert internal control over the arrival rate of passengers could lead to a better process performance since seats can be occupied according to particular patterns. An exemplary policy could be that “if the number of passengers in front of a gate \( G \) exceeds a threshold of \( x \) then only passengers in seat rows \([Y..Z]\) of a plane are allowed to board first.” Such a policy would increase the probability that passengers find their seats more conveniently and thus congestion (see waiting queue in front of airplane after 10 min) could be avoided. This policy would interact with the first activity (e.g., “check boarding pass”) and requires to model (information) feedback loops from “inside the airplane” activities to the boarding activity (so that the boarding activity is updated if boarding is complete for particular rows). The inability of an open loop systems view to account for the additional dynamics arising from internal system control (e.g., policies and information feedback loops) is the main limitation of this particular view. In fact, policies and feedback loops determine the behavior of many real-world processes (Melão and Pidd, 2000) and are thus worth to be considered properly. Remember that information structures have also been considered as an important aspect of the process concept in the preceding text (see Figure 26.1).
Business Process Management and Business Process Analysis

FIGURE 26.3 Exemplary process analysis in a complex dynamic systems view.
26.3.2.3 Business Process as an Interacting Feedback Loop

The view on business processes as interacting feedback loops is a refinement of the more general view on business processes as complex dynamic systems. The difference between these views lies in the nature of a system's control structure. Business processes in the feedback loop perspective can have internal control over their behavior and are thus characterized as closed loop systems. The main concepts used to represent processes as interacting feedback loops are flows of resources, stocks (for accumulating and transforming items), and causal (feedback) loops. The performance of a process is evaluated in terms of its effectiveness to achieve a set of process objectives given a set of regulating policies.

While this view is closely related to system dynamics modeling, it has been rarely applied within a BPM context (Melão and Pidd, 2000). Applications of this view could be useful in two ways: (1) as a means to qualitatively explore process structures by means of causal loop diagrams, and (2) in a quantitative way that translates the causal loop diagrams into equations, which then serve as a basis for process simulation.

The system dynamics view inherits some limitations of the more general view on business processes as complex dynamic systems. The human factor might only be considered in terms of an instrument to be controlled or exercising control (cf. Melão and Pidd, 2000). Moreover, system dynamics modeling might be difficult to apply since the “completeness” and “impact” of identified causal chains and feedback loops might be hard to prove and quantify. It is rather suggested to use system dynamics modeling in a “qualitative mode” to learn about process behavior.

26.3.2.4 Business Process as a Social Construction

This view on business processes does not assume that a process can be perceived and designed in a rational way but rather emphasizes that a business process is an abstraction, meaning, or judgment as a result of a subjective construction and sense-making of the real world. Since the people that are affected by or are involved in a process have different values, expectations, and individual objectives, business processes are perceived differently by different stakeholders. Since these different perceptions can be conflicting, process changes have to be negotiated. Quality of a process is thus evaluated in terms of cultural feasibility (cf. Melão and Pidd, 2000).

In BPM practice, this view could be applied by means of the so-called soft systems methodology (SSM) (Checkland and Scholes, 1990). SSM employs systems thinking as a means to reason about people's perspectives on a business process. That is, a process description is used as a sense-making interpretative device to generate debate and learning about how a process is or should be carried out. A business process in SSM terms is thus a “would-be purposeful human activity system consisting of a set of logically interconnected activities through which actors convert inputs into some outputs for customers” (cf. Melão and Pidd, 2000, emphasis added). The human activity system is subject to environmental constraints and could be viewed from different angels depending on individual perceptions or social constructions of a process.

There are research reports demonstrating how to apply SSM in process change projects (e.g., Cahn and Choi, 1997; Galliers and Baker, 1995). However, research has yet to fully explore the potential of SSM in a BPM context, for example, with regard to linking SSM with more technical or “hard modeling” approaches (e.g., as supported by the complex dynamic systems or information feedback loop view).

Among sociopolitical elements, culture has been reported an important factor in BPM (Harmon, 2007; Spanyi, 2003), which has been further investigated in academic BPM studies (vom Brocke and Sinnl, 2011; vom Brocke and Schmiedel, 2011). BPM culture is defined as a set of organizational values supportive for realizing BPM objectives (vom Brocke and Sinnl, 2011). The so-called CERT values have been identified: (a) customer orientation, (b) excellence, (c) responsibility, and (d) teamwork; all four values have been found to be supportive for effective and efficient business processes (Schmiedel et al., 2013). The values are incorporated into a BPM culture model, and it is suggested that BPM initiatives are more likely to succeed if the specific organizational culture incorporates elements of the BPM culture model (vom Brocke and Sinnl, 2011).
A main limitation of this view on business processes is its sole reliance on cultural feasibility, which may impede more radical process designs and changes (cf. Melão and Pidd, 2000). Moreover, while this view is capable of capturing sociopolitical issues, it does not disclose how such issues could be dealt with. More recent research in BPM, however, does seem to overcome this limitation by designing tools for measuring, for example, the cultural fitness of an organization for BPM to very early identify barriers for process change and take preventive actions (like skill development and culture development) accordingly (Schmiedel et al., 2012).

26.3.3 Integrating the Multiple Perspectives on Business Processes

As can be seen from the aforementioned discussion, business processes can be viewed from several perspectives, which provide an account for the multifaceted nature of reality (cf. Melão and Pidd, 2000). While each of the discussed views has its limitations, they might well serve as a starting point for pluralistic approaches to BPM. For example, if a process under study is not well understood and bears potentially problematic implications for a group of people, then soft approaches can be used initially to gain consensus about what the process under study process looks like. A static approach as suggested by a mechanistic view on business processes can then be employed to document and communicate a process structure. In addition to such temporal combinations of the views, different process context can also call for specific process views to be predominantly applied. The mechanistic view on processes, for instance, is useful for analyzing and improving technical, well-defined processes, while the interacting feedback loop view or the social construction view might be more appropriate to study less structured business areas. Davenport (2005), for instance, discusses how the management of knowledge work can benefit from a process-oriented view. Also, the primary objective of process management can have an influence on the right view: If the analysis and improvement of process behavior is of interest, then the process analysis can be complemented by an approach that employs systems thinking. In particular, if (unpredictable) complex interactions within a process are not well understood, then approaches to disclose the process dynamics might prove to be useful, like process simulation, process mining, or business activity monitoring. Finally, if a business process is assumed to have feedback loops or some intrinsic control structures, then the systems view on processes should be extended to also consider the dynamics of policies.

