FH JOANNEUM - University of Applied Sciences

Process Instance Adaptation
in Process Aware Information Systems

Master Thesis
submitted at the Master Degree Program Information Management
for the degree „Diplomingenieur für technisch-wissenschaftliche Berufe“

Author
Michael Mair, BSc

Supervisor
FH-Prof. Mag. Dr. Robert Singer

Graz, 2014

FH | JOANNEUM
University of Applied Sciences

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknow-ledgement has been made in the text.

Graz, 17. Dezember 2014

Michael Mair

Contents

List of Figures

List of Listings

Abbreviations

ACM	Adaptive Case Management
AI	Artificial Intelligence
API	Application Programming Interface
BPA	Business Process Automation
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMN	Business Process Model and Notation
ECA	Event Condition Action
EPC	Even-driven Process Chain
ERP	Enterprise Resource Planning
GUI	Graphical User Interface
IDE	Integrated Development Environment
ITSM	IT Service Management
OCR	Optical Character Recognition
PAIS	Process Aware Information System
S-BPM	Subject-oriented Business Process Management
SOA	Service Oriented Architecture
SQL	Structured Query Language
UML	Unified Modeling Language
UX	User Experience
WF	Windows Workflow Foundation
WS-BPEL	Web Services Business Process Execution Language
XaaS	Everything as a Service
XPDL	XML Process Definition Language
YAWL	Yet Another Workflow Language

Abstract

Business processes can be sophisticated if implementing various kinds exceptional paths. Instead of modeling every possible deviation of a path in a predefined process definition, exceptional paths can be added on demand. However, such flexibility requirements bring business process management systems to the limits of its capacity.

Runtime flexibility, whether triggered by process model optimizations or needed for ad hoc changes represents a key requirement in today's process aware information systems. Beside stating prerequisites for providing robust support for flexible scenarios from a technical point of view, additional features needed for the convenient usage by end users are discussed. By providing comparison between four process engines, which offer support for runtime flexibility, enables practical insides in those capabilities of real world solutions. Limitations of flexibility features for each business process solutions compared are discussed respectively.

Restrictions on flexibility features depend on the overall concept. BPMN (Business Process Model and Notation) seems to be not appropriate for the demand of flexibility in business processes. Despite the fact that there are alternative concepts available, BPMN as the de facto standard cannot be substituted that easily. Additionally, process management systems consist of several components, which make them to the complexity hiding solutions they are today. However, solely making technology mature for instant changes does not imply the convenient execution of exceptional situations. Nonetheless, it defines a fundamental requirement.

Dynamic processes already can be implemented to a certain extent. Unpredictability can be handled as well with the help of case management. Yet companies will be required to shift IT from traditional static process capabilities to highly flexible intelligent workflows.

Kurzfassung

Geschäftsprozesse mit einer Menge von implementierten Abweichungen vom Standardpfad können sehr komplex werden. Statt jede Ausnahme in einem vordefinierten Prozess abzubilden, können Abweichungen nach Bedarf hinzugefügt werden. Jedoch bringen solch dynamische Anforderungen Geschäftsprozessmanagementsysteme an ihre Grenzen.

Anpassungsfähigkeit zur Laufzeit, egal ob diese durch Optimierungen am Prozessmodell oder durch Ad-hoc-Änderungen hervorgerufen werden, bildet die Schlüsselanfordung in heutigen Informationssystemen. Neben Voraussetzungen für robuste Unterstützung für anpassungsfähige Szenarien aus technischer Sicht, werden Merkmale für die brauchbare Anwendungen durch Benutzer behandelt. Mit dem Vergleich von vier Prozess-Egines werden praktische Einsichten in die Möglichkeiten von realen Lösungen dargelegt. Einschränkungen bei der Flexibilität der verglichenen Processsystemen werden entsprechend dargestellt.

Beschränkungen von Anpassungsmöglichkeiten hängen vom übergreifenden Konzept ab. BPMN (Business Process Model and Notation) scheint nicht die passenste Möglichkeit zu sein, um Anpassungsfähigkeit bei Geschäftsprozessen durchzuführen. Trotz der Tatsache, dass alternative Konzepte verfügbar sind, bildet BPMN den De-facto-Standard. Dadurch ist es schwierig dieses oft verwendete Konzept abzuschaffen. Außerdem bestehen Geschäftsprozessmanagementsysteme aus mehreren Komponenten und machen sie zu Systemen, die die Unübersichtlichkeit der vielen technologischen Herausforderungen reduzieren. Jedoch bringt eine ausgereifte Technologie alleine noch keine komfortable Ausführung von sofortigen Änderungen mit sich. Und dennoch bildet sie die grundlegende Anforderung.

Dynamische Prozesse können zu einem bestimmten Ausmaß bereits implementiert werden. Auch Unvorhersehbarkeit kann mit entsprechenden Systemen abgebildet werden. Dennorch werden Unternehmen aufgefordert ihre IT von traditionellen Prozessen to anpassungsfähigen, intelligenten Prozessen zu verlagern.

1 Introduction

Repetitive processes are executed by systems instead of being manually performed decades ago. In order to continue with automation, more dynamic business processes are desired to be moved to automated systems. Thus, computer systems have to evolve to be capable of dealing with this dynamic aspect. Even though experts tried to include all kinds of exceptional paths in process models, the amount and variability of requirements do not allow such adaptations to be developed effectively. Since exceptions can appear in various shapes, unpredictability needs to be considered when automation needs to be implemented effectively. Gartner even predicts that »Organizations that do not move to intelligent processes in the next five years will find themselves at a disadvantage in their respective industries« [22]. According to that statement, many companies are required to prepare for the next level of process management in order to move rigid processes to intelligent procedures. Additionally, Gartner categorizes the degree of flexibility of processes in three classes (cf. Figure 4.2). Dynamic processes, ad hoc processes and intelligent processes have to be considered. Dynamic processes to a certain extent already can be implemented. Case management and Adaptive Case Management as such, provides capabilities for moving the focus from processes to data, which forms the basis for collaboratively working on highly flexible requirements where unpredictability becomes the major focus. Intelligence of processes basically combines dynamic processes and its ad hoc counterpart. Business intelligence plays a major role in supporting the system with dynamic real time data. Even though flexibility cannot be automated in such a way it can be done with traditional processes. Thus, manual tasks will be supported by providing a new user experience with intelligent suggestions based on know how of the entire company or even industry.

In order to shift processes to those flexible requirements business process platforms need to be aware of all necessary information. Additionally, systems have to meet all prerequisites to be able to change process definitions during their instances' execution. Such platforms already provide support for process instance changes. Although features for changing instances during runtime exists, there are still limitations when it comes to the degree of flexibility a process aware information system is able to provide. It can be seen, that truly there are deviations between requirements dynamic or even intelligent processes are needed to fulfill and what features systems offer. Beside technical restrictions, providers of business process solutions offer several useful tools for modeling, monitoring and execution of processes. However, focus has been laid on how to technically implement instance changes and illustrating its limitations. By just providing technical features for implementing changes of workflow instances, solutions could not leverage the most useful advantage of business process execution, its possibility of hiding complexity and focus on business operations instead of struggling with the technology underneath. Yet the question remains, if this plain technical capability is enough for certain users. Probably there is a need for providing access to flexibility to a broader field of users.

Before starting right away with examples of state of the art flexibility features, chapter 2 provides information about business processes and why automation is so dependent on computer systems. Afterwards the reason for flexibility requirements are discussed by stating pros and cons of change and a current example of what can happen when not recognizing transformations in technology and furthermore in the markets respectively. Additionally, chapter 3 deals with unpredictability and common approaches of how to compete with unpredictable processes. Chapter 4 discusses the contradiction of handling unpredictability in an automatic way. In the section assumption are made regarding usability as an important factor for executing flexibility throughout the entire organization. Moreover, issues, requirements and possibilities are mentioned in order of dealing with unpredictable situations. How processes actually can be updated during runtime is introduced in chapter 5. Different versions of business process management systems exist and four process engines are described, how instance updates can be accomplished.

2 Processes and Automation

Processes can be found everywhere, in our body, in natural surroundings, in evolution, in outer space and certainly in the order of our last book through an online retailer. Because there is such a huge amount of processes around us every day, the topic seems to be rather banal to us. Humans try to get rid of a variety of processes and that is the reason why they let others do the work for them. A great idea because those »others« are of course computers in different shapes, sizes and realizations. Indeed it is a great idea when thinking of how error prone humans are in doing repetitive tasks. Unfortunately there is more than just ordinary processes. There are different kinds of processes, which have to be handled differently. According to humans nearly every process would be automated by computer systems. But it is necessary to get more precise when it comes to those processes. Since processes of the body, the nature and outer space to some extent are already automated, the focus lays more on systems created by humans themselves. Organizations in businesses are meant to be those systems. As already mentioned there are different types of processes, which should be considered when talking about automation, which will be explained in section 2.1. Nature's processes have been automated by an exceptional organism and humans use a similar approach with the generic term Business Process Automation (BPA). In section 2.2 an explanation of automation systems will be given. One reason for automation was already stated, which was the source of error, but this is not the single purpose of automation, which will be clear in section 2.3.

2.1 Different Process Types

When people think of types of processes in the context of business process management they usually have in mind three different kinds.

- operational processes
- supporting processes
- management processes

Those three classifications are perfectly right, but if considering the automation aspect there has to be another classification. Therefore Reichert and Weber [67] classify these kinds of processes in either prespecified and repetitive processes or knowledge intensive processes. Distinction between those categories often cannot be made accurately, which is the reason for the intersecting sets in Figure 2.1. Certainly, the most valuable processes to automate are prespecified and repetitive procedures.

Repetitive processes in general are executed many times. Every time an instance gets executed, it gets executed the same way before with different data. So it is quite

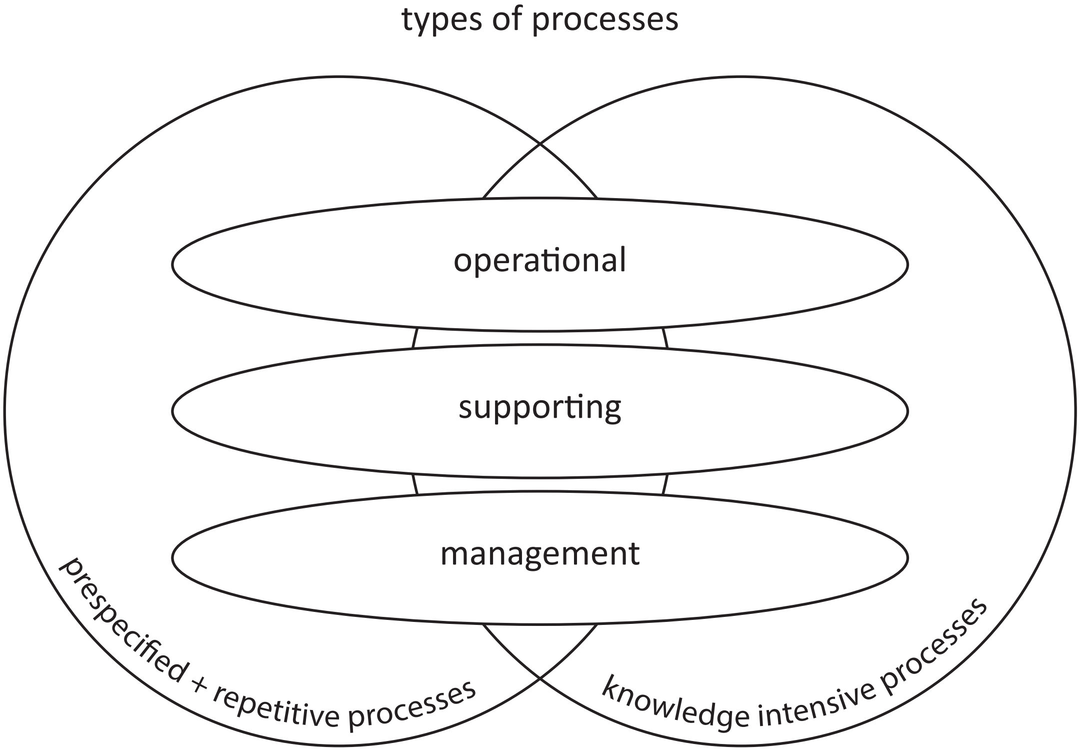

types of processes
operational
supporting
management
prespecified + repetitive processes
knowledge intensive processes

Figure 2.1: Types of Processes

obvious that computers can do the same things thousands of times with hardly any mistakes. Besides the accurate execution, the speed how fast computers can handle data is beyond dispute. The majority of a repetitive procedure's logic is known prior to its execution, which makes it easy to tell the computer what to do in advance.

Consequently, to order computers what to do in advance makes things easier but unfortunately this becomes highly sophisticated when it turns to processes, which require certain knowledge in order to get executed. Computers just know hard facts, which can be easily captured. Knowledge intensive procedures cannot provide those hard facts all the time. At this time just humans can handle such sophisticated decisions. Nevertheless there are prespecified processes in the context of knowledge intensive procedures. Such processes are defined but in a much weaker way, which is called loosely specified. There are procedures, which cannot be specified with any kind of task. In such a case tasks are that dynamic that it depends on the data they belong to, i.e. to know, which tasks to do is just known when certain data is available to the case. More about dealing with unpredictability will be covered in section 3.5.

2.2 Business Process Automation

As already mentioned automation saves time and provides accurate execution of processes. According to Gartner[1] »processes span organizational boundaries, linking together people, information flows, systems and other assets to create and deliver value to customers and constituents«, which means it is a quite complex topic where many facts have to be considered respectively. Business process automation is the automation of these kind of processes. Technology defines a big part in BPA because of the fact that a lot of automation effort depends on the underlying technology in order to successfully fulfill the automation need. Basically all the automation and self-running tasks help employees in a certain business to accomplish their work faster and more accurate. The following sections describe those magic systems supporting execution of business processes.

2.2.1 BPA Systems

In order to be able to execute processes, many different services are necessary. As can be seen in Figure 2.2 the workflow engine is the part, which actually executes process instances. Basically the term workflow is similar to the term process. Nevertheless there is a slight difference in the context of business process management. The term workflow identifies an executable process. According to [83] workflows, compared to business processes contain all necessary information needed for their automation, i.e. order of execution, general data and resources. So workflows are technical realizations of processes [41]. This is the reason why in Figure 2.2 one service is called workflow engine. In general, BPA systems exist of many different components. When starting at the very beginning of the automation ladder the process has to be defined somewhere. First a model of the process has to be defined. Since there are different types of business process languages and notations, each BPA suite has its own modeler or editor. With the modeler either the process or workflow model is generated and will be available to the workflow engine for further execution. The engine communicates with necessary services in order to execute the workflow respectively.

There are several process and workflow definition languages. BPEL, BPMN, EPC, S-BPM, UML, YAWL, XPDL are some of them whereas different vendors defined individual forks and definitions. Business Process Model and Notation (BPMN), for example, is one of the most popular definitions. There is a lack of definition for executable business processes in this kind of notation. With the plain BPMN 2.0 specification [63] it is hardly possible to create an executable process. This fact becomes clearer when having a look at the specification in the section with the title »Mapping BPMN Models to WS-BPEL«, i.e. a mapping has to be accomplished in order to turn the model in an executable process. Compared to S-BPM, there is no need for a mapping to an actual workflow definition. A model defined with S-BPM is executable on its own without any mappings or intermediate steps to perform [31]. Beside BPMN is the de facto

[1] Gartner IT Glossary, http://www.gartner.com/it-glossary/business-process-management-bpm/

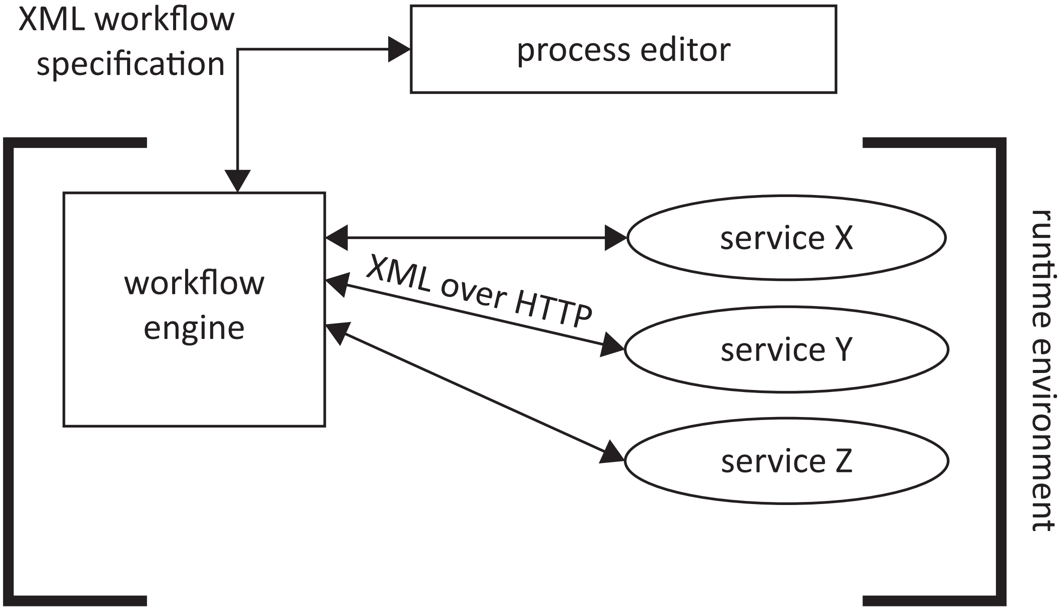


Figure 2.2: Collaborating Services of BPA (based on [2])

standard to define business processes semantically, it did not accomplish the goal of smooth implementation [18, 64]. Hence BPM software providers created their own specifications, frequently based on the BPMN standard but only suitable for their own solution.

2.2.2 BPA System Components

As pointed out in the previous section BPA systems or process aware information systems (PAIS) have to communicate with several components of the overall system infrastructure. Respectively there exists a variety of different components in such systems.

Before drawing the attention to a PAIS' components [67], depicted in Figure 2.3 there is an even more abstract view on a business process automation system. By dividing it into a build time environment and a runtime environment it makes it conscious of prespecified modeling and the afterward execution. Later on, especially in chapter 4, the problem of this kind of separation will become clearly evident.

Workflows for certain BPA systems will be built within the respective process editor. The editor helps defining, configuring and verifying an executable process model. Because problems can already occur during build time, to verify a process model saves problems later on when process instances are running. One of the most sensible problems occur when a workflow is caught in a deadlock. If this happens a process in hanging in a state without going to the upcoming or previous task. By definition a business process must not hang at a certain activity, it always has to be finished. Besides checking if deadlocks exist several other issues, e.g. syntax failures or availability of required data, can be found with the help of respective editors.

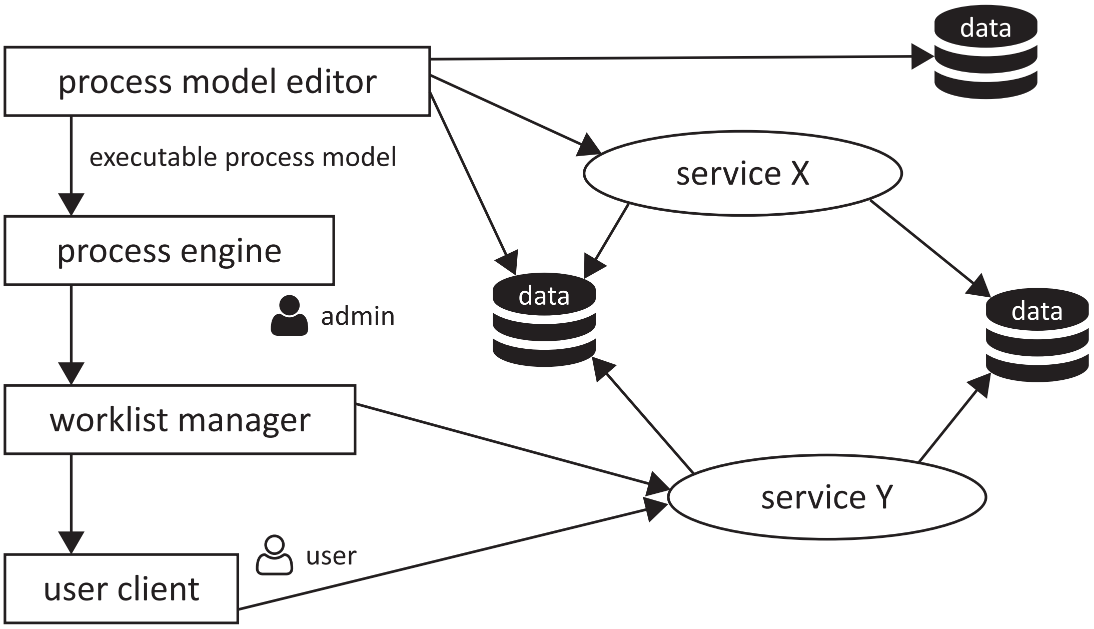


Figure 2.3: Components of a BPA System (based on [67])

In comparison to build time components, the runtime environment consists of far more components and services. After defining the workflow with the process editor, the specification can be deployed to the process engine, which is responsible for its execution. This execution depends on different steps and responsibilities, as pointed out by [67].

