VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

QUERY-DRIVEN CONTEXT-AWARE WORKFLOW

DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE — Ing. et Ing. VOJTĚCH MATES
AUTHOR

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

DOTAZEM ŘÍZENÉ KONTEXTOVĚ ORIENTOVANÉ WORKFLOW

QUERY-DRIVEN CONTEXT-AWARE WORKFLOW

DISERTAČNÍ PRÁCE
PHD THESIS

AUTOR PRÁCE — Ing. et Ing. VOJTĚCH MATES
AUTHOR

VEDOUCÍ PRÁCE — Prof. Ing. TOMÁŠ HRUŠKA, CSc.
SUPERVISOR

BRNO 2015

Abstrakt
Tato práce pojednává o návrhu architektury systému pro řízení podnikových procesů, který je schopen reagovat i na měnící se kontext již běžících procesů. Architektura systému je navržena tak, aby systém byl schopen měnit jednotlivé fragmenty procesů dle aktuálního kontextu. Tímto zajišťuje i řízení přeplánování procesu, přičemž využívá k rozhodování optimální variant historická data pořízená vlastním během systému.

Abstract
The thesis presents an architecture design of the business process management system which is able to adapt to the changing context of already running business processes. The architecture of a system is designed to change particular fragments of the processes according to the current context. Thus it enables managing and rescheduling of processes while using historical data in the search for optimal variant of process performance.

Klíčová slova
Systém pro řízení podnikových procesů, adaptabilní architektura, orchestrace podnikových procesů

Keywords
Business process management system, adaptable architecture, business process orchestration

Citace
Vojtěch Mates: Query-driven Context-Aware Workflow, disertační práce, Brno, FIT VUT v Brně, 2015

Query-driven Context-Aware Workflow

Prohlášení

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně pod vedením Prof. Ing. Tomáše Hrušky, CSc., a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

....
Vojtěch Mates
31. srpna 2015

Poděkování

Chtěl bych poděkovat vedoucímu práce, Prof. Ing. Tomáši Hruškovi, CSc., za vedení této práce. Díky patří také ostatním kolegům z Ústavu informačních systémů, FIT VUT v Brně, za jejich cenné připomínky k této práci. Poděkování také patří mé rodině a všem přátelům za jejich podporu.

Výzkum v rámci této práce byl podpořen grantem MPO Czech Republic MPO TIP FR-TI3 039.

Acknowledgement

First and foremost, I would like to thank my supervisor, Prof. Ing. Tomáš Hruška, CSc., for his guidance related to this thesis. I would also like to thank all the team of the Department of Information Systems, at BUT FIT, for their valuable suggestions regarding my work. Finally, I would like to thank my family and all friends, for their encouragement and support over the years.

The research in this thesis has been supported by MPO Czech Republic MPO TIP FR-TI3 039.

Contents

List of Figures

Chapter 1

Introduction

Companies tend to be more process-oriented in their management style. One of advantages in this approach, compared to the hierarchical model, can be better sharing resources in a company which can influence overall productivity as well. The other reason why the process-oriented approach can be more suitable is a level of standardization. If we have standardized processes, it will be easier to manage and control results of performing activities and it will also increase productivity, especially for repeatable processes. Standardization also decreases the error rate. There is very limited deviation from the process definition, because it is desirable to define each step in advance.

However, there are issues where using well-defined processes are difficult to implement, because it is not simple to predict all possible performance variants (e.g. medical treatment) or it is necessary to change a process very often because of the changing environment of the process. Thus, there is a contradiction. It is necessary to manage a process as precisely as possible, however, it is often unknown what the process will look like and what occurrences will arise. Thus, it can be helpful if a system could be more adaptive at least on context-related changes while keeping the maximum advantages (e.g. monitoring, decision control, etc.) from current systems.

Therefore, the goal of thesis is *a description on how to add context-aware adaptability to process-aware systems, which is not well supported by current systems*. The process orchestration of the system is driven by query which defines the set of resulted goals and constraints related to the context for the process performing. The process orchestration is based on a process database which contains the subprocesses. Each subprocess leads to the particular result. The process orchestration driven by query is composed of the configuration which leads to fulfilling the final goals while choosing the best option based on heuristic analysis provided by a concept based on *Dynamic Objects*.