Except for the social construction view on business processes, all views considerably assume that processes can be objectively perceived and process performance is rather independent of the people who participate within a process. Taken to an extreme, these views would suggest that processes can be optimized (even automatically) based on some quantifiable attributes. For complex and less technical processes, this assumption, however, has proven problematic already during the reengineering wave. Many re-engineering projects failed due to an overemphasis on designing technically sound processes at the cost of fostering social, cultural, and political resistance to change (Willcocks and Smith, 1995). Thus, a technically feasible and sophisticated process is a necessary but by no means a sufficient condition for the success of process change projects. Contemporary BPM therefore needs to follow a more holistic picture particularly integrating both technical and social elements of organizations as socio-technical systems (Bostrom and Heinen, 1977). BPM is, therefore, considered an integrated management discipline (Rosemann and vom Brocke, 2010) that seeks to build organizational capabilities to continuously improve and innovate business processes according to their very specific nature and given objectives and context factors.

For successful BPM initiatives, the problem, therefore, is not to favor a particular view on business processes. Instead, the contrary is the case. In order to capture the full richness and complexity of a situation, business process managers benefit from approaching the analysis and improvement of business processes in a multifaceted way (cf. Davenport and Perez-Guardo, 1999). Each view has its merits and limitations with regard to the analysis and improvement objectives to be achieved in a particular situation. Table 26.2 summarizes the different views on business processes as discussed earlier with its assumptions, limitations, and possible areas of application.
26.3.4 Process Analysis in the Context of a BPM Life Cycle

Having clarified the multifaceted nature of business processes, we now briefly discuss how BPM activities are affected by particular views and why process analysis is a significant task in an overall BPM initiative. Figure 26.4 depicts a BPM life cycle, which sketches the relationships between the activities to be conducted while managing business processes.

The BPM cycle starts with designing, documenting, and implementing a process. As opposed to the assumptions of the reengineering movement, process implementations do not always require the use of IT. For example, implementations of human-driven processes would require some simple procedures or policies to be obeyed by employees or other process participants. Once a process is up and running, a process manager might be interested in how the process performs with regard to some performance measures and with regard to the some competitors. Once a process performance gap is identified, corrective actions have to be planned in order to change the current process and transform it into a process with a more desirable performance. Once the process has been redesigned, it is again subject to performance measurement, and the cycle starts over again.

What can be seen from the description and from Figure 26.4 is that process analysis is a prerequisite to almost all BPM activities. Designing and documenting a process necessitates the initial analysis of existing process structures. Identifying performance gaps necessitates an analysis of the process behavior. Overcoming a performance gap and redesigning a process necessitates an understanding of the current process behavior and also an understanding of the to-be process as well as of the potentials to be unlocked by a new process design (cf. vom Brocke et al., 2009). Process analysis is thus central for gaining transparency about processes (in terms of structure and behavior).

In this regard, process modeling languages constitute an important methodological basis for BPM. Examples of such modeling languages are the event-driven process chain (EPC) or the UML activity diagrams. Selecting the “right” modeling language and the “right” view on a process in a given situation is becoming more and more difficult as BPM initiatives become more complex. This chapter provides a brief overview of some more widely used modeling languages and then introduces one specific modeling language that meanwhile has become an international industry standard: the BPMN. We employ a
complex dynamic systems view on business processes. We focus on the capturing of process structures under this view and the analysis of a particular kind of process behavior: economic value creation.

This particular type of behavior addresses an important question in BPM, which has so far received little attention. Process managers frequently have to choose a process (design) alternative (out of the large number of alternatives available in total) by judging which alternative appears to be the most beneficial one. Thus, this question is critical to the successful management of business processes. This chapter, therefore, particularly takes a look at this important dimension and presents methods that allow their users to compare and determine the economic value of different process reorganization alternatives. The methods for creating structural process transparency and process value transparency as presented in this chapter build upon each other in order to achieve maximum synergies between the modeling and the economic assessment of processes. Furthermore, the chapter demonstrates how useful financial key indicators, such as total cost of (process) ownership or ROI (of process redesign), can be calculated. Finally, the chapter presents a real-world example and briefly discusses its results.

26.4 Creating Transparency about Process Structures

26.4.1 Process Modeling Languages: A Brief Overview

In order to be able to create transparency about process structures, process modeling languages are used, which allow to describe the sequence of activities (in terms of time and logical order) within a process. As far as the development and use of such process modeling languages is concerned, a trade-off has to be made regarding the formalism to be employed. On one extreme, processes can be described in natural language (text-based modeling). On the other extreme, formal modeling languages can be used that restrict the use of the “modeling vocabulary” (modeling based on modeling grammars, cf. Recker et al., 2011). Natural language-based modeling languages are quite easy to understand. However, they are characterized by a large degree of freedom regarding the representation of process structures, which may be a reason not to use them for certain analyses as they might be too inexact. Formal modeling languages, on the other hand, do support such analyses, yet they are understood by relatively few experts only. The use of methods for process management therefore is closely connected with methods for semi-formal specification of business processes.

For example, the EPC (Keller et al., 1992) is a frequently applied process modeling language and is used for customizing the SAP® enterprise resource planning system. Nowadays, due to the considerable number of available modeling formalisms, process modeling languages are subject to standardization initiatives, for example, the UML (OMG-UML, 2011) or the BPMN (OMG-BPMN, 2011, as of Version 2.0 called business process modeling and notation). Especially the development of BPMN has been influenced a lot by some related and already widely used modeling languages (cf. OMG-BPMN, 2011, p. 1). Besides the EPC and UML activity diagrams, Integrated DEFinition for Process Description Capture Method (IDEF3; cf. Mayer et al., 1995) and ebXML Business Process Specification Schema (ebXML BPSS, cf. OASIS, 2006) have been considered throughout development of the BPMN. In addition, the BPMN execution semantics explicitly builds upon the formally defined semantics of Petri nets (Petri, 1962), which in turn is based on the concepts of “tokens” (see explanations further in the subsequent sections). Apart from BPMN, also EPC and UML activity diagrams each incorporate the Petri nets semantics.