- instance creation of workflows
- execution of those workflow instances
- managing data created and needed by workflow instances
- orchestrating application services and sub processes referenced by activities of the process model
- invoking respective application services
- execution monitoring

To sum up the tasks of a PAIS, the system has to have an overview about all dependencies. Every step has to be completed with additional rules and constraints being considered. Even the smallest task has to be accomplished and brought to a certain state without preventing dependencies from working smoothly. The system's responsibility reaches from fully automated flows to partly automated procedures as well. If workflows require manual input and activity of a human being, the BPA system is liable for the assignment of the respective item to a chosen process actor until a certain deadline is reached. Execution has to be kept within the limits of rules, policies and constraints, which can lead to highly sophisticated dependency management.

2.3 Drivers and Enablers of BPA

It already has been pointed out why humans want to automate business processes. However, why do people need support in terms of automation was not yet discussed in detail. The thing is in business there is an ideal conception of how companies should behave. Collaboration runs smoothly, responsibility is clear, work is done fast, of course without any failures and everything happens to the utter satisfaction to the customer. In real life sometimes this is not the case in how businesses work. Honestly this would be rather boring. If nothing would change at all, Darwin would not have been able to write his theory of evolution. Stagnation in business verges on a disaster. The upcoming chapter 3 deals with change and its pros and cons.

First of all automation of business processes means that certain work will be done either completely by a machine, computer or other sort of programmable technology or is just partly automated, which means it requires human interactions. When continuing with the process classification in operational, supporting and management procedures, some of them are more difficult to automate than others. It seems quite obvious that the automation of manufacturing standardized car parts is easier than a heart surgery of an individual. The words standardized and individual already indicate the feasibility of automation today. In the end it depends on the exceptions, which can happen and its impact on the overall success of the procedure. Although it might not be possible to fully automate a heart surgery today, it does not mean humans are not able to achieve it in 50 years, if technology will be mature enough.

At the first glance the benefits of automation are obvious. Companies are always seeking for efficient ways to run the business, which immediately turns to automation. By efficiency it is meant that procedures speed up, i.e. cycle time decreases, employees save time, which reduces costs. Beside increased velocity, errors can be minimized, which makes it unnecessary to spend time on fixing issues. At the second glance there are reasons for automation that can be easily understand when thinking of large organizations with thousands of employees, the same amount of suppliers and an even higher number of customers. In order to be capable of the effort to coordinate all these actors for a smoothly running business, standardizations, simplifications, rules and constraints are necessary with an extensive focus on the communication flow.

Seamless integration of heterogeneous infrastructures and software systems between different companies would be an unbelievable effort without some sort of standardization. Large companies have to deal with globalization, which indicates collaboration with different suppliers all over the world. Supply chains have to be integrated in both the company's information system and each supplier's infrastructure as well. To stay on top of things reduction of complexity is key for maintenance and optimization reasons. It is particularly important to define collaboration among all the channels a company is interacting with. Moreover this kind of interaction strongly depends on communication, which has to be well defined as well. Unnecessary requests in terms of emails or telephone calls could disturb workers from their tasks. Therefore streamlining communications in process flows can minimize distraction of staff who is not responsible for certain requests. Obviously it makes it easier and faster to know where to refer to. In best cases employees can mainly focus on business issues they are responsible for due to simplification of work.

Beside simplified work for employees, automation brings advantages to a monitoring perspective. Because of influences, unable to control by companies, there is a focus on detecting bottlenecks and sections of importance for optimization. With automation there is the possibility of logging events and states, which makes it easy to use this data for evaluation. Based on the phrase »trust, but verify« surveillance of workflows makes it possible for companies to converge to the goal of smooth and planned business operations. Detecting weak points simplifies either to proactively handle shortages or react to issues respectively. Due to certain governmental regulations organizations often are required to support transparency of processes. The government wants to know who to blame for certain tasks, which makes it simple, thanks to automated traceability and specified responsibilities.

As already pointed out, the focus of automation mainly lays on highly repetitive processes because those are the procedures executed quite often and can be defined prior to their execution in many cases. In such workflows proportion of the effort to automate the process and the impact on business such an automation has, pays off. Some processes, which have not yet been automated probably are not valuable enough to be done by a computer because the effort of configuration and requirements on technology is not given. Basically automated processes support knowledge workers in doing their job. To be able to focus on one task makes it more efficient to work on. No distractions causing errors and a clean dashboard let them handle issue after issue.

Because of this concentration on work, those kinds of employees often are not able and not required to deal with technically related questions about process automation in some complex looking process editors. If business users are used to the syntax of, for example, BPMN they are able to understand and probably define processes in this notation. When it goes down to automation there is the need of specifying data types and dealing with further kinds of technical questions. Additionally to their main profession of handling certain business cases and the bunch of different signs in BPMN it would not be that easy to quickly adjust an automated procedure to their needs.

Although S-BPM seems to be easier when it comes to the number of symbols and the syntax, the question remains if knowledge workers are required to take care of such tasks. Maybe there will be supporting technologies, which makes it easier to create an automated process for business people. Maybe there will be no need for business people to handle those kinds of questions and information systems will be automatically handle such issues for them. Thanks to change we will see what will happen.

3 Boon and Bane of Change

From a semantic point of view change is neutral, although some people might not agree with that fact. It is no surprise that for some humans, alteration is negatively afflicted. Since people are creatures of habits it is quite obvious that adjustment in most cases does not create a pleasant feeling. In fact change is necessary but does not go hand in hand with an individual's feelings and objectives. History exhibits that change is needed (cf. evolution of the species) in order to move forward and to survive in whatsoever way. And the fact remains for businesses as well that adaptation of behavior to requirements is particularly needed. Why requirements are changing will be discussed in section 3.2.

3.1 Pros and Cons of Change

Every upside has its downside and so it is with change. This is not just related to business operations in particular. Since it is well known that not the strongest will survive but the most adaptable, it is necessary even if it is uncomfortable. As an example Nokia was the world's leading manufacturer of mobile phones until there was a change in how phones are handled. In this case Nokia did not know a change was going on because of new inventions from competitors like Apple, who until then, was no competitor of Nokia.

Nokia stagnated although they brought several new devices to the mobile phone market. Even though they revealed new and sometimes quite weird looking products all of the products had hardware buttons in comparison to the iPhone. As already pointed out, stagnation is terrible for a company and especially when Apple is going to change and expand their product range to new markets.

For Apple, the rearrangement of their product range was a new opportunity for the computer organization. Nevertheless they were willing to take risks with their new invention. It can be seen today change paid off for Apple and helped them to growth. Managing new and unknown things requires willingness to learn how to handle new situations and scenarios. Acquiring new skills means to grow, which is one of the main objectives companies are heading to. Additionally to new skills coming with handling change there is even more an organization is able to learn. According to [11] »organizations that excel at change have a competitive advantage«, which means to be the most flexible company, the one who can adapt to changes at a faster pace than its competitors, will have significant advantages. Enterprises who are required to change often know how to handle the process and will probably adapt to be as flexible as possible to react to alterations optimally. But continuous change has also its downsides. Today process automation, for instance, needs to be consistent to a certain extent.

Nevertheless to be as flexible as possible has another advantage. As everyone may know, changes are not always the best way. Changes can go wrong when heading into the wrong direction and commitment is not given. Particularly in such cases again it is good to be flexible in order to escape the negative change and turn things into better positions. Often it is not obvious when there is a demand for a change. In case of Nokia, they simply did not see that there was an upcoming change, which indeed was quite bad for the company.

There is no secret that change is hard to achieve, which often goes hand in hand with the previously mentioned commitment. Commitment of the right people is necessary in order to successfully implement new processes. If there is no commitment successful change probably is hard to achieve. If change has to be forced due to indispensable drivers, people in an organization understand why change is needed. Whereas change, initiated by strategic opinions and decisions made by not fully accepted chief officers, cannot be understood by the staff. Especially in this case commitment of the right informal leaders is key. The latter is often unpredictable because employees do not know what will happen to them. Several books address this kind of question and provide methods how to handle such attempts.

3.2 Reasons for Change

An organization is defined by its members, i.e. employees. If a company has to change it always affects people within this firm. In case of Nokia there was a strong competitor who pushed into the market with innovative technology. Drivers for change can be caused by different reasons [19]. Demand for change can either come from the market or can be a response to failure. It can be a need for organizational progression compared with new opportunities in technology, services and markets. To be concrete, Apple saw the new opportunities in touch displays and triggered a change in the mobile phone industry by revolutionizing the way phones were controlled by consumers.

Every business process has to be improved [44] continuously because of alterations in different areas. According to [37] it is quite easy to explain why processes are always changing so relentlessly. On the one hand there are needs for change because of a company's bad times. The organization is short on money and therefore has to strive after efficiency of how work is getting done. But this is not the only reason why process optimization is needed. On the other hand there are the good times where money is available for investment. Processes need to be more productive and new market possibilities emerge. Apparently there is a reason why the Deming Cycle is so famous. Continuous improvement, be it in economically good times or in economically bad times, is ubiquitous because of the aim to be better, steadily. Beside continuous improvement there is the need for collaboration among different systems. This can happen because of new suppliers or company take-overs and merges. In the latter case there is the necessity to decide, which processes of, which organization will be taken.

According to [29] change is happening faster nowadays, especially in the IT industry. There are four reasons affecting BPM in companies.

- social collaboration
- mass customization
- consumerization
- XaaS (Everything as a Service)

Sharing opinions with employees and suppliers increases information of how things can be done better. There is also the fact that consumers share their opinion about usages with the world in social networks, which adds another huge amount of data in order to detect where changes and improvements are necessary.
People want products with at least an unique touch although mass production is much cheaper. In order to give customers the possibility to stand out from the mass, companies provide services for customers to individualize their products.
Some firms react to their employees knowledge and property of their own IT devices. Therefore companies make the most of this by reorganizing their IT support with possibilities for the staff with projects like »bring your own device«.
Additionally IT services will be outsourced in order to save money and outsourcing responsibilities, i.e. there is no application management needed when just needing an ERP system thanks to XaaS.

3.3 Business Process Evolution

Change is omnipresent and consequently also business processes are affected by the alteration caused by their environment. Like organizations, processes develop further needs i.e. there is an endless improvement going on, which is called an evolution [67]. Hypothetically, even if there was no change at all evolution would still go on because technical errors, design errors or quality issues could occur, which force the process to be adapted respectively. In contrast to these internal drivers for optimization there are drivers for their counterpart as well (cf. Figure 3.1). Due to this adaptation business processes have to expose flexibility to a certain extent.

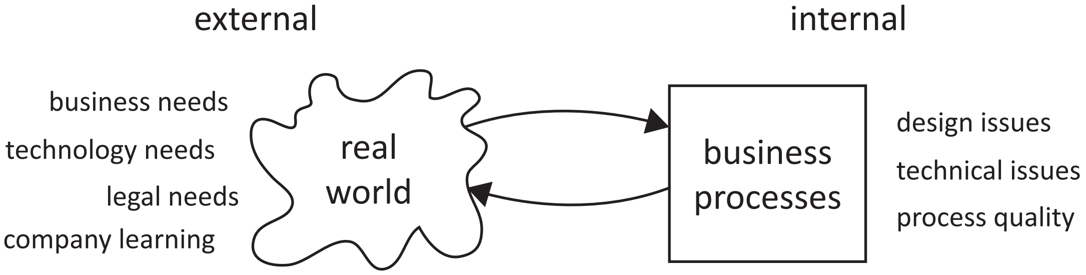


Figure 3.1: Internal and external Drivers for Process Evolution (based on [67])

Reichert and Weber characterize flexibility in [67] in the context of processes with a taxonomy as can be seen in Figure 3.2. Beside looseness, variability and adaptation the fourth major flexibility need is evolution. The first child of evolution in the taxonomy addresses drivers, which have already been discussed. Secondly, with extent it is meant to, which intensity and what impact a change has on the process. Whereas incremental optimizations have only little impact due to small changes, reengineering approaches deal with radical alterations of the business process. By classifying evolutionary adjustments by a temporary point of view, makes it discussable how long those changes are valid. On the one hand such adjustments can be valid as long as until it is overwritten by another adaptation. On the other hand there is the possibility for changes to last just for a specified amount of time, which is defined beforehand (cf. section 4.6 for a more technical illustration). The last child on the evolution branch is behavior, which is divided into an observable process behavior and the internal structure of the system. For example, moving activities in a process or adding new ones changes behavior of the process whereas architectural refactorings, e.g. separating large processes into smaller ones, does not affect the overall behavior but makes a difference to the internal structure.

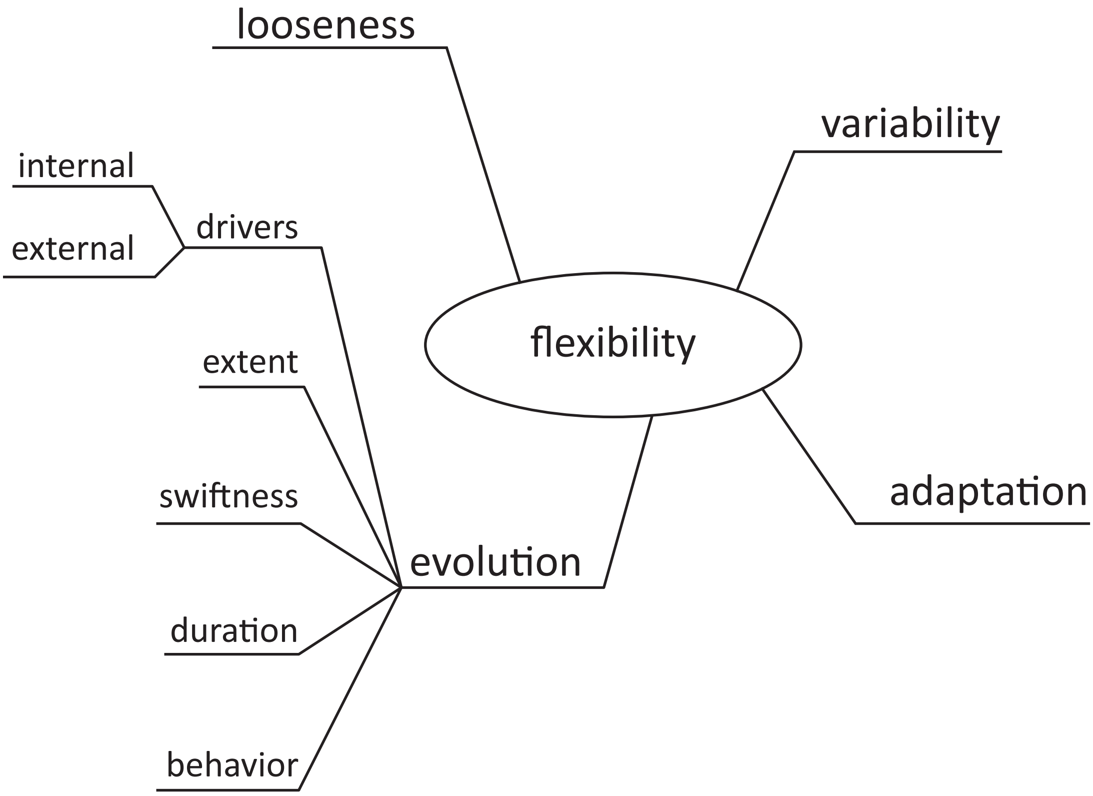


Figure 3.2: Taxonomy of Flexibility in the Context of Processes (based on [67])

In case of swiftness the consequences on dependencies of process optimizations have to be considered. Process evolution can either affect the process model, which can be seen as the blue print for a business process or a particular process instance. An instance in this case is a particular realization of the blue print, i.e. the process model (cf. Figure 3.3).

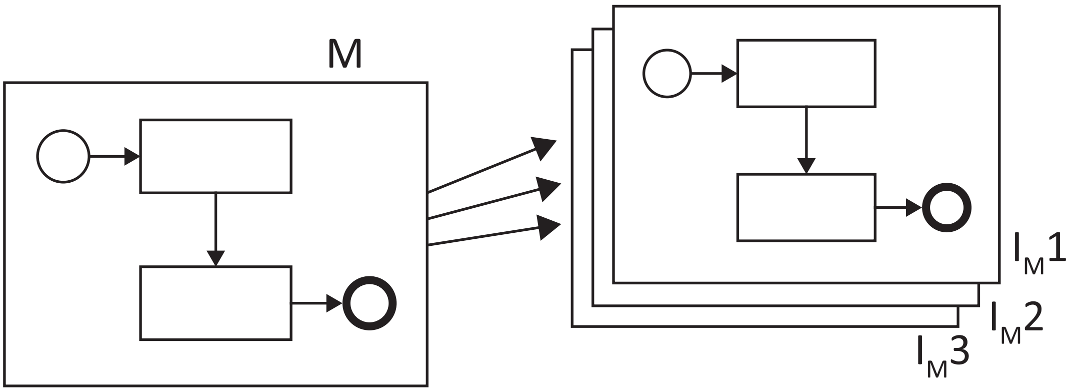


Figure 3.3: Visualization of each Instance I derived from Model M

In the scenario of process evolution, each instance of a certain model is related to it. Optimizations of the model for example can, but do not need to, take effect in its instances respectively. This does not mean that only the model can be updated, also one particular instance can be optimized (cf. sections 4.6.2 and 4.6.3), as illustrated in Figure 3.4. One consequence regarding an update on the model level concerns its instances. Already running instances may be updated during execution and others may finish execution with the started version of its definition. The latter would apply the most recent version to new created instances.

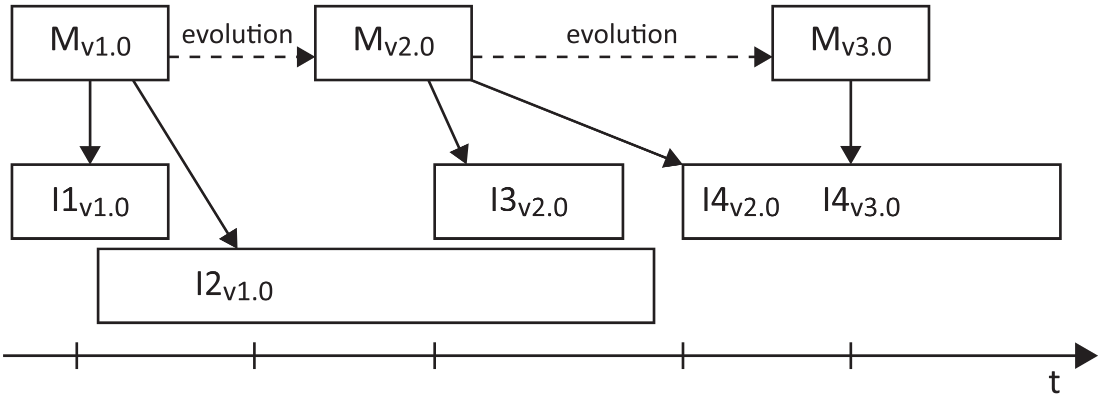


Figure 3.4: Process Evolution of Model M and its Instances I

Regarding updates of instances during runtime, is the main subject of this work. Practically, concrete examples of dealing with runtime updates will be provided in section 5. Theoretically speaking, process models and instances in particular have to meet some requirements (cf. section 4.4) in order to make sure processes are executable, especially if runtime updates are provided. Before process models can be released to be executable they have to be verified, represented by three main points [67].

- syntactical correctness
- correctness of data flow
- consistent instance states

According to the actual process modeling language used, a model has to be syntactically correct. When it comes to activity flows, however syntactical correctness is not the only thing in order to be sure a process model is save to be executed without any errors. Hence there are checks for a model's behavior, which extend the validation of a model's syntax [90]. In case of a model's behavior there are three points, which have to be considered additionally to its syntax. First, a workflow has to be able to be completed once it is started. Second, after its completion all of a workflow's activities are completed as well. Third, there are no dead activities, i.e. activities, which cannot be accessed because of the activity flow or respective conditions.