The process modeling languages described in Table 26.3 address different modeling purposes. EPC and IDEF3 are primarily used for specifying the functional requirements of business processes, mainly for designing organizational structures (in order to be able to meet requirements regarding documentation, certification, benchmarking, or process-oriented reorganization) (cf. Becker et al., 2011). UML activity diagrams and Petri nets, on the other hand, are predominantly used for designing process-oriented application systems. Some Petri net variants are used for modeling workflows (i.e., processes of
## TABLE 26.3  Overview of Process Modeling Languages

<table>
<thead>
<tr>
<th>Modeling Language</th>
<th>Short Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EPC</strong> (Keller et al., 1992)</td>
<td>EPC is a semi-formal modeling language for describing processes. EPC represents processes as <em>sequences of events and functions</em>. The EPC does not distinguish between start events, intermediate events, and end events. The architecture of integrated information systems (ARIS) (Scheer, 2000) uses EPCs for describing the <em>process/control view</em>, which integrates the four other ARIS views (<em>organization, function, data,</em> and <em>output</em>). The EPC is particularly suited for specifying the functional requirements of business processes (i.e., the modeling of business processes from a business perspective).</td>
</tr>
<tr>
<td><strong>IDEF3</strong> (Mayer et al., 1995)</td>
<td>Similar to BPMN and EPC, IDEF3 was developed with the intention to be easily learned and used by practitioners. IDEF3 is domain neutral and primarily supports the documentation of <em>process knowledge</em> (Mayer et al., 1995). IDEF3 process models are predominantly used for <em>process analysis</em> aiming at process re-organization. IDEF3 supports two strategies of documentation (<em>process centered</em> and <em>object centered</em>), for which two views (<em>schematics</em>) are provided (<em>process schematics</em> and <em>object schematics</em>). An IDEF3 process description may comprise one or more process and object schematics.</td>
</tr>
<tr>
<td><strong>UML activity diagrams</strong> (OMG-UML, 2011)</td>
<td>UML activity diagrams represent one a particular UML diagram type. They are used for describing the behavior of <em>application systems</em>. Activity diagrams illustrate the dynamic aspects of single use cases, showing start and end events but not intermediate events. Although activity diagrams just like BPMN 2.0 are based on <em>Petri net semantics</em>, the execution semantics of the two modeling languages differ widely with regard to certain aspects. For example, a process described by an activity diagram ends when at least one final node has been marked with a token (see the discussion on the token concept in the subsequent text). In contrast, a BPMN process remains active as long as there are any tokens in a process. When specifying the functional requirements of business processes, practitioners often combine activity diagrams with other UML diagram types in order to be able to separately illustrate functional requirements and technical requirements in a <em>use case diagram</em> or to describe relations between business relevant process object types in <em>class diagrams</em>.</td>
</tr>
<tr>
<td><strong>Petri nets</strong> (Petri, 1962)</td>
<td>Petri nets come with a simple graphical process modeling language that can be used both for describing the behavior of application systems and for modeling business processes. Petri nets are based on a concept of <em>tokens, places</em> (i.e., <em>events</em>), and <em>transitions</em> (i.e., <em>functions/activities</em>). Petri nets do not allow the expression of different views of a business process. By means of the token concept, Petri nets provide an <em>instrument for abstraction</em> that allows to describe and <em>formally analyze the semantics of control flows</em>. A lot of the more recent modeling languages use Petri nets semantics, which allows analyzing processes with regard to cycles, deadlocks, or inaccessible paths.</td>
</tr>
<tr>
<td><strong>BPEL</strong> (OASIS, 2007)</td>
<td>The OASIS is standard “business process execution language for web services (BPEL4WS or BPEL)” is an <em>XML-based, machine-readable</em> process description language. A BPEL process can be understood as a <em>web service</em> that interacts with other web services. BPEL can be used to describe both <em>inner-organizational</em> and <em>cross-organizational processes</em>. The BPMN defines <em>rules</em> for transforming graphical BPMN process models into BPEL models.</td>
</tr>
<tr>
<td><strong>ebXML BPSS</strong> (OASIS, 2006)</td>
<td>The ebXML business process specification schema (BPSS) is an <em>XML standard</em> maintained by OASIS that allows describing processes in a textual, <em>machine-readable</em> way. ebXML-based process descriptions specify which business partners, roles, collaborations, choreographies, and documents are involved in the execution of a process. The focus is on describing <em>cross-organizational exchange of business documents</em>, i.e., the modeling of cross-organizational processes.</td>
</tr>
</tbody>
</table>
fine granularity) where transitions of activities are controlled by application systems (a process engine). Process descriptions used as input for such application systems are thus required to be machine readable. Such descriptions can be provided by non-graphical, XML-based process modeling languages (such as BPEL or ebXML BPSS), which allow to generate technical, machine-readable process descriptions that can be interpreted and executed by a process engine. The graphical representation of process models is not a primary aspect of these process modeling languages. However, there are a few languages that allow to seamlessly map graphical process descriptions onto technical, machine-readable process descriptions in order to close the gap between process modeling from a business perspective (functional requirements of a business process) and the actual PE. Sample notations that aim at linking a business perspective with a technical perspective on business processes are the BPMN (OMG-BPMN, 2011), the activity and sequence diagrams of the UML (OMG-UML, 2011), or yet another workflow language (ter Hofstede et al., 2010).

In recent years, BPMN has developed into an industry standard for process modeling. BPMN aims to provide a notation for graphical description of business processes that allows to consistently connect both process design and process implementation (cf. OMG-BPMN, 2011, p. 1). BPMN process descriptions are supposed to be understood by both technical developers and business analysts. Furthermore, the BPMN aims at adequately visualizing technical, XML-based process descriptions in order to be interpretable by human actors. In general, BPMN is supposed to facilitate the end-to-end modeling of processes, starting with process descriptions from a functional perspective, which are then more and more refined until a detailed BPMN model is developed from which a technical process description (e.g., in BPEL) can be generated.

Version 1.0 of BPMN was launched in 2004. It was developed by the Business Process Management Initiative Notation Working Group, which then was merged with the Object Management Group (OMG) in 2005. Since then the OMG has been responsible for maintaining the BPMN. In 2011, the OMG published Version 2.0 of the BPMN, following Version 1.2. Among the new features are two additional diagram types: conversation diagrams and choreography diagrams. As of Version 2.0, it is also possible to extend BPMN with own notation elements.

Taking into account certain element types that had already been part of earlier versions of BPMN, Version 2.0 was extended primarily with elements that focus very much on the modeling of executable process models. Thus, BPMN is focusing more on the design of workflows. While there are a large number of BPMN notational elements available, in practice only a relatively small number of elements are actually required and used for the modeling and design of business processes (the so-called BPMN core, cf. zur Muehlen and Recker, 2008). The following section presents a selection of these core elements and concepts of BPMN.

### 26.4.2 Business Process Modeling and Notation

The presentation of central concepts and elements of BPMN refers to literature covering the fundamentals of BPMN (Havey, 2005; OMG-BPMN, 2011; Weske, 2007). The concepts and elements to be presented in the following sections have been selected according to their significance for getting a basic understanding of BPMN models. After providing a short overview of the BPMN diagram types, one specific diagram type, the Business Process Diagram (BPD), together with its most important notational elements is presented in detail. Subsequently, one section is dedicated to the analysis of the behavior of BPMN process models by discussing the BPMN execution semantics.

#### 26.4.2.1 BPMN Diagram Types

As has already been mentioned, the BPMN supports the modeling of sequences of activities (in terms of time and logical order). Prior to Version 2.0, the BPMN considered one diagram type only: the BPD. Unlike other process modeling languages (such as EPC or IDEF3), the BPMN did not provide separate views on process models (e.g., value chain view, organizational structures, or data view).
As of Version 2.0, however, the BPMN offers the possibility to model a “global view” of the sequence of interactions taking place between several collaboration partners (i.e., of inter-organizational processes). Therefore, the BPMN has been extended with two additional diagram types: conversation diagrams, which specify conversations taking place in an interaction scenario (who is involved, what messages are exchanged), and choreography diagrams, which specify the course of a conversation. Both diagram types establish an external, cross-organizational view on the interplay between collaboration partners. The details of affected internal processes of conversation participants are then specified in BPDs.

For multi-perspective process modeling, the BPD has to be extended with additional diagram types, which are not part of BPMN (such as UML class diagrams for specifying complex organizational and data structures). Because of the central role of the BPD, its most important elements are presented in the following section.

26.4.2.2 BPD: Core Elements

The notational elements associated with the BPD represent so-called object types used to describe business processes (cf. OMG-BPMN, 2011, p. 27 ff.). The BPMN makes a distinction between flow objects, connecting objects, swimlanes, artifacts, and data objects (see Figure 26.5). The BPMN comprises much more elements than shown in Figure 26.5. The elements presented here are considered to be core elements of BPMN and the ones that are most frequently used in BPMN-based process modeling projects (cf. zur Muehlen and Recker, 2008).

Flow objects are used to describe the sequence of activities (tasks). Basically, the BPMN distinguishes atomic activities and activities that can be refined into sub-processes. That way it is possible to describe hierarchical process structures. Besides the tasks, events and gateways represent central notational elements of the BPMN. Events can initiate a process (start events), occur as external events during PE, be generated internally as a result of the execution of a task (intermediate events), or terminate a process instance (end events). When events occur, a process engine (i.e., an application system that coordinates and monitors the execution of the process) saves process-related data. The more a process description focuses on the technical implementation of a process, the more important it is to actually model events. Process modeling from a functional perspective often focuses on start events and end events only, including events related to state changes of relevant process objects (e.g., sending or requesting messages, creating or modifying documents). Besides simple, un-typed events, the BPMN provides multiple event types, of which message events are frequently used in process modeling. Gateways, finally, represent a third core notational element. They are used to illustrate decision points within a process.

Flow objects can be related to each other by means of connecting objects. There are three types of connecting objects provided by the BPMN. Sequence flows connect tasks with each other and specify the order in which they are supposed to be executed. Message flows connect tasks contained in different pools (see Figure 26.6) in order to be able to model a communication channel for exchanging message objects. Associations are used to relate flow objects to artifacts in order to be able to integrate additional information into the process description (e.g., to add information about input and output conditions for single tasks).

Swimlanes enable the specification of responsibilities regarding the execution of tasks. In particular, swimlanes allow the description of basic organizational structures. The BPMN divides swimlanes into pools and lanes. Self-contained processes are modeled within pools. Lanes are used to group tasks according to responsibilities for their execution.

Artifacts are used to integrate context specific information into a process model. The most frequently used artifact type is annotation. As of BPMN Version 2.0, it is possible to integrate additional custom notational elements (i.e., which are not defined by the BPMN standard) as artifacts in order to be able to adapt the BPMN to individual process modeling requirements (cf. OMG-BPMN, 2011, p. 8). Artifacts can be connected with flow objects and sequence flows.

Data objects represent all information and documents used or produced within a process. Unlike the EPC, for example, the BPMN does not provide the possibility to represent descriptions of
Flow objects

Events
Signifies state changes that happen or may happen in the course of a process execution. A distinction is made between start, intermediate, and end events.

Tasks
Signifies something that needs to be done in the course of a process execution.

Gateways
Signifies decision points in the course of a process execution.

Connecting objects

Sequence flow
Signifies the sequential order of activities.

Message flow
Signifies the flow of messages between activities or pools.

Association (directed/undirected)
Signifies a relation between an artifact and a flow object.

Swimlanes

Pool and lanes represent responsibilities with regard to the execution of individual activities.

Pool typically aggregate all activities executed by one organization.

Lanes are used to further specify responsibilities.

Artifacts

Task
Collapsed subprocess
Loop
Ad hoc
Transaction
Multiple instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

Tasks

gated
Collapsed
subprocess
Loop
Ad hoc
Transaction
Multiple
instances
Compensation

Data-based
XOR
Inclusive
OR
AND

Untyped start, intermediate, and end events
Receive message events
Send message events
Termination: immediate termination of process instance

complex data structures within a process model or a separate view. Data objects are connected with flow objects via associations.