The second requirement a process has to fulfill concerns the data flow. Data flow errors occur due to missing data, unnecessary data and lost data. If a certain workflow branch produces data, which is needed by subsequent activities, this kind of branch, more specifically the activity, has to be accessed otherwise there is no data, which can be consumed by this subsequent task. In case of unnecessary data, for instance, data is produced by a task, which afterwards is not needed by any activity, condition or process environment's system. Lost data refers to overwritten data objects, which makes it useless in most cases. Especially when updating process instances during runtime, the states of the instances have to be consistent throughout the entire instance life cycle. Due to the progress of an instance sometimes it is not possible to update a workflow during runtime. Instance states are very critical and can cause severe problems when updating workflows without taking care of their states.

3.4 Deviation of Reality

In general, process models represent a certain snippet of reality. A model will never be a hundred per cent representation of this reality. It just can try to approach as close to a reality's definition as possible. As a result there is a deviation of reality to its representation of a business process. Figure 3.5 provides a presumption of the dependency between the size of a business process and its corresponding deviation to reality. The more complex a workflow gets, the higher is the deviation of reality. Theoretically speaking, a very simple workflow consisting of a single activity would have a very low deviation to its reality. There hardly is potential for deviation especially when assuming the process task is a very atomic one. Even though deviation was very low, it would never be zero because there will not be a hundred per cent replica of reality.

Now to be more concrete, deviation can be seen easily by executing instances of a certain process model. This divergence is the cause for all the exceptions happening and are not already implemented in the prespecified process model. In some cases it would be possible to implement a lot of potential exception into a certain model. Even though in a lot of workflows it would not be sufficient at all and would be an overkill of requirements, testing and validation effort to take action.

Rather to implement exceptions into the process model, there are different ways to deal with it when they occur. Knowing about what type of exception can pop up or

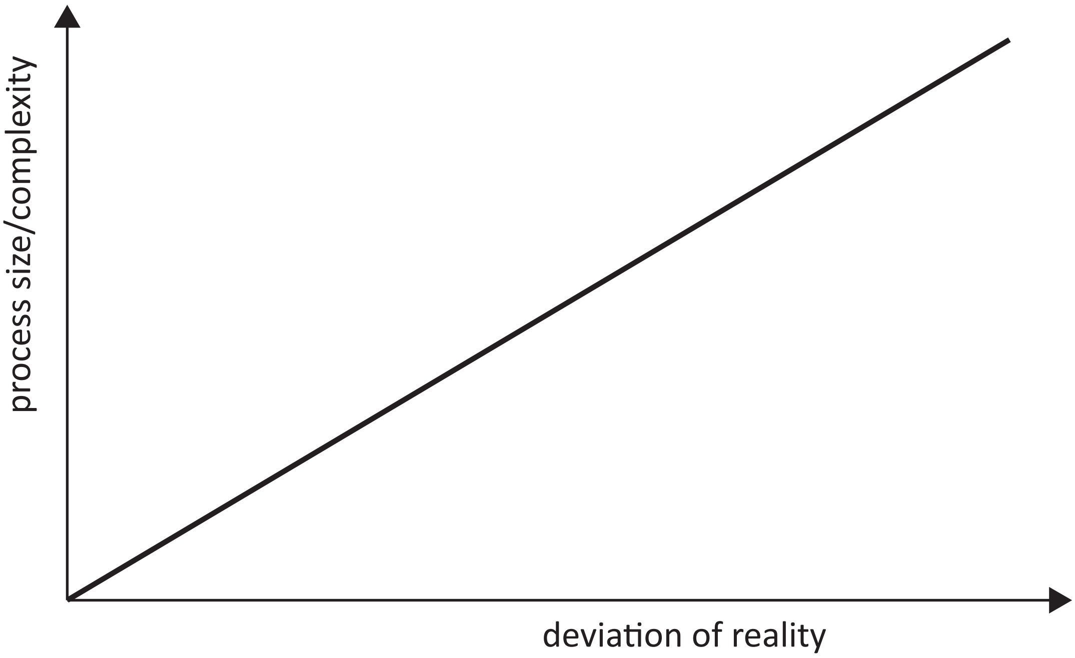


Figure 3.5: Dependence between Process Size and the resulting Deviation of Reality

without a clue when and where an occurrence is taking place are difficult to handle. Predictability and its counterpart is highly related to this kind of question. How to deal with it will be covered in the following section 3.5. In order to handle exceptions there are some existing approaches, which are able to deal with it.

- exception handling with rules
- dynamic exception handling
- recording exceptional paths
- on demand exception handling

This short list provides just an idea how to deal with exceptions in a technology point of view. Thus, the enumeration is incomplete and each item represents just the means of different handling methods, i.e. exception handling with rules can be implemented in different ways.

3.4.1 Exception Handling with Rules

Exceptional situations during execution of business processes in general can be handled by predefined triggers. When a certain exception is thrown it triggers a function, which deals with the exception respectively. This can be seen in every programming language when so called try catch blocks are assigned to certain code sections. In [46] Kim, Choi and Park propose proactive exception handling based on a rule language. They try to prevent exceptions by predicting information and covering the entire scope of possible exceptions. Predictions where exceptions might occur are based on analysis of various data sources like historical data. By definition of rules expectations can be made if it is likely that exceptions occur for instance.

3.4.2 Dynamic Exception Handling

Similar to rule based exception handling, dynamic exception handling is based on context information [52]. Certain solutions for handling exceptions are stored in a database and can be assigned to situations respectively. An ontology helps to evaluate, which available solutions are suitable for a particular case. Theoretically every exception, which already occurred and has been persisted into the solution repository can be handled dynamically. Based on this solution there is no need to change a certain business process model at design time. Moreover in most cases it would not be necessary to change an instance in order to get rid of arbitrary exceptional occurrences. However the solution repository has to be fed somehow.

3.4.3 Recording exceptional Paths

By recording the exception handling process of a particular instance the main goal is to store the exceptional path. This approach was introduced in [33, 70] where its main focus is on enabling a possibility to make static process models capable of exceptional paths. Since the introduction of the approach is made with S-BPM, human interactions are involved. First, humans are responsible for handling exceptions, which are not depicted in the original model. During the execution of the exceptional path through human interaction the individual process is recorded, which is similar to dynamic exception handling in section 3.4.2. Main idea is to collaboratively enhance respective process models in order to include repetitive occurring exceptions. Each process' behavior is represented by its executing subject (see [31] for further information). A human, who is executing a process instance represents one subject. Since certain subjects have to communicate with others, an exceptional path might traverse several subjects' behaviors. Therefore an occurring exception will be handled with the help of several subjects collaboratively working on the issue and recording the executed path.

3.4.4 On Demand Exception Handling

Exception handling on demand is a very similar approach to the exceptional path recording. However, as stated in [29] it basically focuses on decision making at runtime. Furthermore recommendations and examples go strongly into XaaS, i.e. to focus on the parts of software, which are really needed. In this case BPM can be done through a cloud based service where collaboration between process actors can be easily achieved. Decision making at runtime will be realized through collaboration between multiple process members where exceptions can be handled on demand and flexibility is given.

Every type of exception handling has its pros and cons, depending on the kind of process the method is dealing with. Since unpredictability is one of the major issues with exception handling there is a completely different approach to handle hardly predictable processes at runtime.

3.5 Dealing with Unpredictability

As already pointed out in previous sections, repetitive process are predictable and can be quite accurately modeled in advance. The opposite is true to knowledge work, which is unique in many different aspects. Keith D. Swenson points out that »processes that depend upon knowledge and at the same time produce knowledge have a compound dynamic that makes them especially difficult to predict« [82]. In order to model processes, predictability is key to develop a blue print for a workflow. Since not all work is predictable, knowledge essential in order to handle unpredictability and in almost any cases people are managing unforeseen things. There is no sharp cut between the classification of predictable and unpredictable processes, which often results in a combination of both. Aside from that, unpredictability is always a matter of detail. Basically, there is no need to forecast every detail, e.g. it does not matter to predict which data in particular will be injected into a workflow whereas it indeed matters what kind of data will be put into a process. Even highly repetitive processes consist of parts, which cannot be predicted to a certain level of detail. Therefore different approaches (cf. section 4.2) are provided in order to get rid of unforeseen situations.

3.5.1 Ad hoc Changes

When exceptions and unforeseen actions are need, ad hoc changes may be the most natural choice according to a human. Thus there are possibilities in different BPM technologies to handle such adaptations. In most cases people are executing ad hoc processes.

S-BPM provides approaches to define such ad hoc changes at runtime [70]. By offering model behavior extensions or exceptions for a subject's internal behavior it provides the possibility to leave the given path and execute ad hoc tasks. This would help to overcome flexibility issues in predefined process model to a certain extent.

3.5.2 Process Templates and Patterns

Defining process models based on templates is very common. It provides flexibility for adding, manipulating and deleting parts or even the whole template. Often it is easier to start with an already existing approach than starting from scratch. In the BPMN language approaches for ad hoc exception handling correlate with templates to some extent. BPMN does not provide an as ad hoc possibility as one might assume, however it provides ad hoc usage of existing templates [63]. Beside templates, patterns can be used to enable flexibility like an approach provided by [16]. It basically consists of a base process without any exceptional paths, which makes the workflow definition quite simple. Rules then are responsible for providing information which adaptations are required. Additionally patterns describe how adaptation can be achieved.

3.5.3 Adaptive Case Management

Adaptive Case Management (ACM) was invented due to the need for handling highly dynamic processes. Actually handling of this kind of processes just has been enhanced with technology in order to provide a tool for people, making it easier and faster to deal with business requests like monitoring and documentation. People working with ACM have the ability to focus on the actual case and are not required to write certain reports, taking notes and doing other organizational stuff. All of this meta data will be tracked by the system itself. So this implies of course that ACM cannot fully automate highly dynamic processes, at least not the core process, but it gives humans the possibility to focus on the main difficulty of the core process by automating the little supporting processes within a certain case.

3.5.3.1 Concept of ACM

In literature authors often criticize that ordinary business process management is just considering the process itself as the point, where information is organized around. Instead ACM's orientation is based on data, which is referred to as a case. That is the reason why adaptive case management focuses on unpredictable work, e.g. knowledge work because of its loosely definition of the flow of work. Many ACM systems provide definition of template fragments such as documents, tasks and other useful predefined specifications [82].

Actually ACM is just another process definition in order to handle processes. Indeed it is a very very abstract specification.

> »If all knowledge workers do is click buttons, couldn't we automate their work? The key is to know which buttons to click [82].«

This quote of Dana Khoyi states the real reason why ACM is needed as an intermediate layer in business operations.

Furthermore it implies the character of knowledge workers. A process definition could be imagined like illustrated in Figure 3.6. The meta process utilizes the already mentioned ad hoc ability of BPMN in order to describe the non sequential tasks performed during a running case. Of course this process might not be complete, it provides just an idea what the concept of ACM is. Ironically it is specified in BPMN, predefined in a process, which describes another process dealing with unpredictability.
There might be a bunch of possibilities for triggers, which can start the process described here. Basically the core of the flow illustrated consists of ad hoc process activities where several ad hoc processes may be running simultaneously. Additionally several ad hoc activities will be executed, which is indicated by BPMN's loop sign. Since

the ad hoc feature of BPMN can only deal with predefined activities, some of them have already been added to the ad hoc sub process. Beside of handling the logical case structure with possibilities to whether perform an ad hoc activity on an existing case or to create a new one, the repository represents the data store where data of the content management module of an ACM system stores its resources or at least the path to it. Furthermore the signal sign on the ad hoc sub process indicates logging activity in order to monitoring activities and providing transparency.

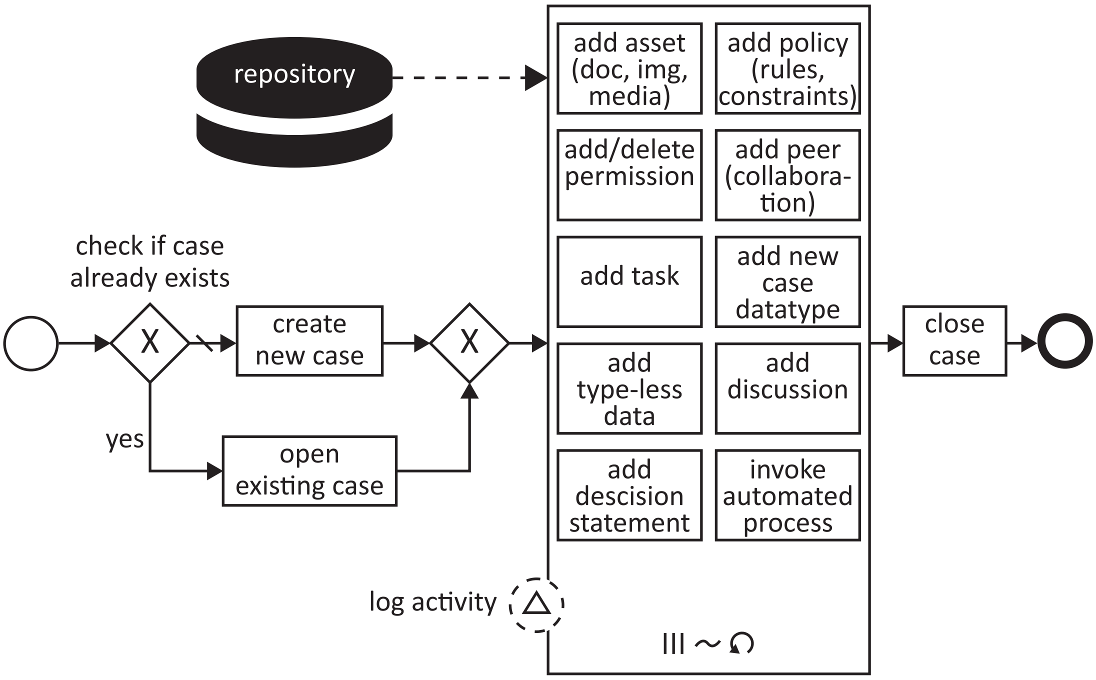


Figure 3.6: Meta Process of ACM illustrated with BPMN

3.5.3.2 Advantages of ACM

In a presentation of PayDox Case Management[1] [78] Victor Senkevich illustrates advantages ACM provides to businesses. Furthermore it is worth to mention that PayDox offers an online demonstration of their ACM system on their web site.

- store knowledge available for all enterprise members
- template creation based on historical cases
- avoidance of irreplaceable employees
- meta data for controlling and measuring of business cases

Of course these advantages make sense but the main question is how does all of the different systems are working together in order to create a smooth solution for dealing with business cases of any type.

[1] PayDox Case Management, http://www.paydox.com/

3.5.3.3 Potential behind ACM

As pointed out by Dana Khoyi in [82] including BPM functionality in ACM systems could lead to a solution, which can be used to deal with real world business issues in an inexpensive and manageable way. Basically this means that adaptive case management would act as an intermediate layer between highly automated business processes and the knowledge worker, illustrated in Figure 3.7. Cases handled by humans just trigger certain automated processes and enables possibilities in order to interact with certain processes. Business cases without predefined, partly automated processes, can be handled and tracked by the system in a manual way. Subsequently recorded information can be used in order to evaluate and implement workflows to simplify work for people and enhance the overall process.

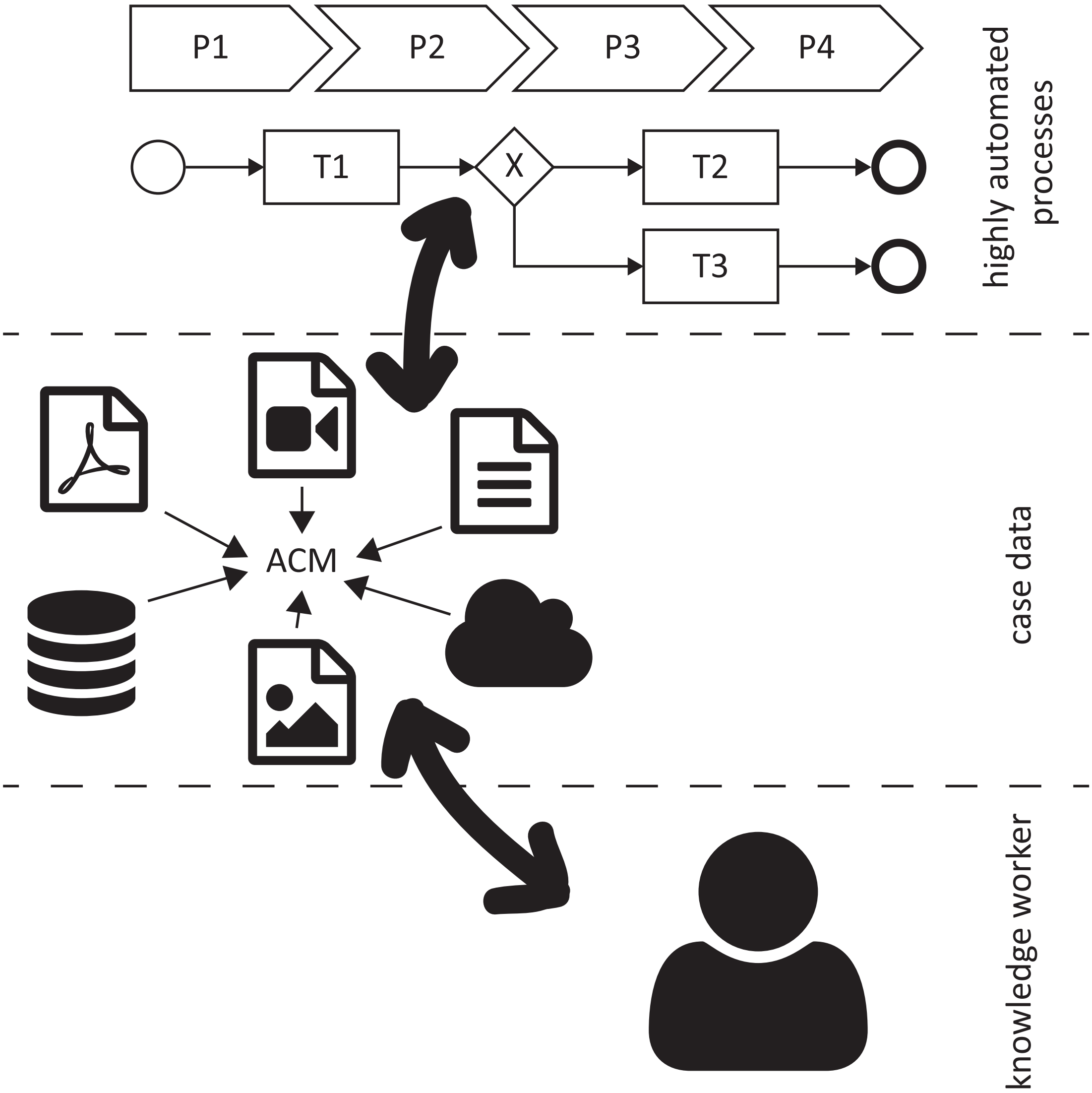


Figure 3.7: ACM as the intermediate Layer in Business Solutions

3.5.3.4 Case Management and its critical capabilities

As already mentioned, a BPM suite that just offers support for fully automated processes lacks for the flexibility aspect and would result in bypassing the system. Since analysis on executed processes (cf. section 3.5.4) yield valuable information, bypassing the system would loose such data. Therefore case management in general and ACM in particular need to be part of the entire solution in order to properly handle any kind of business requirements. Gartner sees the importance of case management frameworks as well and revealed their view on the framework in a research note [43]. According to them, a holistic solution, which provides support for highly standardized and automated processes as well as supporting highly flexible processes reduces the need for custom applications and thus for custom coding. Of course such a solution has to be that flexible, that even in reality there is no need for bypassing the system, which theoretically makes it easier to extend behavior on demand. Basically the capabilities of case management can be classified by different possibilities to cope with data and logically represent this data as the case.

- support any possible data type (analog and digital)
- capture data no matter where it is
- provide possibility of exporting a case and its information
- provide a highly usability friendly platform for collaboration
- ad hoc actions (ACM in particular)

All actions will be formed around this data, which result in a progress being neither serial nor predictable. Since case management is built for unpredictability and collaboration the framework or platform should handle information with priority. Prerequisite of starting with operational work on cases is that there is a case, i.e. information. Therefore, no matter what type of data it is and how it is stored the system has to take care of this and has to provide a mechanism to add such data to the case. Moreover to provide appropriate information, data has to be extracted from such data types. Even though this data is stored as an image, scan, email, voice mail or the web, it must be feasible to extract the information and process it accordingly (e.g. with OCR software).

Furthermore analog information on paper should be possible to seamlessly imported into digital formats best for further processing and usage, e.g. by converting it into electronic forms. Beside handling scans, with the help of appropriate apps mobile devices can ease attaching and capturing content to certain cases. In order to move, exchange or backup cases and its data in particular an additional feature is necessary to ensure a broad coverage of capabilities. Therefore possibilities for exporting data in either way, as a whole or as particular parts, as original format or enhanced format manipulated by the framework or system itself, need to be provided.