Figure 26.6 shows an example of an abstract BPMN process model. The graphic shows two internal processes (one for "Organization X" and one for "Organization Y"), each of which is embedded within a separate pool. The depicted situation typically applies in inter-organizational settings (such as, when a credit card check service is used prior to booking a flight). Processes contained in separate pools are connected with each other by message flows only (so-called choreography). In the example in Figure 26.6, such a choreography is initiated by the internal process of "Organization X." After "Business Unit 1" has executed the first activity (Task 1), the process continues in "Business Unit 2" (Task 2), which sends a message to "Organization Y." Upon receipt of this message, a process is triggered in "Organization Y." Following an activity (Task 1) in "Organization Y," a new message is generated and sent back to "Organization X." Upon receipt of this message, a process is triggered in "Organization Y." Following an activity (Task 1) in "Organization Y," a new message is generated and sent back to "Organization X." Meanwhile, the process of "Organization X" waits until the message from "Organization Y" has been received. When the message has been received by "Organization X," a gateway decides which one of two alternative activities is triggered and executed in "Organization X," depending on predefined conditions (i.e., business rules).

### 26.4.2.3 BPMN Process Model Execution Semantics

The BPMN execution semantics specifies how a process behaves at the time of its execution (cf. OMG-BPMN, 2011, p. 425 ff.). A basic understanding of the execution semantics of BPMN process descriptions is useful to prevent process modelers from designing processes that do not behave as intended. Without considering execution semantics process models that are syntactically correct may still show unexpected behavior at runtime. For example, processes may be blocked (deadlocks), process paths could not be executed (inaccessible paths), or process areas repeatedly executed accidentally (cycles). Such malfunction of processes should be taken into account already during the phase of process modeling. For this reason, the BPMN 2.0 standard explicitly defines execution semantics of BPMN processes. These semantics have been derived from the Petri nets semantics and is based on the concept of tokens.

The token concept allows validating process structures in both the modeling phase as well as during PEs at runtime. Tokens are assumed to flow through a process instance along the control...
FIGURE 26.7 Examples of BPMN execution semantics. (a) Synchronization; (b) No Synchronization; (c) Synchronization (despite OR join); (d) Turnaround time?; (e) Earliest ending?

Interpretation of process structures:
(a) Task 1 and Task 2 are executed before Task 3 starts
(b) Task 3 is executed twice
(c) Like in (a)
(d) The turnaround time is 25 min
(e) The process always ends after 25 min
flow structures. In this sense, tokens can be interpreted as an abstraction of instances of single flow units that are expected to flow through a process (e.g., order “XYZ” that is processed). During PE, tokens can be multiplied, consumed, or created. As long as there is at least one token within a process instance, the associated process is considered being active. In order to validate the runtime behavior of a process, the token flow can be simulated. In the following, it is explained in detail how the token flow influences the execution of a process through a process engine.

Upon occurrence of a start event, a process engine creates a token with a unique identifier. This identifier signifies a particular process instance. If a start event is subsequently triggered, then a new process instance is created together with a new token and token identifier. Beginning at the start node, the token then starts to move forward through the process following the control flow specified by the process model. Tasks are executed only if they have been activated, that is, if a token has arrived on each incoming control flow edge and if each token refers to the same process instance. When a task is executed, all incoming tokens are consumed. If a task has been executed successfully, a new token is put on each outgoing control flow edge. In case of XOR-nodes, a token is put on one and only one outgoing control flow edge. In case of AND-nodes (parallel execution), tokens are put on each outgoing control flow edge (tokens are cloned or multiplied). Unlike the EPC, for example, the BPMN does not provide specific rules or patterns as to how control flows are to be split and merged again. BPMN only provides rules specifying the behavior of mergers of control flows during PE. If a control flow is merged by an AND-node, subsequent tasks and events are not triggered and executed until tokens referring to the same process instance have arrived at all incoming edges of an AND-node. The control flow thereby gets synchronized (see Figure 26.7a), that is, one has to wait until all tasks of all incoming control flows have been successfully executed (see Figure 26.7d). If a parallel control flow is merged by a XOR-node (see Figure 26.7b), no synchronization takes place. As soon as one token has arrived at the XOR-node, it is passed on to activate subsequent tasks or events. If a control flow is split by an AND-node and merged by an outage risk (OR)-node (see Figure 26.7c), BPMN stipulates that the OR-merge behaves just like an AND-merge.

A BPMN process is active until all tokens of a process instance are consumed by an end node (i.e., there are no tokens left in the process any more, see Figure 26.7e). When a token has arrived at the “termination” node (see Figure 26.5), all tokens of a process instance are deleted instantaneously, and the process is immediately brought to an end.

After having discussed how process behavior can be analyzed in terms of process control flow structures and the flow of units within these structures, the next section aims at exploring how the economic consequences of a particular process design can be evaluated. This evaluation is still very often neglected in BPM practice due to a predominant focus on the technical implementation of specified process structures. Neglecting the analysis of economic impacts implied by a certain process design, however, may cause a technically sound process implementation to not deliver a satisfying process performance in terms of economic performance measures.

### 26.5 Creating Transparency about the Economic Value of a Process

#### 26.5.1 Principles of Value-Oriented Process Modeling

The BPMN is a modeling language which is suited for modeling business processes from both a business-oriented and a technically oriented perspective. In doing so, BPMN basically facilitates the specification of structural and logical relations within a process. What BPMN does not take into account, however, are the economic consequences implied by a particular process design. This is tolerable as long as these consequences can be roughly overseen and are intuitively understood by decision makers.

However, assessing intuitively the economic consequences of design decisions made during the process modeling phase has become more and more difficult in recent years. Due to the sheer number of possible design options, it is hardly possible to reliably judge on an intuitive basis which process...
alternative is superior to another and under what conditions this is the case. Because of that, it is important to explicitly analyze the economic consequences (i.e., economic value contribution) of design alternatives already at the time of process modeling (vom Brocke, 2007). In particular, decision makers are often interested in the financial performance resulting from the execution of business processes. When taking this aspect into consideration, it becomes obvious that transparency about process structures alone is not sufficient. What is needed in addition is transparency about the economic value created or consumed by a process. Thus, the process of business process modeling needs to be complemented by an accompanying process of calculating the value contribution of a process (vom Brocke, 2007) enabling a continuous and integrated assessment of the economic consequences pertinent to different business process alternatives (vom Brocke et al., 2009).