Some cases could require attaching data, which cannot be physically moved to a case management system. In this instance the case needs to link that kind of information somehow in order to provide a complete representation of all important aspects. Consequently the module responsible for content management has to consider potential manipulations of external information.

Besides data handling, the next most important point is considered for users to be able to effectively collaborate on a certain case. This includes email, instant messaging or electronic rooms. Most of the users will not be convinced by a system which cannot satisfy with usability. Developers often think usability is hardly important if functionality is not mature. In the scenario of case management usability provides key capabilities to easily increase the stock of information of cases. This most likely enhances functionality due to more accurate data, helping to handle tasks with all possible information.

Primarily case management is implemented to conclude insurance claims, but according to Gartner there are new areas [43] where case management and ACM in particular are considered to be valuable. Probably the number of areas where case management can create additional value will increase if offered solutions will be mature enough.

- mortgage origination
- investment portfolio management
- fraud detection
- grievances
- university admission
- customer complaints

Despite the fact that case management will increase its implementation range, the technology is hyped to high.

> »ACM hype exceeds the reality of what buyers are ready to adopt. Many of these solutions enable case workers to dynamically initiate ad hoc actions on a case, altering the execution behavior directly in the production environment. However, the ”devil is in the details” of vendors’ implementations [43].«

Thus implementation of ACM still needs to be simplified, even though organizations are not yet able to adjust their habits to new technology.

3.5.4 Monitoring changeable Processes

The new gained flexibility helps to simplify prespecified process models due to the fact changes are possible afterwards and even during runtime. Ironically, changing process instances leads to more complexity in the way of how much different process models are used during execution. Although this section does not deal with how to handle this high amount of workflow variations (also called process families [13]), it will provide an overview what analysis can help with for transparency and improvement of business operations [67].

3.5.4.1 Traceability of Execution and Changes

Logging of information before, during and after business process execution is a meaningful concept when it comes to the need of transparency. Beside the government, there are several additional useful ways to utilize this data. Additionally to the question who executed which process at what time, when considering execution of workflows, there is the need to know who changed which process at what time. Regardless if it is an ad hoc change of a single workflow instance or if it is a change of a whole process model.

Furthermore logs and other meta data can be a backup assurance because based on logs process structures or even instance states can be recovered in certain cases. Naturally this needs more than just an unstructured, semantically regardless text message that will be printed out to a temporary command line when a workflow starts and ends. In order to restore the state of particular instances for example, there is the need for storing even data being consumed by an activity before and after a potential manipulation of this data object. Due to the fact, that automated workflows will be executed quite often, in such cases the necessity of restoring process instances to any given time needs to be argued because storage consumption could be high. However, providing this kind of information indeed can lead to several occasions as described in the following section.

3.5.4.2 Optimization based on Changes

Turning single letters into real knowledge is key in order to improve procedures. So why not getting out every single drop of information of the data produced by process execution. It is no novelty that process logs are mined to gain information. Nevertheless rapid changes make it possible to analyze those alterations and deviations from the prespecified path. Mining of process logs strongly depends on various algorithms including heuristic principles and statistical methods. Actually these algorithms turn plain letters into information by extracting, aggregating and comparing data.

Based on this information process engineers are able to optimize models where certain paths became obsolete, detected by analyzing logging information. By studying a lot of process variants, which are based on a certain process definition, it is possible to learn which improvements might be reasonable to implement into the main process. Therefore results are able to recommend optimizations for the main process. Related to detecting obsolete paths on the other side, it is able to say which paths are the most frequently executed. Based on this knowledge, decisions which process sections to focus on for optimization purposes become easier for example. It has to be mentioned that there are a lot more optimizations, which can be extracted by process data analysis. Being able to keep track of even loosely specified processes through the paradigm of ACM, analysis can be extended to this kind of processes.

To sum up ACM is just a collaboration platform, which has to provide excellent usability features similar to real life interaction with both humans and case data at the same time. Additionally, technology has to be mature in order to cope with sophisticated content management and ad hoc actions. According to Gartner [39] ACM is well suited for processes with high risk exceptions due to its focus on data and loosely integrated processes.

4 Flexibility vs. Automation

State of the art technology is able to process many business operations automatically by either custom designed applications, process engines with respectively designed workflows or even a combination of both. Nonetheless changing requirements and unpredictable exceptions are challenging highly standardized procedures. In most cases processes are hardly possible to react to ad hoc changing requirements without the need for adaptation by humans. By dividing a BPMS into build time and runtime modules, it is perfect for prespecified workflows. However, which architecture would be better when comparing traditional business process management systems and ACM systems. Probably it simply depends on the scenario. Rare issues provide possibilities to machines deciding what to do in case of exceptions [61, 52]. Theoretically every predictable exception can be implemented in a prespecified workflow. Even though this could be achieved, it would not classify a process to be flexible.

Humans reach decisions based on their experience, intention and knowledge. Conventional programming paradigms hardly support decision management [77]. Some kind of artificial intelligence (AI) would be needed in order to let machines make decisions based on experience and knowledge. Partly this already is done in the field of IT service management (ITSM) by adding semantics and implementing ontologies in order to let the system make assumptions. Certainly, ontologies and semantics are just a small part to achieve a fully automated and largely flexible system. Eventually a question remains, if it is really effective to develop such a totally automated system, which probably needs a lot of specialized maintenance and usability is beyond question as well.

4.1 Usability increases Flexibility

Basically technical capabilities have to be provided in order to ensure a flexible workflow. As already mentioned a BPMS has to provide an underlying technology for migrating process instances (cf. section 4.6.3) and updating process models (cf. section 4.6.2) ad hoc like. But with this underlying technology there are always experts and specialists needed who are able to actually achieve such changes. End users would have no possibility to perform similar changes without hiding complexity. Therefore usability needs to be almost perfect.

There is a reason why there exists specialists for every industry, sector and module. BPMS, especially if it is implemented throughout the entire company, can contain dozens of modules and custom extensions. This fact often makes it hardly possible for one person to stay on top of things. Certainly this would not be an issue once implementation of the entire system is finished and runs smoothly. But there will hardly be any stagnation in the whole company. There is always something, which will be enhanced, needs to be updated and is causing trouble beside repairing ordinary parts of the system. Assuming just an update, it has to be executed on one part of the system in order to be operational on the hardware. Who can tell what dependencies there are on the remaining system? Every small adjustment would be a stress test for the entire organization which leads to high risks. Therefore ITIL has a strong focus on managing dependencies [71] in order to be capable of maintaining IT infrastructure in companies. In order to minimize the risk of dependencies to other programs, software architecture plays a big role in this kind of question and most of the business process management systems make usage of this expertise to provide loosely coupled modules. Hence, to take care of the design means to increase usability of software. In this case it is not just about the user interface for end users, but it is the method and intelligently making predictable systems, which can be maintained without the need of working at least five years in the same company and occupation.

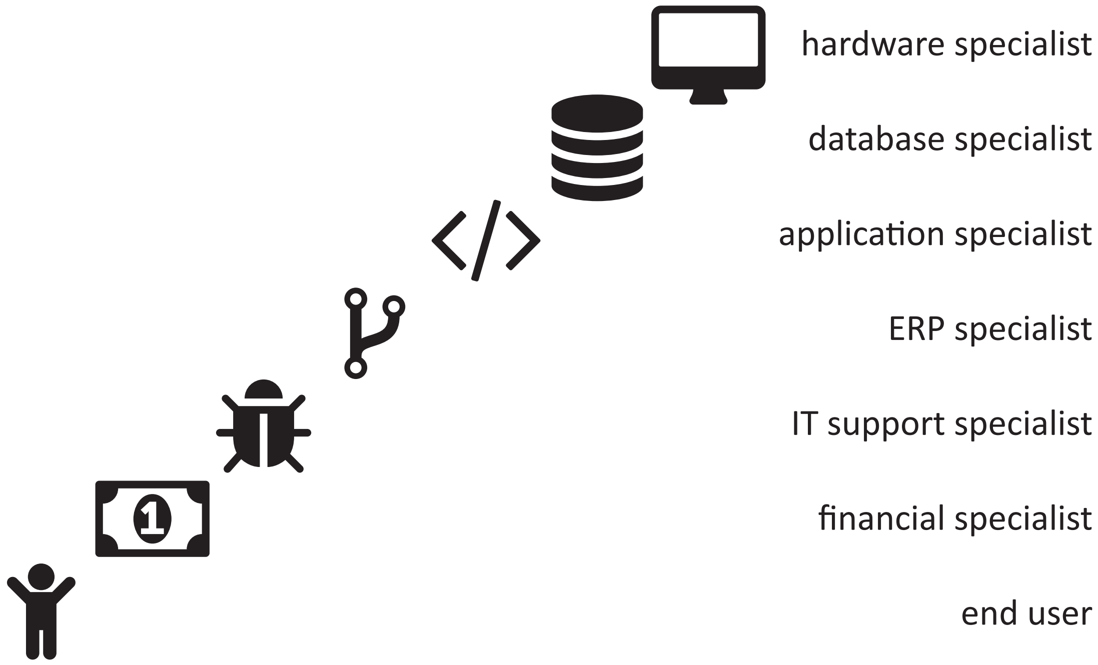


Figure 4.1: Usability in different Staff Classifications

When thinking of usability the first things come to mind is the graphical user interface (GUI). But it is not the look and feel, what makes the user experience (UX) appropriate for certain systems. Providing an overview and clear structure added up with a sensible workflow helps end users. As can be seen in Figure 4.1 the need for UX could be enabled in many areas where people are operating. End users who are working with a well designed GUI might have no additional need for instructions of others and are limited to do something wrong. By applying similar UX to all areas shown in the incomplete list of sectors (cf. Figure 4.1) flexibility even can be increased in all scopes. There are different possibilities how to ensure usability. Beside a clearly structured architecture, ensuring easy adaptations, just simple naming conventions can lead to desired clearness.

At the end of the day, the end user has to handle it in order to be able to enhance working with flexibility features, which only can be ensured if simplicity is given respectively. Complexity of operations has to be hidden [17, 57] behind descriptive buttons and potential usability features combined with appropriate feedback for the user. Additionally a single system can provide simplicity and no matter if processes are fully automated, partly automated or are handled within ACM modules, tracking provides the favored transparency for the company.

4.2 Different Adaptation Types

Before actually dive into the technical aspects of updating workflow instances in section 4.6, there is some background, which has to be mentioned beforehand. In order to know how to deal with different possibilities and certain kinds of adaptations, Figure 4.2 is providing a comprehensible description. The classification is based on research [22], conducted by Gartner in order to supply knowledge about leveraging flexibility for business adaptability.

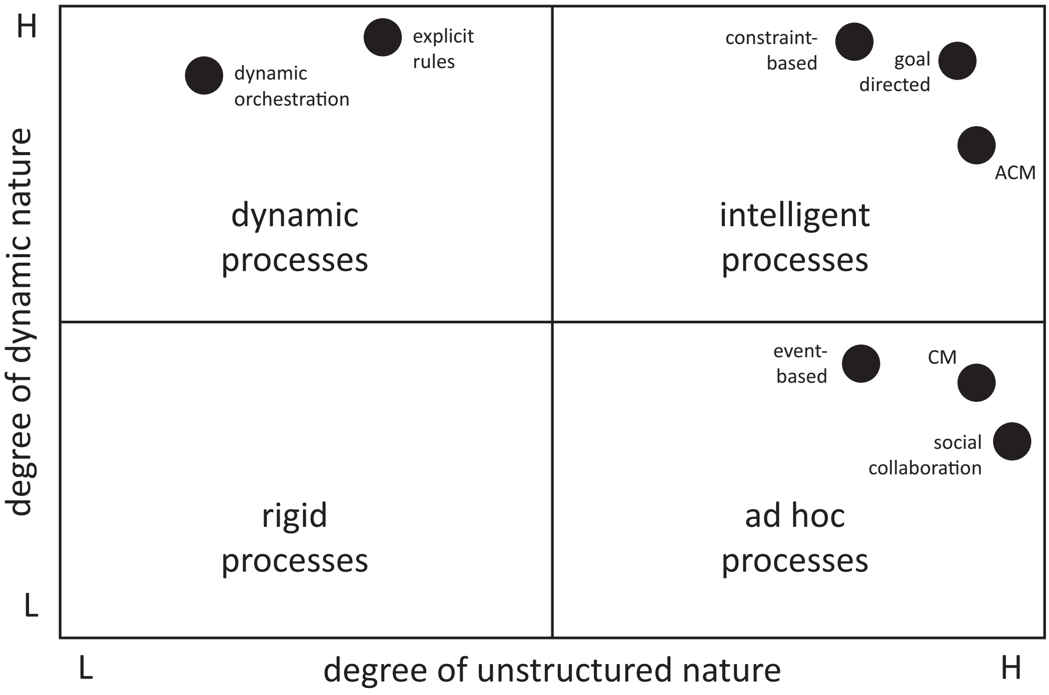


Figure 4.2: Axis of Adaptability (based on [22])

According to Figure 4.2 there are three classifications of flexible processes, besides the ordinary static processes. This classification is driven by two values, which is the degree of dynamic nature on the y-axis as well as the degree of unstructured nature represented by the x-axis.

Firstly, dynamic processes are characterized by a high degree of dynamic nature combined with a high structure, i.e. basically it represents an enhancement of the ordinary

process by adding flexibility capabilities respectively (cf. section 3.4.2). An important fact is that this kind of flexibility mostly has to be specified beforehand, which decreases flexibility significantly. The ad hoc capability of BPMN would be part of this classification.
Now instead of having a high degree of dynamic and a high degree of structure, ad hoc processes represent the opposite configuration with a low degree of dynamic and a high degree of unstructured behavior. Flexibility is ensured by collaboratively working on a shared process instance like it is performed with case management.
By combining high dynamic nature and unstructured behavior intelligent processes provides the classification in the upper right quadrant. In principle automation of highly flexible processes (cf. section 4.6.4) is rather difficult but intelligent processes potentially provide this capability. However most of the execution has to be done by humans with the help of optimized systems and frameworks like ACM (cf. section 3.5.3).

4.3 Flexibility Issues

Computers and thus automation need clear structured constructs. Flexibility is dealing with unpredictability. Consequently problems for automating flexible processes are inevitable.

Based on the taxonomy for flexibility (cf. Figure 3.2), provided by [67] in section 3.3, issues arise for all four branches. The need for change, even in business processes is inevitable. Sooner or later every process model needs to be adapted (cf. section 3.3), be it because of changing laws or because of broken hardware of the underlying server (cf. section 3.2). Issues for variability, looseness, adaptation and evolution will be illustrated by the following sections.

4.3.1 Variability Issue

When coming to the need of differences between products and services it becomes clear, why variability drives the need for flexible processes. A very bold and simple example for variability can be found in the product line of many car manufacturers. No matter when looking for a certain model with distinct features or, a little more abstract, a van where many different models could be appropriate, product variability ensures the need for different processes, however with similar behavior. Thus, a respective business process model can provide the same source process with different variations to handle its variants.

4.3.2 Looseness Issue

As already mentioned, unpredictability plays a big role in being flexible because there are processes and operations, which cannot be specified a priori. However it should be possible to automate at least parts of a process and handling every kind of process in a

BPMS. Because of its unstructured characteristic systems have to be able to deal with this nature by providing ACM modules for example (cf. section 3.5.3.3). Looseness is needed in order to handle emergent actions only popping up during the execution. Processes of this type can also be called goal-directed (cf. section 4.6.4) because just the goal is known beforehand (cf. the upper right quadrant in Figure 4.2).

4.3.3 Adaptation Issue

In order actually to be able to adapt process models a system must provide functions to update models and its related process instances. Deviations of business processes and real world operations will occur frequently. Therefore those processes, which are out of fashion have to be realigned to its real world processes respectively. How this can be achieved from a technical point of view will be described in section 4.6.

4.3.4 Evolution Issue

Evolution has already been described in section 3.3. Due to business environmental changes, real world and thus its business process models have to adjust accordingly. Even though there are no changes in the business environment going on, continuous improvement (Deming Cycle) optimizes workflows. Therefore a BPMS evolves with its corresponding real world processes.

4.4 Flexibility Requirements

Besides providing convenient design for any kind of work throughout the entire company, UX provides just an appealing foundation to enable employees to perform easy changes without an extraordinary amount of risk. Be it either by clearly structured infrastructure or mature user interfaces. This section focuses on technical requirements [67] in order a system to state itself a flexible solution. As already can be discerned from previous sections, it is not enough to just technically provide versioning support for process models. According to Oracle [45] flexibility has to be provided in UX, in assigning work to resources (primarily humans), in enforcing business policies, which includes the change during runtime as well as case management, as it is done through ACM. Depending on the capability to changing processes many problems arise, which have been handled in previous section 4.3 and causing the founding, illustrated in the following sections.

4.4.1 Configurability

Plain old process models were static with a high effort to adjust them. Variability needs are driving configuration of processes, which makes it usable to alter an existing process due to slightly differences. Making process models being configurable

[66] it eases process creation for certain variants. According to a patent [36] process templates are used for configuration purposes.

4.4.2 Differing Process Models

As already stated some business operations need a loosely specified description of a process. Especially when it comes down to long running processes with the potential of high risk exceptions [39] ACM capabilities (cf. section 3.5.3.4) are needed. Applying ad hoc changes to process instances requires the system to handle such changes. Depending on this possibility, additionally it increases the demand of features like secure permission management, proper validation and gapless business compliance.

4.4.3 Versioning Support

There are not just process instances, which need to be adapted but also corresponding process models have to be optimized and thus need refactoring. Coexisting versions of different processes providing flexibility in order to choose which version a created instance should refer to. As will be illustrated in section 4.6 there are some questions and possibilities concerning a to be updated model and its already running instance.

4.4.4 Transparency

Obviously, unpredictability requires increased concentration on changes made on behalf of ad hoc adjustments (cf. section 3.5.4.1). Traceability in combination with accountability ensures to reconstruct situations and previously made changes if necessary. But this is not just true for ad hoc changes which are unpredictable. Hence, transparency has to be enabled throughout the entire execution engine in order to trace back and browse history adaptations. Even though ACM modules might provide unstructured execution of certain flows, people must not be able to bypass the system. Apart from that workers would be able to execute tasks without logging history for potential future utilization.

4.4.5 Business Compliance

No matter what ad hoc changes might be able to alter in workflow instances, all of the changes have to take place inside certain boundaries and limitations. Those rules have to be kept in mind when changing processes respectively. Violation of federal laws can lead to extensive penalties. Furthermore it can ruin businesses and their prestige.

4.4.6 Permission Management

Adding capabilities to change business operations on the fly strengthens the need for robust and secure permission management. Furthermore the user role concept has to be suitable for changing permissions (e.g. permission to publish a new process version). Maybe there is the need for a four eye approval when it comes to new version publications. Moreover, is it allowed for someone who updates a process model to update running instances as well? Several questions arise when granting permissions to users depending on the field concerned processes belong to. Either they contain sensible data, which are not allowed to depend on a decision of just one person but instead need two experts who can decide. Or there are just noncritical processes which are allowed to be adapted by a bunch of people. Howsoever there is one thing which has to be ensured anytime. Access has to be restricted to authorized and known users not least because transparency depends on accountability.

4.4.7 Validation Support

Even in boringly predictable processes errors can occur due to dead locks or missing data. When thinking of updating workflow instances during runtime things have the potential to get even worse. Since running processes are in a certain state and hold data of a certain state changing tasks in such a sophisticated scenario could easily lead to big errors. Therefore every single change, howsoever small, needs to be checked for syntax errors, potential data errors, blocking activities and compatibility to its dependencies (cf. section 4.6.1). Those validation checks can also be seen as some kind of UX feature. If users, especially end users, can be sure changes they make can only have positive impact on the operation they are likely to use the BPMS because they see its benefits and not just because they have to. Users need to trust the system and validation helps them to check before something is wrong.

4.4.8 Concurrent Changes

Concurrent changes can be either by simultaneous adaptation of the same process or by overlapping changes. The latter describes the more sophisticated ones. Whereas there are already solutions for concurrent changes of the same object my different people, focus lays on overlapping changes. Basically overlapping changes describe adjustments to be made on an already changed processes. This is better illustrated by Figure 4.3, which shows an ad hoc change performed on an instance followed by an evolutionary change of the corresponding process model and moreover its immediate change of the instance during runtime.