The analysis of economic consequences can be approached from different perspectives (cf. vom Brocke et al., 2010). A reasonable approach seems to consider economic value in terms of the degree of fulfillment of specific economic goals. The question to be answered is what economic goals are considered to be relevant in each individual case and how different goal dimensions need to be weighed. Stakeholder theory (as originally proposed by Freeman, 1984) provides a conceptual framework here to balance interests and goals of different groups that have a vested interest in a process or an organization. The problem of balancing different interests and assessing different goal dimensions properly has been reflected in various controlling (e.g., the balanced scorecard; cf. Kaplan and Norton, 1992). Process management frequently refers to different goal dimensions, such as cost, quality, and time (the magic triangle). More recent studies have extended this view beyond economic goals, integrating other goal dimensions in order to be able to assess also the ecological and social effects of the execution or reorganization of processes (Hailemariam and vom Brocke, 2010).

The present chapter aims to illustrate how economic consequences of process designs can be assessed by means of methods for capital budgeting. We chose to refer to this specific perspective since it is considered to be of particular importance in a capitalist economy. This perspective is also important since benefits of BPM initiatives are ultimately judged by means financial performance measures (cf. vom Brocke et al., 2010). Moreover, the approach to be presented may serve as a pattern for developing other assessment perspectives. A crucial prerequisite for the presented approach is the existence of process models. The overall goal of the approach is to overcome the methodological divide—that still exists both in theory and in practice—between process modeling and process controlling and evaluation in order to be able to improve both the efficiency and the quality of decisions taken in BPM and process controlling.

### 26.5.2 Constructs for Value-Oriented Process Modeling

In many practical cases, the question arises whether reorganizing a certain process pays off. Two aspects are decisive here: the benefits to be expected from a process reorganization and its price. Both aspects should always be assessed as early as possible in order to be able to ponder different design alternatives and make decisions that are as rational as possible. A lot of studies on process controlling (also often referred to as process monitoring or business activity monitoring) have proposed to use key performance indicators for controlling a process’ value contribution at process run-time (for an overview on this see zur Muehlen and Shapiro, 2010). This proposal, however, is of limited use for assessing planned processes as opposed to evaluating running or finished processes. Studies have shown that the potential profitability of processes is significantly determined already during the phase of process modeling and less during the run-time phase of a process (cf. vom Brocke et al., 2010). What is remarkable though is that hardly any methods for such process evaluation during the phase of process design have been developed so far. Figure 26.8 shows a calculation scheme that can be applied to assess the economic value implied by a particular process design. This calculation scheme assumes that economic value is analyzed in monetary terms and follows a capital budgeting approach. The approach is explained in detail in the subsequent text.

The benefit of a process reorganization or a process (re-)design can be ascertained by comparing the total payments of process ownership (TPPO) for the case with reorganization (process p) and the case
without reorganization (process \( p' = \text{status quo} \)) (level 1 and level 0 in Figure 26.8). The benefit resulting from a process reorganization is expressed by a positive difference between the TPPO\(_{p'}\) of the new process \( p' \) and the TPPO\(_{p}\) of the same process at the status quo level (process \( p \)). The price of a process reorganization is the sum of the payments that need to be made for the transformation of the process (level 1). These payments are referred to as total payments of process transformation (TPPT) in the following. TPPT typically comprise payments for procurement of new IT, for the development of process knowledge, or for training employees affected by a (new) process design.

In order to be able to measure the TPPO and the TPPT, long-term economic consequences of the process reorganization need to be taken into account, which is why the planning horizon for the payments needs to span a substantial period of time (e.g., 5 years). By netting the series of payments (TPPO\(_{p'} - \text{TPPO}_{p}\)) and the investment in the process transformation (TPPT), the total expected payments resulting from the process reorganization can be calculated (level 2). This sequence of direct payments provides the basis for taking into accounting further financial consequences, including
indirect (derived) payments (such as interest and tax payments). To calculate the derived payments, various standard methods for investment controlling can be applied. Apart from applying classical methods for capital budgeting (such as the NPV or the internal rate of return), the method of visualization of financial implications (Grob, 1993) provides a suitable means to make the financial consequences of a particular investment transparent. It allows to make transparent how direct and indirect payments that are related by different financial conditions lead to a particular economic performance measure. This transparency has already proved to be very beneficial for process evaluation purposes (vom Brocke and Grob, 2011) as process design decisions could be directly related to expected payments and specific conditions of funding.

Assessing the TPPO on level 0 and on level 1 can be made at different degrees of detail. The financial planning approach allows both to specify in advance net payments to be expected for each planning period and to further differentiate the calculation of net payments without explicitly linking the payments to a particular process structure. Another interesting approach uses BPDs (see Section 26.3) as a basis for the assessment. To do so, BPDs must be complemented with information about economic consequences, as shown in Figure 26.9 with the BPMN.
Figure 26.9 shows an example of how the payments of a process design alternative can be captured and consolidated through an activity-oriented approach. Out-payments are calculated based on the use or consumption of input objects and resources objects. All payments calculated for each activity (or task) can then be aggregated according to the process structure and extrapolated over several planning periods. This extrapolation of payments can be done either individually or by applying trend rates (cf. vom Brocke and Grob, 2011; vom Brocke et al., 2010).

To calculate the price (TPPT) of a process reorganization, certain calculation templates can be used for putting together typical payments with representational values for certain transformation tasks. It is important here as well to make an assessment that is as complete as possible, with due regard to both time and logical order. An example is given in Figure 26.10.

The approach for calculating the economic value in synch with process modeling, as presented earlier, is subsequently illustrated by means of a real-world example. The example has been taken from (vom Brocke et al., 2009) and used to demonstrate how transparency about both process structured and economic consequences implied by a process can be gained based on BPMN process descriptions.

### 26.6 Application Example

A medium-sized logistics company uses a web-based enterprise portal to support its business processes. The company management is thinking of conducting a process reorganization project with the aim to integrate the route planning process into its portal. In route planning, two kinds of planning types are distinguished: detailed planning and adhoc planning. In order to be able to determine the planning type for each delivery, order prioritization policies have been defined. Delivery orders of high priority are subject to detailed planning. If there is not enough time available for detailed planning, however, adhoc planning is applied instead. The problem with adhoc planning is that resulting routes may turn out to be inefficient as the delivery may not be made in time (leading to contractual penalties), and the truck fleet may not be deployed efficiently.

The fact that route planning has always been done manually by this company and that manual planning is a very time-consuming effort, has led to a drastic overrepresentation of adhoc route plans, also for delivery orders of high priority. By integrating the route planning process into the enterprise portal the company hopes to reduce errors, meet delivery schedules, and achieve better deployment of resources. The technical implementation of the solution is to be done on the basis of
a service-oriented architecture. Prior to the implementation, different design alternatives need to be assessed that particularly impact the effectiveness and efficiency of the planning process:

1. **GlobalRoutePlanning**: An IT solution by means of which route plans can be created over an online interface and saved to the company’s database. Using the service requires specific information, such as delivery orders, truck fleet capacity, order prioritization, delivery addresses, and delivery dates. Using this solution, the process of route planning is fully “out-tasked.”

2. **GeoDataForLogistics**: An in-house solution by means of which internal routing rules and customer data get enriched by external route information provided by a special geographic map service particularly suited to the needs of logistics companies. While this service is able to substantially reduce the planning effort, it also requires the development of a number of data services (wrappers) in-house.

3. **IntelligentRouting**: A web service by means of which fully fledged route plans can be created (similar to **GlobalRoutePlanning**, but only for a particular geographic region). As the geographical data of this service is very much up to date (providing information on construction sites or blocked roads, for example), the planning quality is very likely to be significantly improved by this solution.

The internal route planning process is represented by means of a BPMN process model (see Figure 26.11). In order to be able to depict process alternatives and specify the quantity structure, company-specific notational elements (BPMN artifacts) have been integrated (see explanation of symbols in Figure 26.11). Calculation of payments is basically done according to the scheme illustrated in Figure 26.9. Two resource object types are relevant here: organizational unit (here: dispatching/scheduling) and the services to be integrated. The quantity structure relevant to the calculation of the use of resources is specified within the BPD by means of tables (proprietary artifact).

The process design alternatives are considered implicitly in Figure 26.11. In order to be able to integrate the services into the process, various infrastructure requirements are needed (e.g., purchase and maintenance of an enterprise service bus (ESB), implementation of interfaces, and in-house developments). Deciding in favor or against a certain service in this specific example is expected to have a local impact only, as there are no structural or institutional interdependencies with other process elements.

The BPD illustrates the design alternatives as they were given in the concrete case. Selection of a to-be model here is made on the basis of a comparison of alternatives taking into account financial key indicators. This calculation is shown in Figure 26.12 and illustrated in more detail for the case of the **IntelligentRouting** web service.

As the impact of the design decisions is only local, the calculation of the differences between alternatives can be made by means of a partial analysis for determining the direct payments. When calculating the difference, it has to be determined what the expected additional payments—compared to the status quo level (as-is)—of each design alternative are (to-be).

Using the service for the route planning process promises both a higher number of detailed plans potentially possible and a better quality of realized detailed plans (rDP). It is expected that the processing time of the activity “Create detailed plan” is reduced from 10 to 5 min. It is also expected that the error ratio is reduced (from 20% to 3% in case of the **IntelligentRouting** web service). Taking into account an available capacity of 1,825 working hours per period, a number of PEs of 18,250, and an OR of the web service of 3%, direct cost savings of 22,356 € compared to the status quo level can be expected in period 1 (calculation is based on the assumption of an average advantage of 2.50 € for creating a detailed plan compared to creating an ad-hoc plan).

For the **IntelligentRouting** web service, a transaction-based pricing model is assumed, with an average calculation rate of 0.25 € per transaction. The payments per period to be calculated on a service level are to be calculated on the basis of the execution frequency expected for the task
Potential for process reorganization
Increased capacity and thus increased probability to create a significantly higher number of detailed plans.

FIGURE 26.11 Business process diagram for process (route planning).
## Business Process Management and Business Process Analysis

### Direct payments

<table>
<thead>
<tr>
<th>Period</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>...</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- IntelligentRouting</td>
<td>-25.000 €</td>
<td>11.431 €</td>
<td>12.422 €</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>- Activity level</td>
<td>22.356 €</td>
<td>24.569 €</td>
<td>...</td>
<td>29.614 €</td>
</tr>
<tr>
<td></td>
<td>- Savings</td>
<td>22.356 €</td>
<td>24.569 €</td>
<td>...</td>
<td>29.614 €</td>
</tr>
<tr>
<td></td>
<td>(1) CA</td>
<td>1.825</td>
<td>1.825</td>
<td>1.825</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2) PE</td>
<td>18.250</td>
<td>19.163</td>
<td>22.183</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(3) (PT_{as-is})</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) (PT_{to-be})</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(5) (ER_{as-is})</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(6) (ER_{to-be})</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(7) (OR_{to-be})</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(8) (pDP_{as-is}) = (\frac{CA}{PT_{as-is}})</td>
<td>10.950</td>
<td>10.950</td>
<td>...</td>
<td>10.950</td>
</tr>
<tr>
<td></td>
<td>(9) (pDP_{to-be}) = (\frac{CA}{PT_{to-be}} \cdot (1–OR))</td>
<td>21.243</td>
<td>21.243</td>
<td>21.243</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(10) (rDP_{as-is}) = MIN ({2} ; {8} \cdot {1–(5)})</td>
<td>8.670</td>
<td>8.670</td>
<td>8.670</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(11) (rDP_{to-be}) = MIN ({2} ; {9} \cdot {1–(6)})</td>
<td>17.703</td>
<td>18.588</td>
<td>20.606</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(12) (\Delta rDP = (10)–(11))</td>
<td>8.943</td>
<td>9.828</td>
<td>11.846</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Savings per detailed plan (€)</td>
<td>2.50</td>
<td>2.50</td>
<td>2.50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= Total savings (€)</td>
<td>22.356</td>
<td>24.569</td>
<td>29.614</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Infrastructure level</td>
<td>-25.000 €</td>
<td>-6.500 €</td>
<td>-7.500 €</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>- Template [payments ESB]</td>
<td>-25.000 €</td>
<td>-6.500 €</td>
<td>-7.500 €</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>- Initial expenditure</td>
<td>-25.000 €</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Maintenance</td>
<td>4.000 €</td>
<td>4.000 €</td>
<td>4.000 €</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Adaptation</td>
<td>2.500 €</td>
<td>3.500 €</td>
<td>4.000 €</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Depreciations</td>
<td>5.000 €</td>
<td>5.000 €</td>
<td>5.000 €</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Service level</td>
<td>0 €</td>
<td>-4.426 €</td>
<td>-4.647 €</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>- Service payments</td>
<td>0 €</td>
<td>-4.426 €</td>
<td>-4.647 €</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>- Service charge rate (price per transaction)</td>
<td>0,25 €</td>
<td>0,25 €</td>
<td>0,25 €</td>
<td></td>
</tr>
<tr>
<td></td>
<td>= Service payments</td>
<td>17.702</td>
<td>18.588</td>
<td>...</td>
<td>20.606</td>
</tr>
<tr>
<td></td>
<td>- GlobalRoutePlanning</td>
<td>-1.000 €</td>
<td>3.060 €</td>
<td>5.183 €</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>+ Activity level</td>
<td>23.720 €</td>
<td>26.001 €</td>
<td>...</td>
<td>32.839 €</td>
</tr>
<tr>
<td></td>
<td>+ Infrastructure level</td>
<td>-1.000 €</td>
<td>-1.500 €</td>
<td>-700 €</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>+ Service level</td>
<td>-19.161 €</td>
<td>-20.119 €</td>
<td>...</td>
<td>-22.990 €</td>
</tr>
<tr>
<td></td>
<td>+ GeoDataForLogistics</td>
<td>-32.500 €</td>
<td>11.770 €</td>
<td>13.229 €</td>
<td>...</td>
</tr>
</tbody>
</table>