Starting at time t the process model M is created in its very first version. Next to that at time t+1 an instance is created based on process model M. During execution of instance I an ad hoc change takes place triggered manually by an authorized user. This results I to be derived from M but changed individually, which leads to I’. Afterwards M will be optimized and version 2.0 will be published. Its corresponding instance I’

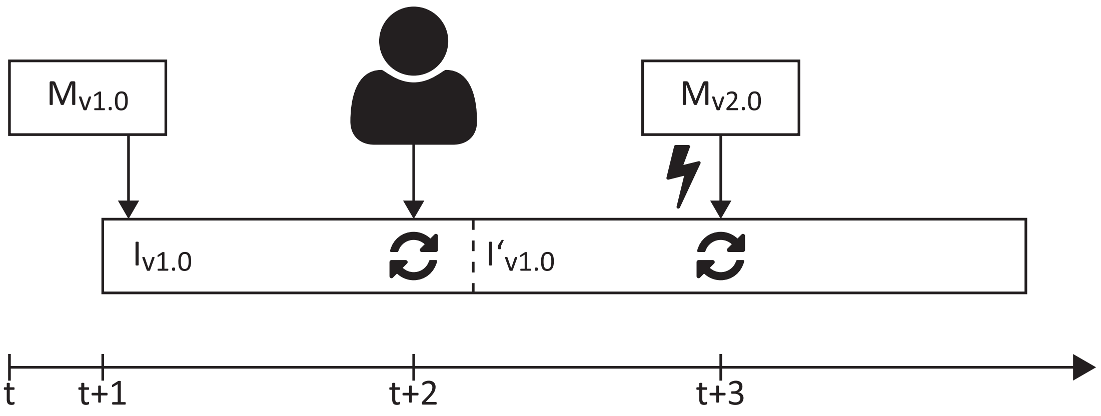


Figure 4.3: Illustration of an overlapping Change

now should be updated instantly to the new model version. Since I' contains changes not considered by its model the update of the instance during runtime at time t+3 is more sophisticated than before because of unknown changes. Therefore special mechanisms have to be provided by the PAIS in order to take care of such changes.

4.4.9 System Learning

The willingness to utilize history changes to improve workflows are considered in many publications, e.g. [33, 88]. Obviously companies are interested in the reuse of existing adaptations, because it has the potential to save time and money (cf. section 3.5.4.2). Again, usability can have a considerable impact on reusing existing actions. For example, when certain exceptions occur during process execution, recommendations made by the system could encourage users to continue execution with existing adaptations made in the past. However this strongly depends on the representation of history changes and its accuracy to fit their reusability in appropriate scenarios. Although system learning hardly seems to be mandatory as a flexibility requirement, it indeed contributes to the overall system package to improve and facilitate users in their business operations.

4.5 Pattern Power

Automation needs clear structure and this becomes even more critical with enabling users to dynamically alter workflows, deleting actions and applying ad hoc changes to process instances. As heavily used in the domain of software engineering, patterns provide known and standardized best practice recommendations of frequently used implementations and issues. Similar to software development the implementation of business processes can utilize patterns to ensure development of parts according to best practice knowledge.
Actually software development is a more custom approach to create tools, much closer

to the computer and technology theme than to the business context. However, this cannot be generalized. Eventually, software development is to translate real world processes into a certain application with the help of a certain notation. In this case this notation is not as abstract as BPMN for example, however there are abstractions in the languages used in software engineering when comparing Assembler[1] and Java[2]. Basically every automated business process implemented with WS-BPEL for example can be implemented within another programming language as well and probably a BPMS is based on Java. Anyway, depending on the given scenario BPM aims to hide complexity by contrast to programming languages. However patterns does not provide notation specific or technology specific constructs since pattern based approaches should be usable throughout the entire scope of process design.

4.5.1 Expressive Modeling

Using patterns in process models means to increase the expressiveness of a process [67]. Business processes by nature can be highly sophisticated, which makes a process hard to maintain. Expressiveness can counteract complexity. This can easily be illustrated by Figure 4.4 and Figure 4.5 by modeling a simple process for creating cereal. Since the order of how ingredients are added to the bowl does not matter, a process specification for a cereal with four distinct ingredients becomes confusing. Figure 4.4 hides most of the possible sequential orders, but it can be imagined how sophisticated it can get when thinking of its permutation.

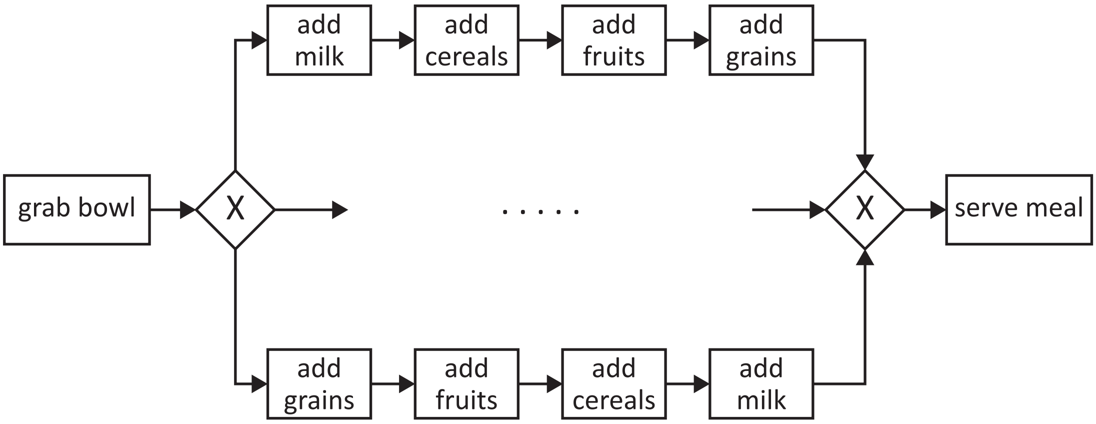


Figure 4.4: Example of ordinary process

Unlike modeling every permutation, Figure 4.5 just uses the interleaved routing pattern[3] to achieve the exact same specification with extreme decreased complexity. Obviously readability is given, if implementing the pattern in case of the cereal process. Additionally to improved readability, patterns can be an extensive time saver. Due to the fact of an optimization initiative the process of making a cereal is cutted by removing the »Add fruits« activity. Whereas in case of the pattern usage there is just

[1] Hardware programming, http://asm.sourceforge.net/

[2] What is Java, https://www.java.com/en/download/faq/whatis_java.xml

[3] Workflow patterns, Pattern 40, http://www.workflowpatterns.com/patterns/control/new/wcp40.php

the need for removing the respective activity, updating the process model in Figure 4.4 would lead to some effort.

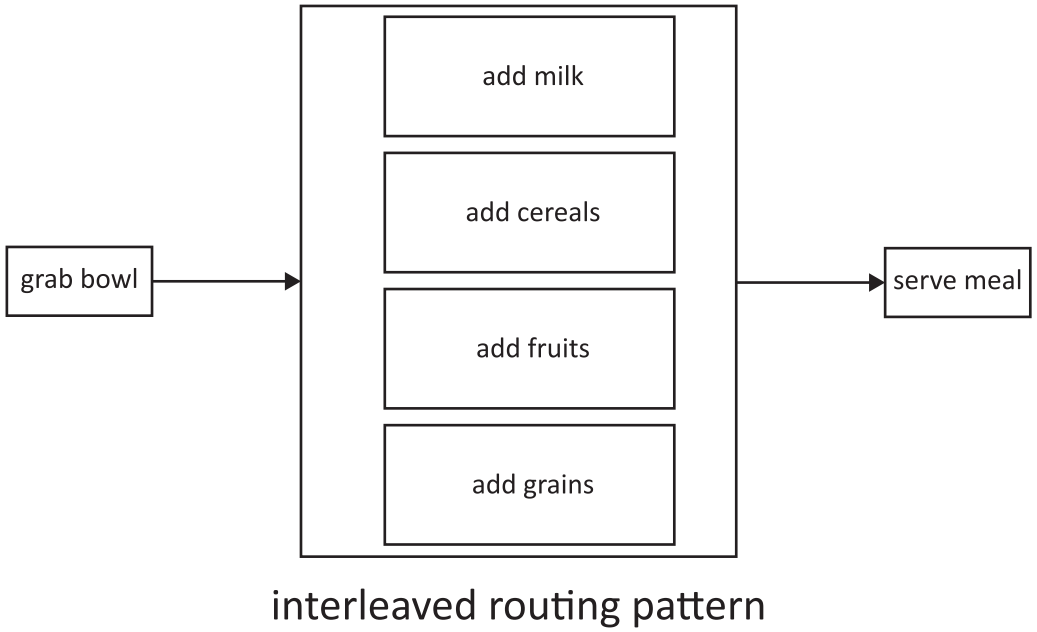


Figure 4.5: Example of process with pattern usage

Despite all advantages of using patterns, usage depends on appropriate scenarios. Similar to symbols used in BPMN, patterns express certain semantics, which needs to be known by process engineers. As pointed out by [18] patterns can have its downsides. Additionally, high expressiveness can lead to more complex BPMS because pattern support needs to be ensured [67].

4.5.2 Control Flow Patterns

Control flow patterns aim to support process modeling of the control flow perspective [75]. In order to simplify understanding and some other aspects, already mentioned in the previous section, control flow patterns standardize constructs of the activity flow. By defining patterns for sequential flows or patterns for parallel flows these patterns raise modeling formations to an abstract view. The interleaved routing pattern in Figure 4.5 is an example for that.

4.5.3 Data Patterns

As the name suggests, data patterns [73] focus on a different perspective of the process model. Since processes utilize data during execution, this data can be represented in the model. Consequently those patterns standardize representations of distinct scopes of variables during workflow execution. Furthermore it provides comprehensible representation of data transfers and preconditions regarding data.

4.5.4 Resource Patterns

The resource perspective of workflows can be identified by various resources. Resources can be human beings, virtual equipment or physical objects. In order to their well arranged usage of resources the range of resource patterns stretches from different kind of distribution possibilities, authorization issues, allocation to visibility [74].

4.5.5 Exception Handling Patterns

Exception handling patterns [76, 53] can only be used for predefined exceptions, which are known prior to a workflow's execution. As stated by [76] the type of an exception has an impact on how an exception can be handled. Handling an exception, which occurred during execution of a certain activity makes a difference by contrast of handling an exception, triggered by a deadline for instance.

4.6 Updating Work Flows

As described in the previous sections, several preconditions have to be fulfilled in order to provide change capabilities smoothly throughout the entire process portfolio of an organization. Since computers and servers rely on structured and strictly defined paradigms, there is a reason why flexibility and workflow execution is a rather complex topic. Nevertheless examples show that it can be achieved with the right concept and appropriate technology as stated by [82, 81]. This section focuses on the hard facts, i.e. the technical issues behind ad hoc adaptations in workflow instances and its definitions.

4.6.1 Preconditions of Adaptations

It has been already mentioned that a BPMS depends on a clear structure. When thinking of ACM the concept itself is focused on loosely coupled processes. Although the basis for this concept is built on strictly defined paradigms. In traditional process aware information systems the concept of how processes are handled might not be as flexible as it could be. Therefore changed processes have to be validated and tested as good as it is possible in order to prove they are ready to be executed. An overview of preconditions has been given in section 3.3. Those preconditions [67] are namely:

- structural correctness
- process definition soundness
- correctness of data flow
- state compliance

Explaining those four preconditions, the assumption has been made that at the initial point the processes to be adapted already fulfill the preconditions declared. Thus, when checking for the structural correctness of a process model it is a proper validated definition. First, when changing a process, syntactical correctness has to be ensured, i.e. a change performed on a correct process has to result in another correct process, which is true for validation in general. Syntactical correctness depends on the language used to represent the process. Since each language has its own, sometimes similar, grammar it has to be proved that the process engine can read this language properly. Otherwise errors are caused by the BPMS. Structural correctness validation is rather easy because is can be accomplished during modeling the process. In Figure 4.6 obviously an arrow is missing between Task 1 and Task 2, which would be instantly lead to feedback in the design application respectively.

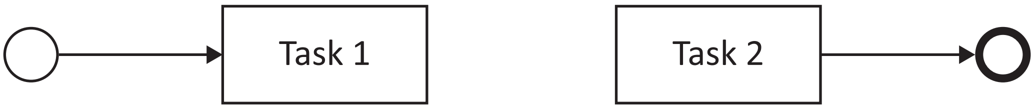


Figure 4.6: Structural incorrect Process

In comparison to the correct syntax, checking for a process' soundness increases the inspection effort. According to a model's soundness [90] it has to be ensured that each process finishes in a proper state. This fact is strongly related to the third point of the enumeration above. Additionally to the option to complete it is important that the entire process is completed without any dead locks for example. In order to complete a process in a proper state all corresponding activities in combination with its instance have to finish the execution. Implicitly this results in the condition of no dead activities. All of the conditions need to be fulfilled for sub processes as well. When taking Figure 4.7 there is the possibility of a dead activity inside the exclusive choice. Depending on the condition either Task 1 or Task 2 could be the unreachable activity. Therefore the condition needs to be validated. If for example the condition results in a hundred per cent probability for one single branch the exclusive choice would omit the other branch. Consequently one branch is never reachable, which makes an activity never executed.

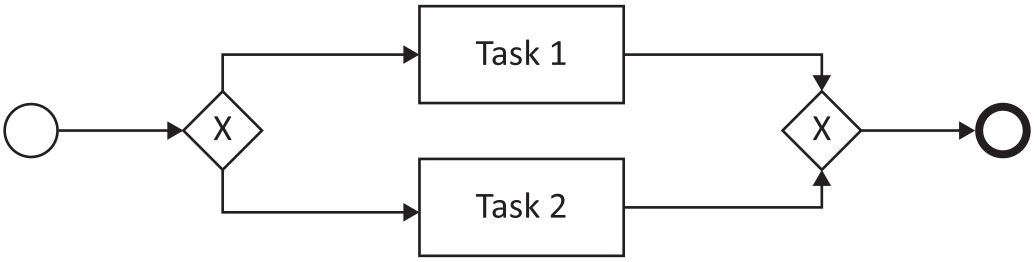


Figure 4.7: Process with no given Soundness

Workflow execution depends on data objects. Data objects can be easily represented by variables and as such data objects can be manipulated by certain activities in the process. Manipulation can cause loss of data, stored in such variables. Consequences contain missing and unnecessary data as well as lost updates. Figure 4.8 outlines a lost update scenario. In this scenario Task 1 writes data to variable x. Subsequently Task 2 overwrites variable x with other data. Task 3 now receives data in x but does not know that variable x has been overwritten and the origin data stored by Task 1 is lost. Although this might not result in an error, the consequences can be fatal, e.g. wrong process output.

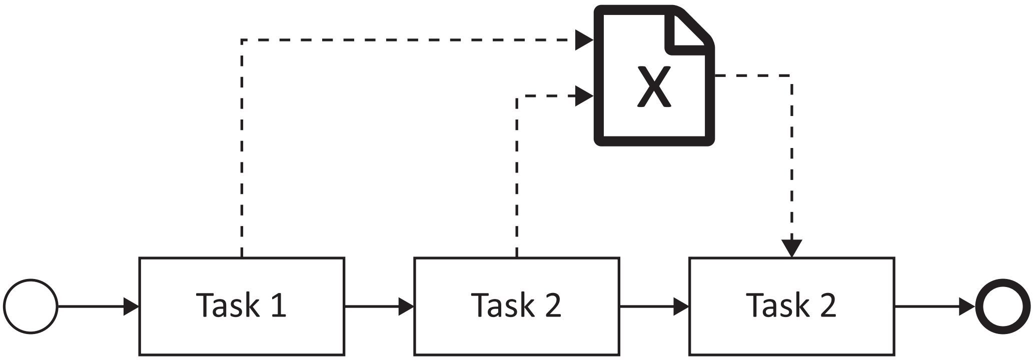


Figure 4.8: Process with a lost Update

State compliance does not belong directly to the modeling perspective, rather to the executing perspective. Accordingly in the context of process adaptations instances are changed, i.e. running processes will be changed. Since instances are running, they are in a certain state, which means the flow is at a specific point in its execution. Depending on the progress and on the location where the alteration is needed a process is either possible to be changed or not. If an update on a workflow instance would not be validated prior to its implementation, errors can arise. Figure 4.9a shows the process model of an already running instance. The instance should be changed of this trivial model. Task X will be put after Task 1 and before Task 2. Looking at the instance of the model in Figure 4.9b, Task 1 has already been finished whereas Task 2 is in a running state indicated by the play sign. After applying the change of the insertion of Task X the instance is displayed next to the pristine instance illustration. Without managing states Task 2 is still running while Task X will never be touched because progress is already too far.
As already referred to, instances might be changed during ad hoc adaptations. However this is not the only set up where changes to running processes are likely to happen. Even when process models are updated it can be possible to update its corresponding instances to the new version. In either case, states have to be managed in order to ensure compliant implementations.

So validation helps to ensure smoothly running instances even if they are changed during runtime. Unfortunately some considerations have to be made, regarding Murphy's Law. Thus, if »anything that can go wrong will go wrong« is true, then even validation sometimes can fail. Especially incorrect states cause problems that have to be dealt with accordingly. As stated by [67] there are five strategies that help to cope with non compliant process instances.

- partial rollback
- delayed migration

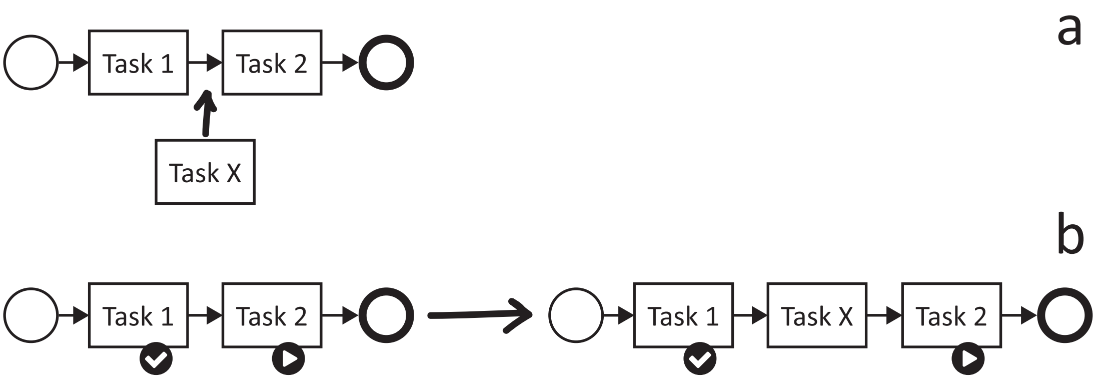


Figure 4.9: Non state compliant Instance

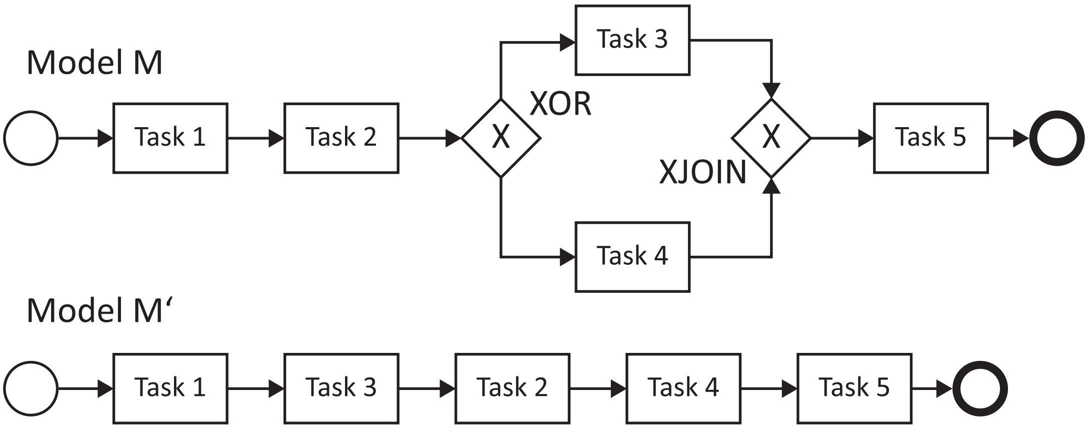
- always migrate
- instance specific adjustments
- type specific adjustments

Simply put, partial rollback reverts the non state compliant process instance to a compliant state. This does not necessarily mean that the entire adaptation will be reverted, but essential parts of that change, e.g. undo certain additions. In order to deal with running processes it is also possible to delay the migration. This can be achieved if the section to be changed is inside a loop. If the change is not state compliant when the first iteration after the adaptation is in progress, it can wait to the following iteration when it becomes state compliant and the update can be part of the flow. Similar to the partial rollback, delayed migration can only be executed in appropriate situations.

The next three strategies represent some more advanced techniques to ensure state compliance. Firstly, the so called always migrate strategy is quite easy to explain. Every instance will be migrated unless it is non state compliant. Otherwise the change will be neutralized for the non compliant instance. As a consequence the original changes would not take effect on those non compliant process instances. The name already suggests that instance specific adjustments might be similar to the previous strategy and yet it is different. If insertions of additional activities do not matter where to happen for example, the insert location can be adjusted. Depending on the progress of an instance, the change can be applied after another activity. Again this strategy depends on the situation of the change because this cannot be applied to deletions of activities. Lastly, type specific adjustments follow the other direction of the instance specific adjustments. In order to migrate as many of the running instances the change itself will be altered. The number of migrateable instances can be increased by applying changes as late as it is possible in the process model.

4.6.2 Model Changes

Process model changes are quite similar to instance changes, however validation is easier. Since models do not have an execution state, there is no need to check for the state compliance at first glance. Depending on the impact of the model changes, corresponding instances, which might adopt the updates from their models might be affected (cf. next section 4.6.3).



Change primitives from M to M'
1. delete edge T1 T2
2. delete edge XOR T3
3. delete edge T2 XOR
4. delete edge XOR T4
5. delete edge T3 XJOIN
6. delete edge T4 XJOIN
7. delete edge XJOIN T5
8. delete node XOR
9. delete node XJOIN
10. add edge T1 T3
11. add edge T3 T2
12. add edge T2 T4
13. add edge T4 T5

Change operation from M to M'

1. move Task 3 between
Task 1 and Task 2

Figure 4.10: Comparison between low Level and high Level Changes

Basically model adjustments can be handled by two different approaches [67]. Differentiating between low level and high level changes reveals that low level changes have their downsides. Low level changes are defined by primitives, which basically consist of add node, remove node, add edge and remove edge. Thus when removing a node, corresponding edges have to be removed as well, which can be considered as error prone in certain cases because it has to be done manually. High level changes on the other hand hide this complexity. With change operations such like add, remove and move not just single nodes but also blocks and groups can be updated easily. A comparison can be identified by looking at Figure 4.10 where process model M is transferred into process model M'.

4.6.3 Instance Changes

Changes at the process model level can have direct impact on corresponding instances. Especially in terms of long running instances such affects can become necessary [67]. Handling or not handling those affects on running processes can basically be achieved in three ways. First, leaving the currently running instance finish with the behavior in its initial version, which refers to not handling in this case. Handled will be the instances in the second way of dealing with this cascading update, even though terminating instances respectively seems to be some kind of rude, which can be done by systems. Lastly, the actual way of handling updates for instances allows for a migration of the corresponding instance. An advantage of the first and second method is that almost no manual task is required whereas in case of the third method manual activity is rather likely. Manually migrating instances would be error prone without validation mechanisms (cf. section 4.6.1). Thus proper migration can be achieved with adjustments of the current state of an instance in order to proceed execution smoothly. Adjust variables, revert execution to another location in the execution flow and consequently adjust states of activities. Therefore providing the capability for altering instances' past is essential.

Process instance changes, which are not caused by its updated models are the result of ad hoc changes. Ad hoc adjustments to running processes are often required by handling unanticipated exceptions in order to ensure most flexible handling [67]. Therefore a PAIS has to provide several techniques to allow a user to handle either unforeseen or anticipated exceptions (cf. section 3.4). Needed capabilities include manipulating nodes (e.g. addition and deletion), postpone the execution of activities, to antedate execution of activities no matter if preconditions are fulfilled or not. Technically speaking adaptations of instances indeed can lead to changes in the corresponding process model, however logically an instance can be seen almost independent from its definition. When talking about ad hoc instance changes the latter will be assumed.

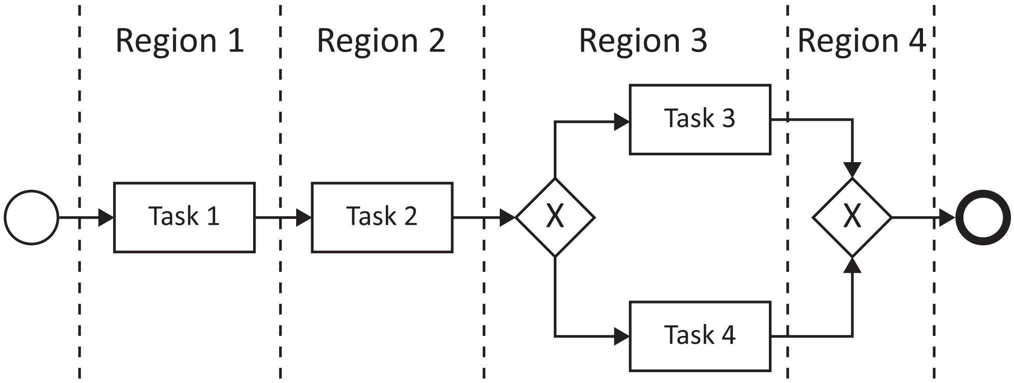


Figure 4.11: Process Regions and State Adaptations

Instance changes must not violate correctness and have to ensure proper execution, as explained in section 4.6.1. Furthermore adjustments made in running processes must not affect other instances at the same time. Proper execution includes sufficient checks for state compliance. This depends on the process region to be adjusted. If for example the execution flow is just in the beginning section of an instance, no state adjustments would become necessary. In case of the process in Figure 4.11 even if Task 3 has been completed and Task 4 is still in a running state, adaptations could be made between the parallel join and the process end in Region 4 without the need for further state adjustments. Thus, no yet entered regions do not need state adaptations when changes are applied to them.

Even though changes happen manually, it has to be ensured to still provide transparency of adaptations. (cf. section 4.4.4). Moreover it can be differentiated between permanent and temporary changes [67]. In case of ad hoc actions generally thinking such changes would be valid until the completion of the instance. Straight flows, containing just exclusive ors, parallel splits and activities, do not need to make decisions on permanent or temporary changes because changed regions will just be executed once. When thinking of loops such decisions become reasonable. Certain iterations perhaps need to be handled by an ad hoc change whereas all following iterations in a loop do not need this adjustment. Therefore temporary changes would definitely make sense.

4.6.4 Automation of Adaptations

In order to define what is possible when thinking of automated adaptations, the more interesting question is what is actually useful. A fine line can be imagined regarding exception handling and ad hoc manipulations on running process instances. Assuming ad hoc actions as something more manual, exception handling could be defined in a much more automatable way. Exception handling does not compulsorily have to be done manually. Analogously to programming languages' exceptions, which will be thrown in certain situations, can be caught and can be handled accordingly. In this case the exception handling in software development is predefined and thus easier to handle but unfortunately highly static. Nevertheless separation of ad hoc changes and exception handling is still fuzzy.

When thinking of hundreds of process instances referring to a single process model, automation can save huge amounts of resources and keeps the flexibility feature of the BPMS still working, no matter how much instances have been created. Therefore automated instance migration (cf. previous section 4.6.3) can have a huge impact on the feasibility of instant changes. Compared to an ad hoc change, an adaptation caused by a model optimization knows how the adaptation looks like because it is predefined by its corresponding model. Nevertheless state compliance can be an issue depending on the progress of the running process. In case of unpredictable exceptions, automation struggles because of various problems. Ordinary programming languages are static and cannot adapt during runtime. However the concept of dynamic software development is rather old (cf. Lisp[4] programming language and its state of the art dialect[5]) but still exists.

Although ad hoc actions first and foremost are handled manually, automation can still be useful in order to increase UX and thus supports the user performing an ad hoc change. Giving automated suggestions based on historical data could be one method of assistance followed by an excellent validation check as already discussed in section 4.1. Beside automation of ad hoc changes, in order to automate exception handling, first of all a PAIS has to detect such a situation whether it is explicitly detected or implicitly identified. In either way the possibility of notifications in case of an autonomous change should be considered in order to inform persons responsible of something automatically happened. Basically there are three concepts of how to automate adap-

[4] Lisp, http://groups.engin.umd.umich.edu/CIS/course.des/cis400/lisp/lisp.html

[5] Clojure, http://clojure.org/dynamic

tations [67].

- rule based automation
- case based automation
- goal based automation

Rule based approaches are also called event condition action (ECA) approaches. This method detects exceptional situations and can identify changes needed for certain process instances. Adaptations of running instances are described in [61] where not yet entered regions can be modified by adding or deleting activities. By specifying an abstract ECA rule model, parts to be altered can be identified and modified respectively. However this approach includes estimates and thus still contains uncertainty.

Case based reasoning utilizes already used changes to be applied in appropriate situations. In case of an exception previous adaptations are fetched and if suitable past changes could be found those changes are implemented instantly. Before such changes are applied to an instance, appropriate locations, where the change should be implemented will be determined.

Finally, goal based approaches define the goal of a process and extract the process model out of these specifications. Since the way the goal is accomplished by this approach is dynamic, occurring exceptions during runtime would lead to adaptations accordingly. This approach is still limited to some scenarios because special treatment of loops and other more sophisticated constructs are not that easy to handle by goal based specification. Gartner mentions a company[6] in their cool vendors report [27], which offers goal driven work support by providing plans to choose from specified by a certain goal. The organization uses an AI approach to create such plans. Gartner states this as an advanced approach in the ACM sector.

[6] IActive Intelligent Technologies, http://www.iactiveit.com/

5 Highly flexible Process Engines

There are several engines out there, covering the execution of predefined workflows and predictable exception handling as well. Some provide proprietary solutions for an even more abstract view on the implementation of business processes in contrast to ordinary programming. Others offer possibilities that can handle the execution of processes as part of an overall business suite. All of the following examples provide flexibility during the execution of processes to a certain extent. How far these capabilities can cover the previously mentioned needs for providing a flexible BPMS, will be shown in detail. However, the focus will be laid to the technical realization of how a process engineer or programmer can alter workflows during runtime. Usability, as described in section 4.1 as an impact on the degree of how flexible a PAIS can be, will not be evaluated. Nevertheless it can be implied, how convenient it is for a process engineer to apply changes on running instances. Certainly this will not include the perspective of how it can be achieved by the end user.

5.1 Windows Workflow Foundation 4.5

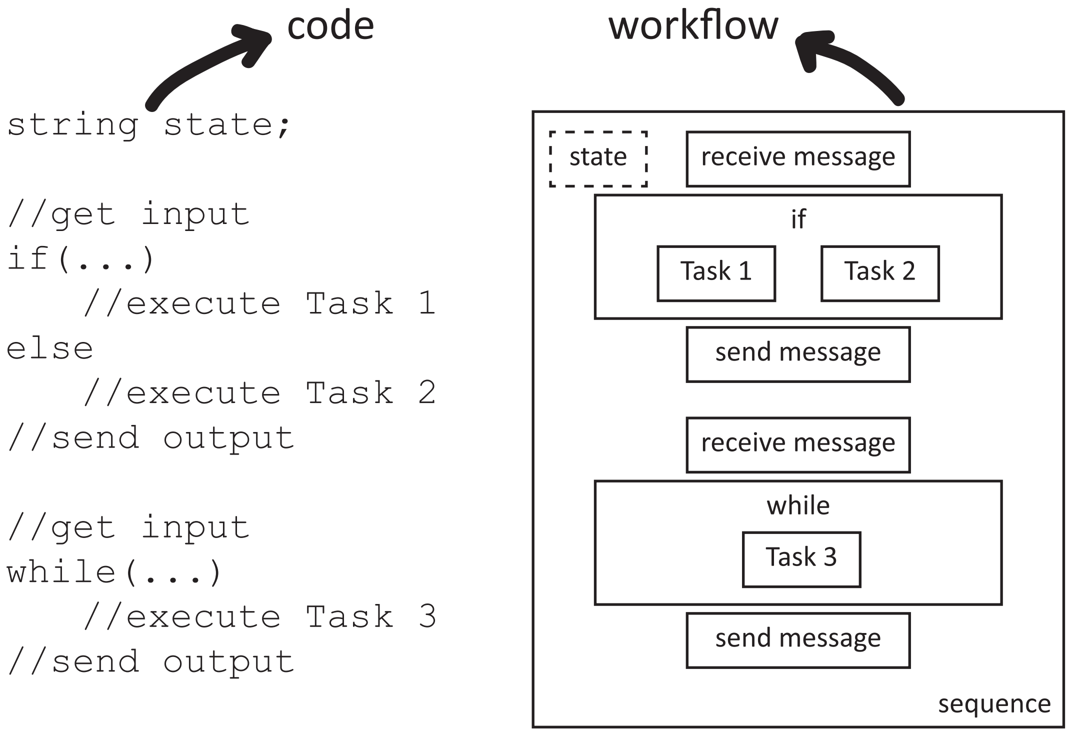
Microsoft's Windows Workflow Foundation (WF) provides capabilities for a descriptive programming experience. As a .NET[1] framework, it improves the way how to implement business logic. Approximately until 2012 WF was just part of .NET without an separate server. With Workflow Manager[2] 1.0 Microsoft introduced a server, which can host WF 4.5 definitions.

In order to simplify implementation of business logic in certain programs, a more declarative model has been created [25]. Microsoft introduced a new possibility of writing program code in .NET by describing it with a flow. Actually this flow can be described with characters in the form of source code as well. However, activities and arrows are visual objects, which can be easily understood by humans. Hence, implementing continuously changing business logic within programs with the help of workflows makes it possible to change certain flows when needed. Although adaptation of program code is possible as well, workflows are more comprehensible (cf. Figure 5.1). Since workflows provide a more abstract level of developing applications and services, additional advantages can be discovered in fields of scalability, automatic tracking and the usage of ordinary workflow tools like implementing a parallel split. Moreover, with WF it is possible to define state machine specifications that can be handled equally to the flow chart definitions.

Microsoft's integrated development environment (IDE) Visual Studio provides tools

[1] .NET Framework, http://www.microsoft.com/net

[2] Workflow Manager, http://msdn.microsoft.com/en-us/library/jj193528%28v=azure.10%29.aspx



```
string state;

//get input
if(...)
   //execute Task 1
else
   //execute Task 2
//send output

//get input
while(...)
   //execute Task 3
//send output
```

Figure 5.1: Comparison between ordinary Code and a Workflow (based on [25])

for visually creating workflows. Even though creation of business operations can be simplified, it is not meant to be used by end users. Preferably it is a usability improvement (cf. section 4.1) for maintaining business logic and simplifying programs by hiding complexity for the application specialists (cf. Figure 4.1).

5.1.1 Capabilities of dynamic Update

In WF there is the possibility to define several versions of a workflow alongside each other. Thus, if there exists workflow W with several versions, instances to made of the one or the other can be specified. Assuming there exist three versions of the workflow, W1.0, W1.1 and W2.0. Now, when instantiating a workflow the definition to use can be selected. If nothing is specified, the latest version[3] will be taken, which would be W2.0 in this case. By being aware of various version of a workflow, WF sets the cornerstone for the update of instances.

The feature of updating instances from one version to another is called dynamic update. Based on different identities of a workflow, a persisted instance can be updated. Since the status of an instance has to be kept in mind, the engine has to know how to get from one version to the other. Therefore a so called update map will be created, as illustrated in Figure 5.2 in order to provide the necessary information about the differences. In his presentation [42] about the future of WF, Ron Jacobs describes the

[3] What's new in WF45, http://msdn.microsoft.com/en-us/magazine/hh781025.aspx

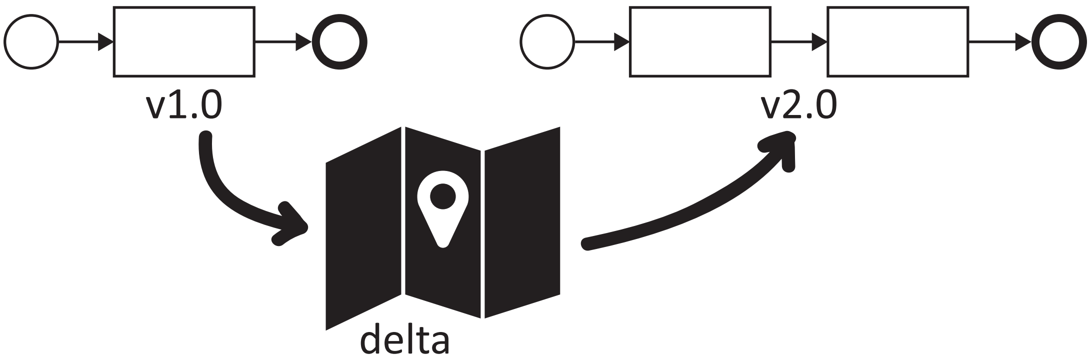


Figure 5.2: The Update Map to get from one Version to the other

update map as a guideline of how to get from one version to the other. The runtime takes this information and modifies the instance accordingly.

Long running instances are persisted in the database in order to be continued when necessary (e.g. requested input has reached the workflow). In case of urgent bug fixes or changing business requirements such persisted instances can be continued with the most recent workflow definition.

5.1.2 Limitations of dynamic Update

Dynamic update helps windows workflows being more flexible even though instances are already running. Nevertheless, there are limitations when it comes to the degree of flexibility. First of all, dynamic updates are just true for dynamically updating the definition in a later state. This means activities, which already have been executed cannot be changed (cf. section 4.6.3). Thus, if instances' progress is too far, some adaptations cannot be applied to them. In order to provide end users with the capability of dynamic update, it probably has to be wrapped into a user friendly program with an appropriate GUI.

5.1.3 Example Realization of dynamic Update

Technically speaking, the transformation of an instance from one version to the other is done within four steps[4].

1. prepare for update
2. make changes
3. create the update map
4. apply the update map

[4] Dynamic Update, http://msdn.microsoft.com/en-us/library/hh314052(v=vs.110).aspx

In order to generate the previously mentioned UpdateMap, the workflow definition to be updated needs to be prepared. The WF framework provides a class called DynamicUpdateServices, which contains static methods to be called. Step one and three rely on this class to first prepare the workflow definition and generating the UpdateMap after applying the necessary changes. Listing 1 illustrates how a newly created workflow can be prepared for an update.

Listing 1 Call on PrepareForUpdate()

```
// Create a new workflow
Activity workflow = new MyFlowV1();
// Prepare the workflow
DynamicUpdateServices.PrepareForUpdate(workflow);
```

Differences between the original workflow and the updated definition need to be determined by the framework. Therefore the first step identifies all objects of the workflow, that are necessary for the comparison between the old and the new version. After identifying all objects, the workflow is cloned and attached to the original definition. With now having a backup of the original workflow, the definition can be changed in order to implement the necessary adjustments, as stated in the second step.

Once changes has been made to the workflow definition, the UpdateMap can be generated by simply calling the appropriate method in the DynamicUpdateServices class (cf. Listing 2). In this case the map is returned to a variable and will be used afterwards to apply the UpdateMap. However, sometimes it is necessary to persist the map on the file system to be able to postpone the actual update of an instance.

Listing 2 Create the UpdateMap with DynamicUpdateServices class

```
// Create the UpdateMap and apply it to a variable
DynamicUpdateMap updateMap =
   DynamicUpdateServices.CreateUpdateMap(workflow);
```

Step four consists of several sub steps. First, the id of the persisted instance to be updated needs to be determined. Based on this id the respective instance can be loaded from the database (cf. Listing 3 at line 6). Second, the updated definition will be used to create a workflow application. In the last step, this workflow application can be used to load the old instance and update it to the new definition. Afterwards the workflow instance will be persisted back into the database by calling the unload method.

Listing 3 Apply the UpdateMap to a persisted Instance

```
// Get id of the persisted workflow to be updated
Guid id = GetPersistedWorkflowId();
// Get the persisted instance from the database
SqlWorkflowInstanceStore store =
   new SqlWorkflowInstanceStore("Server=.\\SQLEXPRESS");
WorkflowApplicationInstance instance =
   WorkflowApplication.GetInstance(id, store);

// Create a workflow application for applying the map
WorkflowApplication app =
   new WorkflowApplication(new MyFlowV2());
// Now load the instance and apply the UpdateMap
app.Load(instance, updateMap);
// Persist and unload the workflow instance
app.Unload();
```

5.2 camunda BPM platform 7.2

Compared to WF the camunda BPM platform[5] is a Java based framework for process automation. Furthermore camunda provides a holistic tool set for managing processes with tools like camunda Modeler, camunda Cockpit (monitoring and operations), camunda Cycle (synchronizing models) and a public API. In combination with the open source attitude the BPM platform makes it open for Java developers. BPMN is the preferred process definition language in camunda's BPM platform.

Although camunda provides a complete suite, the company still is keen on providing the possibility of embedding it into custom projects. The platform's flexibility is even visible in the versatile implementation scenarios of the framework. Regardless of whether providing camunda's process engine in a shared, container-managed environment within an application server, as a standalone remote server or implement the engine upon various cluster nodes for scalability and fail-over capabilities.

The framework is based on the Activiti[6] project, which has been forked by camunda for using BPMN integration in Java. With its many tools it makes the platform more than a framework.