### Formulae for evaluating the economic potentials

- \( pDP_{as-is} = \frac{CA}{PT_{as-is}} \)
- \( mDP_{to-be} = \frac{CA}{PT_{to-be}} \cdot (1–OR) \)
- \( rDP_{as-is} = \text{MIN} \{PE ; pDP_{as-is}\} \cdot (1–ER_{as-is}) \)
- \( rDP_{to-be} = \text{MIN} \{PE ; pDP_{to-be}\} \cdot (1–ER_{to-be}) \)
- \( \Delta rDP = rDP_{to-be}–rDP_{as-is} \)

### Symbols

- \( CA \) Capacity (hours/period)
- \( PE \) Process executions (#)
- \( PT \) Processing time (min)
- \( ER \) Error ratio (%)
- \( OR \) Outage risk (%)
- \( pDP \) Possible detail plans (#)
- \( rDP \) Realized detail plans (#)

**FIGURE 26.12** Calculation of direct payments of the process reorganization.
“Create detailed plan” (\(=DP_{to,b0}\)). In period 1, for example, for 17,702 detailed plans created, the service payments amount to \(-4,426\) €. If the route planning process is out-tasked, the calculation rate— at a lower OR—is 1.05 € per transaction, so that the expected payments rise accordingly (\(-19,161\) €).

Apart from the activity-based payments (\(TPPO_p\) and \(TPPO_{p'}\)), the payments for the process transformation (\(TPPT\)) also need to be taken into consideration. In this case, most of these payments primarily result from investments in the technical infrastructure. Using the IntelligentRouting web service requires the implementation of an ESB solution. In the case example, it is assumed that the company pursues an incremental implementation strategy. Therefore, a decision in favor of the IntelligentRouting web service brings about all payments for purchase of technical infrastructure (25,000 €) as well as all follow-up payments for maintenance and adaptation that occur periodically. In case GlobalRoutePlanning is used, with activities for detailed planning being out-tasked, payments for the technical infrastructure are substantially lower. If GeoDataForLogistics is used, more payments for the technical infrastructure are expected due to the relatively high implementation and development effort required.

With direct payments being consolidated by means of methods for capital budgeting, the future values of investing into the reorganization of the process are 30,379 € (IntelligentRouting), 25,424 € (GlobalRoutePlanning), and 26,235 € (GeoDataForLogistics). Compared to the future value of opportunity of 11,425 € (own equity compounded with an interest rate of 6 per cent), implementing any service can be considered beneficial to the company. The IntelligentRouting web service is the design alternative which generates the highest additional future value.

### 26.7 Conclusion and Outlook

This chapter aimed at providing an introduction into the basic concepts of BPM. It focused on phenomena pertinent to the task of business process analysis, as this task very central for BPM. While incrementally developing a multifaceted perspective on business processes and the task of business process analysis, this chapter particularly calls for integrating process modeling and process evaluation. We think the integration of process modeling with an economic analysis represents a meaningful contribution to the field of BPM. Current approaches for process modeling do not allow for assessing the economic value of process reorganization projects. Approaches for process evaluation or controlling (such as activity-based costing), on the other hand, have no differentiated understanding of process structures and take no advantage of process descriptions already available in practice.

As a consequence, we call for an integrated analysis of the structure of a process and its economic value contribution using an integrated set of methods and instruments. As both, established methods for process modeling (e.g., BPMN) and for capital budgeting are to be applied, the main challenge is to create interfaces between these two domains. How such interfaces can be realized on the basis of BPMN process descriptions has been one of the foci of this chapter. In particular, the chapter presented a procedure model to calculate the return on process transformation, which integrates both a process structure perspective as well as a process value perspective.

Applying these methods in practice has shown that it is not always necessary to calculate at a high degree of detail. Using the set of methods presented in the article allows to simplify the evaluation procedures by starting to capture process structures in their entirety on a higher level of abstraction in order to be able to conduct a kind of “pre-assessment” first. Evaluations on a higher level may take into account only a few approximate values at the beginning, which can later be refined successively according to what is deemed necessary. Following this principle, one-day or two-day workshops often are a very good instrument for developing a basic understanding of which direction to go when thinking of reorganizing a certain business process.

More generally, we see the BPM discipline further establishing into a comprehensive management discipline. While contributions over the past decades were mainly focused on modeling, designing, and implementing selected processes of interest, BPM in the future will focus much more on developing capabilities inside organizations to continuously improve and innovate their business. With this,
the view of processes as social constructions in particular will play an increasingly important role. While the relevance of the social factor in BPM is often pointed at, little contributions are yet available to facilitate the management of processes as social constructions. Evaluating the financial performance of alternative process design is one approach to facilitate communicating process innovation and process transformation to different stakeholder groups, such as shareholders, in particular. Such approaches can help to strengthen capability areas such as strategic alignment and governance. Further areas or future research include people and culture as two distinct capability areas dedicated to the human factor in BPM.

References