5.2.1 Capabilities of Version Migration

Updating running instances in the camunda BPM platform is called version migration. Similar to WF, camunda is capable of allowing multiple versions of a process specification to exist. By default all new instances start with the latest version of the process

[5] camunda BPM User Guide, http://docs.camunda.org/7.2/guides/user-guide/

[6] Activiti Components, http://activiti.org/components.html

definition. However, this can be specified, when creating an instance. Process instances, which already exist at the time the process model is updated, will continue execution with the model version they were started. If yet the optimized model have to be applied instantly to all running instances, migration can be performed. Thus, according to camunda's documentation, simple migration scenarios are supported by using a certain class (cf. Listing 4).

From the programmatic point of view those simple scenarios can be extended. While camunda is thinking of providing more support for the version migration, they suggest two alternatives for migrating versions in a more sophisticated manner. In either way the current process instance can be terminated and a new one can be started. First alternative is to forward the state of the newly created instance to the state of the terminated one. Of course this is a highly error prone task because all kinds of preconditions have to be met when not resuming a created instance at the actual start point. Moreover transparency has to be ensured manually.
Instead of enhancing the overall version of the original process model, the other alternative is to provide a separate migration version. This migration version can be thought of an completely customized version, especially designed for the proper migration of an instance.

5.2.2 Limitations of Version Migration

Theoretically there are few limitations when considering the mentioned alternatives for migrating a process instance. However this has to be performed manually by an advanced programmer and process developer in order to bear in mind all possible error sources. Thus proper execution is a highly risky task, which needs to be tested sufficiently. Aside from that, no end user would be able to perform such a version migration.

Only simple migration can be executed with a provided class by the framework. However this possibility is not exposed in the public API because migration is considered an advanced topic, which indeed is quite obvious when thinking of things that can go wrong. So simple migration has limitations in order to be fulfilled without migration errors. Therefore new process definitions need to comprise all objects that have already been executed in the currently running instance, i.e. it depends on the state of the instance where all executed activities have to be still available.

Identically to WF, camunda's version migration is just available for persisted instances. BPMN provides the usage of so called boundary events [63]. Since camunda is relying on BPMN as a modeling language, those boundary events are limiting simple version migration. For that reason boundary events cannot be handled properly when being attached to the activity, the instance is waiting in. Thus, the signal boundary event on Task 1 in the process model of version 2 in Figure 5.3 would not be triggered. Because persisted instance is waiting in Task 1, the engine is not able to react to an event attached to the activity in this case. When reversing version 1 and 2 in the scenario of Figure 5.3, existing boundary event on Task 1 is not possible to be removed because it already existed when the instance has been persisted while waiting in Task 1.

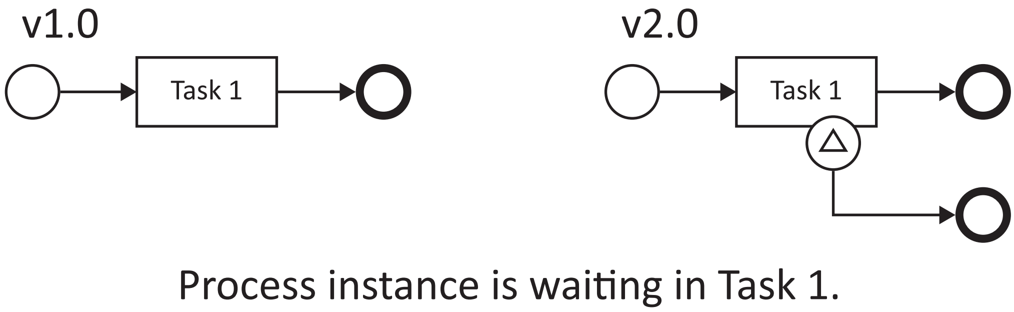


Figure 5.3: Ignored boundary Event on Task 1

5.2.3 Example Realization of Version Migration

Since updating instances is considered an advanced topic, there is no special service that does the migration like in WF. Simple migration can be accomplished by creating a certain command, which will be consumed by the CommandExecutor. This executor is provided by the process engine, as can be seen in Listing 4. How to migrate the new version will be defined by the class SetProcessDefinitionVersionCmd. The class specifies the appropriate command out of the instance id and the version respectively. Although version migration is theoretically possible, support for this feature is hardly given because the majority has to be done by the developer.

Listing 4 Migrate a Process Instance to a new Version

```
// Set command for instance 1a2b3c and version 2
SetProcessDefinitionVersionCmd command =
   new SetProcessDefinitionVersionCmd("1a2b3c", 2);

// Now execute the command
((ProcessEngineImpl) ProcessEngines.getDefaultProcessEngine())
   .getProcessEngineConfiguration()
   .getCommandExecutorTxRequired()
   .execute(command);
```

5.3 Oracle Fusion Middleware 12.1.3

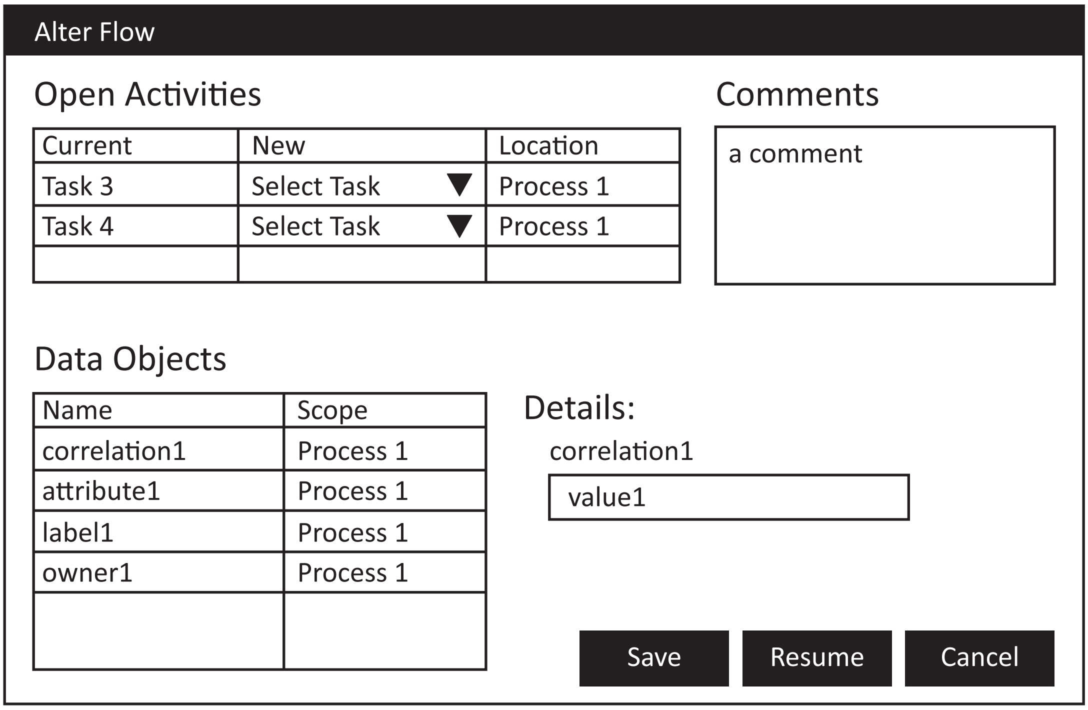
Oracle's Business Process Management Suite is part of the Oracle Fusion Middleware platform. It states itself as a business innovation platform for operation either on premises or in the cloud. The BPM Suite 12c[7] enables development and implementation of business applications with additional monitoring features based on the business process paradigm. It provides a bunch of tools for managing business applications in a more abstract way.

[7] Oracle BPM Suite 12c, http://www.oracle.com/us/technologies/bpm/suite/overview/index.html

In comparison to the previous two frameworks, Oracle provide an even more holistic approach with the aim of entirely manage business applications in BPMN. Different tools provide support for trained end users. Those tools give users the ability to manage their applications and operational work combined with several monitoring possibilities.

5.3.1 Capabilities of Instance Migration

With Oracle BPM Suite, users obtain the ability of managing running process instances in several ways[8]. The suite provides this ability through a GUI, which makes it more convenient for users, than programmatically alter processes. If a process is changed and redeployed, users can choose if corresponding instances should be kept for later migration. Instance migration can be managed by a separate graphical interface, which provides the user with information about current activities and its correlation to other processes. The GUI's main purpose is to alter the flow of an instance. Therefore all current activities are shown in the interface including a drop down for choosing the new activity to be the new current activity, i.e. the state of the process will be changed from being at one activity to another one. UX is increased by only showing activities, where the state can validly be changed to (cf. scenario in Figure 5.4).



Current	New	Location
Task 3	Select Task ▼	Process 1
Task 4	Select Task ▼	Process 1

a comment

Name	Scope
correlation1	Process 1
attribute1	Process 1
label1	Process 1
owner1	Process 1

Details:

correlation1

value1

Figure 5.4: Interface for altering the Flow of an Instance

Additionally, correlations can be reset in order to omit communication with certain other processes. Correlation represents indicators for different business processes,

[8] Modifying Running Instances with Oracle, http://docs.oracle.com/middleware/1213/bpm/bpm-user/bpmug_alt_flw_mig.htm

which exchanges messages among themselves. By defining such correlations, processes know with whom to communicate. Overall adjustment can be commented with custom text. The alter flow interface in Figure 5.4 shows three buttons where certain action can be applied. By taking the resuming action, changes will be applied to the instance and the instance will be resumed. In case of deciding to not instantly adjust the flow of the instance, changes can just be saved and the instance will be suspended. The new flow can be applied later on. Further, on changing the flow can be canceled.

Basically, Oracle differentiates between just altering the flow of an instance and altering the flow caused by a redeployment of a process definition. Beside few exceptions, the user interface for migrating an instance to a newly deployed specification looks the same as illustrated in Figure 5.4. So it is possible to tell the instance where to continue execution of activities, where an instance is waiting in, have been removed.

In order to define processes, Oracle provides support for different objects. Such objects for defining the process specification can either be ordinary BPMN components or other objects, Oracle is providing support for like BPEL and business rules components. However focus lays on the BPMN components. The Oracle BPM Suite is capable of automatically migrating instances if just new activities, tasks or data objects are added or data associations are changed.

5.3.2 Limitations of Instance Migration

Planning to change a process definition and applying it to running instances come with some restrictions by the use of certain objects. Thus, deletions of sub processes and gateways are not compatible with instance migration. Yet there is an exceptional case, because this does not count for exclusive gateways. Like camunda, Oracle has problems with BPMN's boundary elements as well, no matter whether adding or updating them. Sub processes could cause problems as well. Movement of activities into certain structures like sub processes and inside gateways leads to migration compatibility issues.

In a blog post Capgemini made a few suggestions for additional requirements [87] of Oracle's instance migration feature. The author criticizes, that some process executions are not clearly visible when alterations have been made for a large amount of processes. Because of that, he states some requirements from a controller's perspective. According to him in some cases the feature should not be allowed at all. Furthermore, for better control in risky situations the permission level should be adjustable in order to provide four-eyes principle for approval.

5.3.3 Example Realization of Instance Migration

Listing 5 Example of a Migration Definition with an Ant Task

```
<project name="instance-migration" basedir="." default="test">
   <import file="ant-bpm-instance-migration-lib.xml"/>
   <property file="locator.properties"/>
   <property name="reports.dir" value="test/output/reports"/>
   <property name="plan.dir" value="test/output/plans"/>

   <locatorConfig id="bpm.host" host="host1" port="44000"
      user="admin" password="admin"/>

   <target name="generate.report" depends="init">
      <compositeInstanceFilterDef id="all.instance"
         compositeDN="default/MigrateBetweenRevisions!1.0"/>
      <locatorSession configId="bpm.host">
         <migrateCompositeInstances filderId="all.instance"
            revision="2.0" outputFile="${reports.dir}/R1.xml"/>
      </locatorSession>
   </target>

   <target name="migrate.instances" depends="init">
      <compositeInstanceFilterDef id="all.instance"
         compositeDN="default/MigrateBetweenRevisions!1.0"/>
      <locatorSession configId="bpm.host">
         <migrateCompositeInstances filderId="all.instance"
            revision="2.0" migrationPlan="${plan.dir}/P1.xml"/>
      </locatorSession>
   </target>

   <target name="init">
      <mkdir dir="${reports.dir}"/>
   </target>

   <target name="clean">
      <delete dir="${reports.dir}" includes="**/*"/>
   </target>
</project>
```

When updating a process definition, corresponding instances change their states into »pending migration«. In order to raise these instances to the updated process specification, an instance can either be migrated »as is« or can be adjusted and being migrated based on these adjustments. In case an automatic migration is possible, no adjustments needs to be performed. Before the actual migration can be performed, Oracle provides analysis for migration. Thus, after redeploying an optimized version of the definition, migration of running instances may need to be performed on several instances. Basically, this is done within four steps and starts with the mentioned analysis.

1. discover which instances have to be migrated
2. run the migration feasibility report
3. create the migration plan
4. perform the migration

Technically, Ant Tasks[9] define the analysis and the migration. When looking at the example Ant Task in Listing 5, on line 10 properties how the migration report will be generated are specified. A filter can be specified, which instances should be considered for the feasibility report. The feasibility report provides information about which instances can be migrated automatically, which of them have to be adjusted and which instances cannot be migrated at all. As can be seen on line 24, the plan of how manually migrate instances is provided. Specification of how this can be achieved, is stored in an XML file, which will be defined by user through a GUI. Similar to the interface for an ordinary flow adjustment (cf. Figure 5.4), reassignment of removed activities can be realized.

5.4 IBM Business Process Manager 8.5.5

IBM provides a similar approach to Oracle's BPM platform. As a holistic BPM solution, IBM offers[10] tools for process design, process execution, process monitoring and features for optimizing business processes. In order to ensure advantages of BPM in an entire company, the suite is fully scalable in combination with a service oriented architecture (SOA). Furthermore, the IBM Business Process Manager offers content management solutions with basic case management features. According to their product flyer, ad hoc support including collaborative and social working capabilities are provided as well.

5.4.1 Capabilities of migrating running Instances

Since IBM's solution provides features for process optimization, the system is required to offer update mechanisms respectively. Long running business process definitions are likely to be improved during their execution and thus their corresponding instances are required to deal with updated models. The IBM Business Process Manager is capable of handling such requirements. Workflow models are preferably modeled in BPEL. Therefore IBM provides capabilities for adding and removing BPEL's basic activities (see [62] for BPEL specification) as well as modifying the activities' properties during

[9] Apache Ant, https://ant.apache.org/

[10] IBM BPM Manager Advanced, http://www-03.ibm.com/software/products/en/business-process-manager-advanced

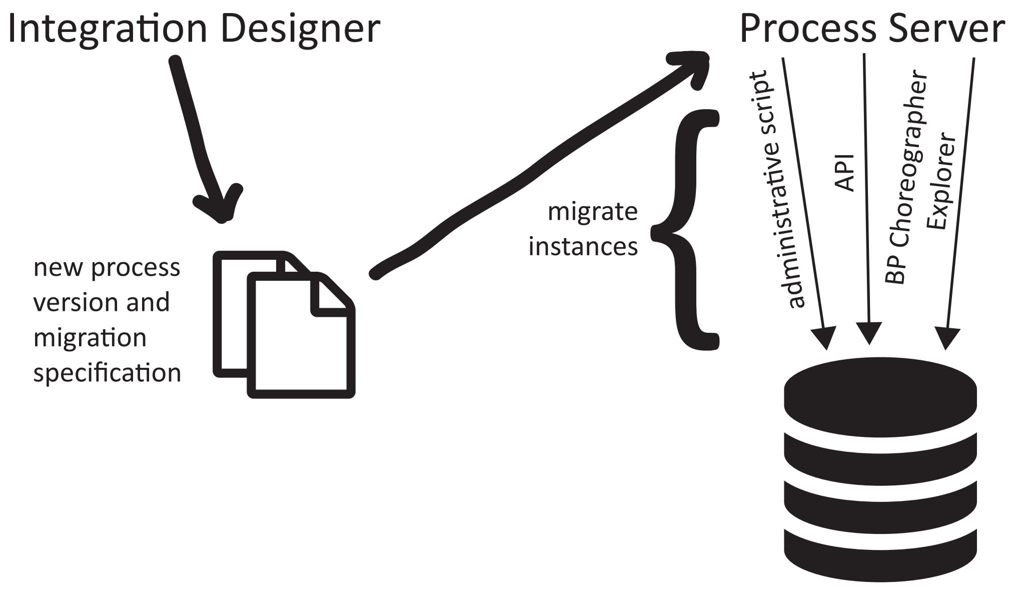


Figure 5.5: Steps needed for migrating running Instances

runtime. Basic activities are Assign, Empty, Human Task, Invoke, Java Snippet, Reply, Rethrow, Terminate, Throw and Wait. Moreover Fault Handlers and Event Handlers in general can be adjusted. Even Loops, Choices and Branches are supported to be allowed during runtime.

In order to migrate an instance, where its model has been altered, a migration specification is needed. Figure 5.5 shows how such a migration process is performed. After updating the process model, a migration specification will be created. With the assistance of this specification the process server migrates the instances. Which instances to migrate and which version of the workflow definition will be used can be specified in the migration specification. Regarding transparency of instance migration the system supports event tracking in combination with available version historization.

5.4.2 Limitations of migrating running Instances

In case that just properties of basic activities are adjusted, migration is not a problem. As soon as changes to the business logic are applied, several restrictions arise when instances should be updated. In general, regions of an instance, where adjustments have been made may not be entered otherwise migration will fail. More precisely, this is partly true for more sophisticated BPEL objects. Since Event Handlers are assigned to a certain scope in the process, this scope must not have been entered by the execution progress in order migration to succeed. This is the same for Loop and Choise elements. Branches, however, do not need to be after the execution state entirely. Just the changed region in the affected branch needs to be untouched by the execution. Beside restrictions in more sophisticated elements, following mechanisms are not supported in case of migration.

- compensation logic

- SQL snippets
- correlation
- custom properties
- administrative tasks
- removal or modifications of variables
- data map activities

5.4.3 Example Realization of migrating running Instances

As illustrated in Figure 5.5, a migration specification will be used in order to tell the process server how to handle running instances. After the new version of the process is created it can be deployed on the process server including the migration specification.

Listing 6 Command for Bulk Migration

```
install_root/bin/wsadmin.sh -f migrateProcessInstances.py
   [([-node node_name] -server server_name) | (-cluster cluster_name)]
   (-templateName template_name)
   (-sourceValidFromUTC timestamp)
   [(-targetValidFromUTC timestamp)]
   [(-slice slice_size)]
```

Instance migration can be accomplished in three different ways. By providing an administrative script several options help to successfully migrate running processes. Furthermore the platform exposed the migration capability to the API. Lastly, convenient migration is provided by the Business Process Choreographer Explorer, which is part of the solution.

Listing 7 Output when starting migrateProcessInstances Script

```
WASX7209I: Connected to process "server1" on node linuxNode01 using SOAP
    connector;
   The type of process is: UnManagedProcess WASX7303I: The following options
        are passed to the scripting environment and are available as
       arguments that are stored in the argv variable: "[-server,␣host1,␣-
       templateName,␣TestProcess1,␣-␣sourceValidFromUTC,␣2014-11-29T14:44:01
       ]"

The process instance migration is running on server 'host1' on node 'node1'.
     Please check the log files of the server to get information about the
    progress and results of the migration.
```

Basically, a migration specification is created by just defining the source and the target version of the migration. Additionally, differences can be shown by the system. Since a lot of instances can be referenced to a certain process model, migration of each instantiation might not be efficient. Therefore IBM provides the possibility of a bulk migration[11]. Migration in bulk can be performed with the help of an administrative script. Therefore IBM provides the script migrateProcessInstances.py. As can be seen in Listing 6, the server, the template and the start time needs to be specified.

After executing the command, just a start message will be displayed in the console. Like described in Listing 7 the actual progress and the result can be seen in the corresponding log file[12]. Obviously instances, which are in an end state (like finished, terminated, failed and compensated), are not to be migrated.

[11] Migrating BPEL Instances in Bulk, http://www-01.ibm.com/support/knowledgecenter/SSNKAY/com.ibm.wbpm.bpc.doc/topics/tadmin_instance_migrate.html

[12] Technical Migration Guide, http://www.ibm.com/developerworks/bpm/bpmjournal/1305_norelus/1305_norelus.html

6 Conclusion

Being innovative is the most crucial aspect for a lot of organizations in order to compete in the global market. Innovation depends on time, risk and most importantly, change. Changes cannot be avoided and therefore people have to live with it. However, if changes are useful and appreciated, depends on the viewing perspective. What if companies could decide to be the initiator for a change? Probably a lot of firms could gain an advantage out of being the initiator for certain changes. So utilizing changes instead of combating them sounds like the most appropriate choice. Thus, innovation indeed can help businesses to gain an advantage. Eventually, advantage can be defined by the difference of innovation between an organization and its competitors. Beside changes, time and risk influences innovation capabilities as well. Because competition is hard to handle without being innovative, companies are forced to take risks and trying new things. As a result, firms strive to minimize risk and still looking for innovations. For a state of the art organization, information technology plays a major role for administration and operation together with support throughout the entire institution. Since firms are managed by computer systems, such systems are required not just to support innovation but rather drive this kind of intentions. Development of specialized computer programs, swallows money and more importantly, time. That is the reason why companies have been looking for an abstract way to tell computer systems what they can help them with. Results, among others, are known as process modeling languages, state machine illustrations and rule based specifications of how describing computer programs in a more natural way. Hiding complexity by simplifying program specifications leads to faster implementation of such systems. Time is the third dependency innovation rely on. Thus, if technology is optimized to do things faster and additionally can enable new attempts of doing business, companies meet the preconditions for being an innovative company.

Computer systems are intended to solve repetitive tasks by maximizing speed and minimizing failures. As a result programs are static and have not been built to be flexible in the past. Nowadays IT changed its paradigms and is keen on providing flexible, lightweight and scalable solutions with decreasing maintenance effort. Process engines basically execute corresponding process specifications. Even those specifications need to be adaptable to some extent. Full featured solutions come in many shapes and sizes and two big ones have been mentioned in this work (cf. section 5.3 and 5.4) in case of their capability of instance adaptability. Those solutions provide support for case management (cf. section 3.5.3) as well as for traditional business process execution. Since case management introduced a new handling of unpredictable processes, this paradigm is meant to be operated by humans, i.e. knowledge workers. However, highly repetitive processes, which are automated need to be updated from time to time. Beside IBM and Oracle, other companies provide similar approaches for dealing with BPM in a flexible manner. Still there are limitations when it comes to instance migration in order to reference updated process model versions. And this is just considered the technical point of view, where usability has not been paid attention at all.

Although BPMN is a de facto standard when it comes to define business processes, its usage in some projects is questionable. As mentioned in section 2.2.1, execution of plain BPMN is hardly possible, which imply BPMN would just be a plain modeling notation. Because of the widespread usage and famous supporters of this notation systems with support for BPMN will not stop to exist. Alternative approaches are available although its maturity in all aspects of the system is not that clear. One example is S-BPM [31], which is based on state machines. As long as BPMN based systems are usable and are able to be refined, companies will use them. Nevertheless, when thinking of instance migration, BPMN is restricting features from being feasible. For example, attached events limit the capability of instance migration in PAIS as described in section 5.3.2. Actually, people just want an easy way to tell the computer system what to do and as already explained, easy in this case means that complexity is totally hidden. At least technological complexity should be hidden, because business requirements still can be sophisticated to some extent. Probably AI will help to reveal new possibilities to communicate with computers in an advanced way. In the best case humans tell the system what they want to be the outcome and the computer will try to accomplish the task. Basically, this has already been done by goal based automation, described in section 4.6.4. Yet maturity is far from being useful throughout the entire process solution.

For now, flexibility to a certain extent is feasible, however for whom it is actually realizable has to be discussed. Certain adaptations still require the knowledge and experience of a specialist, which has not the kind of understanding of the business side as the knowledge worker would have. Although there are GUIs, where user experience needs to be improved. Several companies revealed mobile apps for dealing with business requirements and thus with the advantage of using a mobile device for capturing data, change settings and several other things, which could help the company aligning their processes to the real world. In the end it is just the level of abstraction compared with the convenience to use and work with those system, developed as a holistic solution.

Bibliography

[1] *Cooperative Information Systems, 1999. CoopIS '99. Proceedings. 1999 IFCIS International Conference on*, 1999.

[2] *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*. ACM, Sanibel Island, Florida, USA, 2003.

[3] *Proceedings of the 14th international conference on World Wide Web*. ACM, Chiba, Japan, 2005.

[4] *Proceedings of the 2007 ACM symposium on Applied computing*. ACM, Seoul, Korea, 2007.

[5] *27th ACM Symposium On Applied Computing (SAC'12), 9th Enterprise Engineering Track (EE'12)*. ACM Press, 2012.

[6] *Proceedings of the 16th European Conference on Pattern Languages of Programs*. ACM, Irsee, Germany, 2012.

[7] *Proceedings of the 1st International Workshop on Model-Driven Engineering for High Performance and CLoud computing*. ACM, Innsbruck, Austria, 2012.

[8] *Enterprise Distributed Object Computing Conference (EDOC), 2013 17th IEEE International*, 2013.

[9] Jacky Akoka, Mokrane Bouzeghoub, Isabelle Comyn-Wattiau, and Elisabeth Métais, editors. *Conceptual Modeling — ER '99*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999.

[10] Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors. *Business Process Management*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007.

[11] Dean Anderson and Linda S. Ackerman-Anderson. *Beyond change management: How to achieve breakthrough results through conscious change leadership*. Pfeiffer essential resources for training and HR professionals. Pfeiffer, San Francisso, 2nd ed edition, 2010.

[12] KrzysztofR Apt, editor. *Logics and Models of Concurrent Systems*. NATO ASI Series. Springer Berlin Heidelberg, 1985.

[13] Clara Ayora, Victoria Torres, Manfred Reichert, Barbara Weber, and Vicente Pelechano. Towards run-time flexibility for process families: Open issues and research challenges. In Marcello La Rosa and Pnina Soffer, editors, *Business Process Management Workshops*, volume 132 of *Lecture Notes in Business Information Processing*, pages 477–488. Springer Berlin Heidelberg, 2013.

[14] Barbara Staudt Lerner, Stefan Christov, Leon J. Osterweil, Reda Bendraou,

Udo Kannengiesser, and Alexander Wise. Exception handling patterns for process modeling. *IEEE Transactions on Software Engineering*, 36(2):162–183, 2010.

[15] Benjamin Nagel, Christian Gerth, Enes Yigitbas, Fabian Christ, and Gregor Engels. Model-driven specification of adaptive cloud-based systems. In *Proceedings of the 1st International Workshop on Model-Driven Engineering for High Performance and CLoud computing*, pages 1–6. ACM, Innsbruck, Austria, 2012.

[16] Birgit Zimmermann and Markus Doehring. Patterns for flexible bpmn workflows. In *Proceedings of the 16th European Conference on Pattern Languages of Programs*, pages 1–9. ACM, Irsee, Germany, 2012.

[17] Ralph Bobrik, Manfred Reichert, and Thomas Bauer. View-based process visualization. In Gustavo Alonso, Peter Dadam, and Michael Rosemann, editors, *Business Process Management*, volume 4714 of *Lecture Notes in Computer Science*, pages 88–95. Springer Berlin Heidelberg, 2007.

[18] Egon Börger. Approaches to modeling business processes: A critical analysis of bpmn, workflow patterns and yawl. *Software and System Modeling*, 11(3):305–318, 2012.

[19] David L. Bradford and W. Warner Burke. *Reinventing organization development: New approaches to change in organizations*. Pfeiffer essential resources for training and HR professionals. Pfeiffer, San Francisco, 2005.

[20] Dietmar Bruckner. *Probabilistic Models in Building Automation: Recognizing Scenarios with Statistical Methods*. PhD thesis, Vienna University of Technology, Vienna, 2007.

[21] Gino Brunetti, Thomas Feld, Lutz Heuser, Joachim Schnitter, and Christian Webel, editors. *Future business software: Current trends in business software development*. Progress in IS. Springer, 2014.

[22] Michele Cantara. Leverage dynamic and ad hoc processes now for business adaptability: Garnter inc. (G00247368), 2013.

[23] Michele Cantara, Teresa Jones, and Kimihiko Iijima. Marketscope for business process management platform as a service: Gartner inc. (G00248339), 2013.

[24] F. Casati, S. Ceri, B. Pernici, and G. Pozzi. Workflow evolution. *Data & Knowledge Engineering*, 24(3):211–238, 1998.

[25] David Chappell. The workflow way: Understanding windows workflow foundation, 2009.

[26] Asit Dan, Frédéric Gittler, and Farouk Toumani, editors. *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010.

[27] Rob Dunie, Michele Cantara, Janelle B. Hill, and Samantha Searle. Cool vendors in business process management, 2014: Gartner, inc. (G00258622), 2014.

[28] Fabio Casati, Stefano Ceri, Stefano Paraboschi, and Guiseppe Pozzi. Specification and implementation of exceptions in workflow management systems. *ACM*

Trans. Database Syst., 24(3):405–451, 1999.

[29] Thomas Feld and Michael Hoffmann. Process on demand: Planning and control of adaptive business processes. In Gino Brunetti, Thomas Feld, Lutz Heuser, Joachim Schnitter, and Christian Webel, editors, *Future business software*, Progress in IS, pages 55–66. Springer, 2014.

[30] Herbert Fischer and Josef Schneeberger, editors. *S-BPM ONE - Running Processes*. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2013.

[31] Albert Fleischmann. *Subject-oriented business process management*. Springer, Berlin and New York, 2012.

[32] Albert Fleischmann, Werner Schmidt, Robert Singer, and Detlef Seese, editors. *Subject-Oriented Business Process Management*. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2011.

[33] Robert Gottanka and Nils Meyer. Modelasyougo: (re-) design of s-bpm process models during execution time. In Christian Stary, editor, *S-BPM ONE – Scientific Research*, volume 104 of *Lecture Notes in Business Information Processing*, pages 91–105. Springer Berlin Heidelberg, 2012.

[34] Guangxin Yang. Process inheritance and instance modification. In *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, pages 229–238. ACM, Sanibel Island, Florida, USA, 2003.

[35] D. Harel and A. Pnueli. On the development of reactive systems. In KrzysztofR Apt, editor, *Logics and Models of Concurrent Systems*, volume 13 of *NATO ASI Series*, pages 477–498. Springer Berlin Heidelberg, 1985.

[36] D. A. Harmon, R. J. Kezys, B. H. Portier, and M. G. Steele. Business process management, configuration and execution, 2014.

[37] Paul Harmon. *Business process change: A guide for business managers and BPM and six sigma professionals*. Business Process Change. Elsevier, Amsterdam and Boston, 2nd ed edition, 2007.

[38] Hernâni Mourão and Pedro Antunes. Supporting effective unexpected exceptions handling in workflow management systems. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 1242–1249. ACM, Seoul, Korea, 2007.

[39] Janelle B. Hill. The case for case management solutions: Gartner inc. (G00235833), 2012.

[40] IEEE, editor. *Commerce and Enterprise Computing (CEC), 2010 IEEE 12th Conference on*, 2010.

[41] Stefan Jablonski. On the complementarity of workllow management and business process modeling. *SIGOIS Bulletin, August 1995/Vol. 16, No. 1*, pages 33–38, 1995.

[42] Ron Jacobs. Windows workflow foundation futures: Pdc10, 2010-10-29.

[43] Janelle B. Hill, Kenneth Chin, Rob Dunie. Critical capabilities for case management frameworks: Gartner inc. (G00250141), 2014.

[44] John Jeston and Johan Nelis. *Business Process Management: Practical Guidelines to successful Implementations*. Routledge, Abingdon, Oxon, 3rd ed edition, 2014.

[45] Ajay Khanna. Managing unpredictability using bpm for adaptive case management: Oracle corporation. 2013.

[46] Kwangmyeong Kim, Injun Choi, and Chulsoon Park. A rule-based approach to proactive exception handling in business processes. *Expert Systems with Applications*, 38(1):394–409, 2011.

[47] Johannes Kotremba, Stefan Raß, and Robert Singer. Distributed business processes - a framework for modeling and execution. *CoRR*, abs/1309.3126, 2013.

[48] M. Kradolfer and A. Geppert. Dynamic workflow schema evolution based on workflow type versioning and workflow migration. In *Cooperative Information Systems, 1999. CoopIS '99. Proceedings. 1999 IFCIS International Conference on*, pages 104–114, 1999.

[49] Markus Kradolfer, Andreas Geppert, and KlausR. Dittrich. Workflow specification in trams. In Jacky Akoka, Mokrane Bouzeghoub, Isabelle Comyn-Wattiau, and Elisabeth Métais, editors, *Conceptual Modeling — ER '99*, volume 1728 of *Lecture Notes in Computer Science*, pages 263–278. Springer Berlin Heidelberg, 1999.

[50] Matthias Kurz, Albert Fleischmann, Matthias Lederer, and Sebastian Huber. Planning for the unexpected: Exception handling and bpm. In Herbert Fischer and Josef Schneeberger, editors, *S-BPM ONE - Running Processes*, volume 360 of *Communications in Computer and Information Science*, pages 123–149. Springer Berlin Heidelberg, 2013.

[51] Marcello La Rosa and Pnina Soffer, editors. *Business Process Management Workshops*. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2013.

[52] Jurij Laznik and Matjaz B. Juric. Context aware exception handling in business process execution language. *Information and Software Technology*, 55(10):1751–1766, 2013.

[53] B. S. Lerner, S. Christov, L. J. Osterweil, R. Bendraou, U. Kannengiesser, and A. Wise. Exception handling patterns for process modeling. *Software Engineering, IEEE Transactions on*, 36(2):162–183, 2010.

[54] Guohua Liu, Xi Liu, Haihuan Qin, Jianwen Su, Zhimin Yan, and Liang Zhang. Automated realization of business workflow specification.

[55] Guohua Liu, Xi Liu, Haihuan Qin, Jianwen Su, Zhimin Yan, and Liang Zhang. Automated realization of business workflow specification. In Asit Dan, Frédéric Gittler, and Farouk Toumani, editors, *Service-Oriented Computing. ICSOC/ServiceWave 2009 Workshops*, volume 6275 of *Lecture Notes in Computer Science*, pages 96–108. Springer Berlin Heidelberg, 2010.

[56] Guohua Liu, Xi Liu, Qin Haihuan, Jianwen Su, Zhimin Yan, and Liang Zhang. Auto-

mated realization of business workflow specification. 2009.

[57] Manfred Reichert, Jens Kolb, Ralph Bobrik, and Thomas Bauer. Enabling personalized visualization of large business processes through parameterizable views. In *27th ACM Symposium On Applied Computing (SAC'12), 9th Enterprise Engineering Track (EE'12)*, pages 1653–1660. ACM Press, 2012.

[58] Marco Brambilla, Stefano Ceri, Sara Comai, and Christina Tziviskou. Exception handling in workflow-driven web applications. In *Proceedings of the 14th international conference on World Wide Web*, pages 170–179. ACM, Chiba, Japan, 2005.

[59] Microsoft Corp. Workflow manager, 2013.

[60] R. R. Mukkamala, T. Hildebrandt, and T. Slaats. Towards trustworthy adaptive case management with dynamic condition response graphs. In *Enterprise Distributed Object Computing Conference (EDOC), 2013 17th IEEE International*, pages 127–136, 2013.

[61] Robert Müller, Ulrike Greiner, and Erhard Rahm. Agentwork: a workflow system supporting rule-based workflow adaptation. *Data & Knowledge Engineering*, 51(2):223–256, 2004.

[62] OASIS WSBPEL TC. Web services business process execution language version 2.0: Oasis standard, 2007.

[63] Object Management Group. Business process model and notation: Version 2.0: Specification, 2011.

[64] Thomas J. Olbrich. Why we need to re-think current bpm research issues. In Albert Fleischmann, Werner Schmidt, Robert Singer, and Detlef Seese, editors, *Subject-Oriented Business Process Management*, volume 138 of *Communications in Computer and Information Science*, pages 209–215. Springer Berlin Heidelberg, 2011.

[65] Fatima Otori, Marco Peise, Sundararaman Venkataraman, Jingjing Wei, Zhi Qiu Xie, and Yan Zhao Zhang. *Version-to-Version Migration to IBM WebSphere Dynamic Process Edition V7*. IBM redbooks. IBM Redbooks, [online-ausg.] edition, 2010.

[66] Manfred Reichert, Steve Rechtenbach, Alena Hallerbach, and Thomas Bauer. Extending a business process modeling tool with process configuration facilities: The provop demonstrator. 2009.

[67] Manfred Reichert and Barbara Weber. *Enabling flexibility in process-aware information systems: Challenges, methods, technologies*. Springer, Heidelberg and New York, 2012.

[68] S. Rinderle-Ma and M. Reichert. Advanced migration strategies for adaptive process management systems. In IEEE, editor, *Commerce and Enterprise Computing (CEC), 2010 IEEE 12th Conference on*, pages 56–63, 2010.

[69] Bruce Robertson. Hype cycle for business process management, 2013: Gartner, inc. (G00246931), 2013.

[70] Thomas Rothschädl. Ad-hoc adaption of subject-oriented business processes at runtime to support organizational learning. In Christian Stary, editor, *S-BPM ONE – Scientific Research*, volume 104 of *Lecture Notes in Business Information Processing*, pages 22–32. Springer Berlin Heidelberg, 2012.

[71] Colin Rudd and Vernon Lloyd. *Service design: IT Service Management*. The Stationery Office Ltd, London, 2007.

[72] Nicholas Charles Russell. *Foundations of Process-Aware Information Systems*. PhD thesis, Queensland University of Technology, Brisbane, Australia, 2007.

[73] Nick Russell, Ter Hofstede, Arthur H. M., David Edmond, and van der Aalst, Wil M. P. Workflow data patterns. *QUT Technical report*, (FIT-TR-2004-01), 2004.

[74] Nick Russell, Ter Hofstede, Arthur H. M., David Edmond, and van der Aalst, Wil M. P. *Workflow resource patterns*, volume WP 127 of *BETA Working Paper Series*. Eindhoven, 2004.

[75] Nick Russell, Ter Hofstede, Arthur H. M., van der Aalst, Wil M. P., and Nataliya Mulyar. Workflow control-flow patterns: A revised view. *BPM Center Report BPM-06-22, BPMcenter.org*, 2006.

[76] Nick Russell, van der Aalst, Wil M. P., and Ter Hofstede, Arthur H. M. Exception handling patterns in process-aware information systems. *BPM Center Report BPM-06-04, BPMcenter.org*, 2006.

[77] W. Roy Schulte and Michele Cantara. Practical ways to make business operations more intelligent: Gartner inc. (G00263913), 2014.

[78] Victor Senkevich. Adaptive case management (acm): as an instrument of social management of business processes within a company, 2013.

[79] Christian Stary, editor. *S-BPM ONE – Scientific Research*. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2012.

[80] Christian Stary, editor. *S-BPM ONE – Scientific Research*. Lecture Notes in Business Information Processing. Springer Berlin Heidelberg, 2012.

[81] Keith Swenson. *Taming the unpredictable Real World Adaptive Case Management*. Excellence in practice series. Future Strategies, London, 2011.

[82] Keith D. Swenson. *Mastering the unpredictable: How adaptive case management will revolutionize the way that knowledge workers get things done*. Meghan-Kiffer Press, Tampa, Fla, 2010.

[83] Ter Hofstede, Arthur H. M., van der Aalst, Wil M. P., Michael Adams, and Nick Russel. *Modern business process automation: YAWL and its support environment*. Springer, Berlin, 2010.

[84] Maria-Cruz Valiente, Elena Garcia-Barriocanal, and Miguel-Angel Sicilia. Applying an ontology approach to it service management for business-it integration. *Knowledge-Based Systems*, 28(0):76–87, 2012.

[85] van der Aalst, Wil M. P. Patterns and xpdl: A critical evaluation of the xml process

definition language. *BPM Center Report BPM-03-09, BPMcenter. org*, 2003.

[86] van der Aalst, Wil M. P. Business process management: A comprehensive survey. *ISRN Software Engineering*, 2013:37, 2013.

[87] van der Kamp, Martijn. Oracle bpm: Suspend and alter process: Capgemini oracle blog, 2012.

[88] Barbara Weber, Manfred Reichert, Jan Mendling, and Hajo A. Reijers. Refactoring large process model repositories. *Computers in Industry*, 62(5):467–486, 2011.

[89] Stephen A. White. Process modeling notations and workflow patterns: Ibm corp. 2004.

[90] M. T. Wynn, Verbeek, H. M. W., van der Aalst, W. M. P., Ter Hofstede, A. H. M., and D. Edmond. *Business Process Verification - Finally a Reality!* PhD thesis, Queensland University of Technology and Eindhoven University of Technology, 2007.

[91] Yan Ye, Zhibin Jiang, Xiaodi Diao, and Gang Du. Extended event–condition–action rules and fuzzy petri nets based exception handling for workflow management. *Expert Systems with Applications*, 38(9):10847–10861, 2011.

[92] Andrew Zhu. *Microsoft Windows Workflow Foundation 4.0 Cookbook*. Packt Publishing, Olton, Birmingham, GBR, 2010.

[93] Andrew. Zhu. *Microsoft Windows workflow foundation 4.0 cookbook over 70 recipes with hands-on, ready-to-implement solutions for authoring workflows*. Packt Pub, Birmingham [U.K.], 2010.