A high level of flexibility is enabled due to a mixture of several different business process modelling approaches. It enables a natural combination of a declarative and imperative way in one process, briefly described in chapter 4. The higher connection layer uses a goal-based approach which can connect several subprocesses in one coherent process definition. The created relation between goals and business process subprocess provides connection between strategical and operational management. Therefore, redefining objectives and their constraints influences immediately the operation level of the management by a dynamic process orchestration.

The whole approach can dynamically change part of the process without affecting the main business logic structure. Each possible process orchestration is valid because it uses subprocesses and makes a temporal connection based on relations between the producer and consumer. However, changing the parts effects the performance of the process and thus it is desired to choose the best possible configuration for each specific context-depended situation. The architecture design also enables a fault-tolerant mechanism similar to routing algorithm in a network by enabling switching the process variants in runtime in case the part of the process cannot be performed at the moment.

1.1 Structure of the Thesis

The thesis is based mainly on research papers: *Modification of Workflow Management System Focused on Human* [32], which describes briefly the problem of missing support of context in current Workflow Management Systems, the idea of usage Dynamic Profiles for users is mentioned in *Using Workflow Management System for Analysis Based on Properties of Resources*[31].

The papers *Process Mining in Manufacturing Company for Predictions and Planning*[48] and *Analysis Resource Performance and its Application* [49] present the importance of context which was supported by research in a manufacturing company, where an hypothesis supported context relevance was tested on real historic data covering several million records. The datamining methods were used for analysing of real data.

The paper *Context-Based Adaptation of Process Definition* [33] describes the idea of architecture of business process management system which is able to change the overall process logic according to a changing context in runtime.

The application of context-adaptable process definition was also presented in the area of services-oriented architecture in paper *Modelling of Context-adaptable Business Process and their implementation as Service-Oriented Architecture* [30].

The remainder of this thesis is divided into *State of the Art*, *Dynamic Objects in Systems*, *Context-Aware Adaptation using Dynamic Objects* and *Future Work and Conclusion*.

State of the Art In State of the Art, a brief introduction to many related terms and areas to business process management system is provided, especially from the point of view modelling and optimizing which has a relationship to the design of the proposed context-aware architecture which keeps the process orchestration optimized using a configurable model driven by query.

Dynamic Objects In Dynamic Objects, I introduce the problem and concept which can help find and automatically create context-aware *dynamic objects* in system which will be used for representation of context models. The concept of managing *dynamic objects* is used for heuristic analysis for context-aware management system. The desirable process configuration is driven by query defining the objectives and related constrains.

Adaptation of Context-Aware Workflow In this chapter, the proposal of context-aware workflow architecture is presented, the architecture supports multi-modelling approaches and presents a query-driven context-aware approach of managing. The architecture provides dynamic process orchestration which is driven primarily by goals and context declared by user query. The process orchestration is able to dynamically adapt according to the changing context environment and current goals.

Future Work and Conclusion In this chapter, I briefly summarise my approach and emphasise the most important results.

Chapter 2

Process Aware Information Systems

Organizations have identified business agility as a competitive advantage to address business trends like increasing product and service variability or faster time-to-market, and to ensure business IT alignment. In particular, improving the efficiency and quality of their business processes and optimizing their interactions with partners and customers have become crucial success factors for companies.

In this context, a business process can be defined as a set of one or more connected activities which collectively realize a particular business goal [59]. Following this trend, a new generation of information system has been created – process-ware information systems.

An example of the system can be the business process management system (suite) which enables business process configuration, exception handling, process model refactoring, etc. The important issue also deals with flexibility in process-aware systems [51].

2.1 The Business Process Management Suite

The business process management suite can be described as a tool where users can create the common functionality provided by information system in general. This can be done by defining business process models which describing the logic of applications. The approach based on business process models has several advantages:

- *It saves cost for system maintenance.* The process definition represents universal interface for the whole system. Therefore, it can be easier, compared to the common information system, replace, delete or add the new part without destroying other parts of the system. The basic functionality of the system can be visualized (e.g. process models in *process designer tool*) which is also helpful while making changes in the system.

- *It is cost saving for information system development.* The company Bizagi promotes up to 2/3 faster development (more about the product in section 2.1.4) because of faster prototyping of the new system which often denotes the faster acceptance by customers. The unique functionality over the set of data can be often scripted in other programming languages such as JavaScript, etc.

Creating an information system based on process models can enable several extra feature sets:

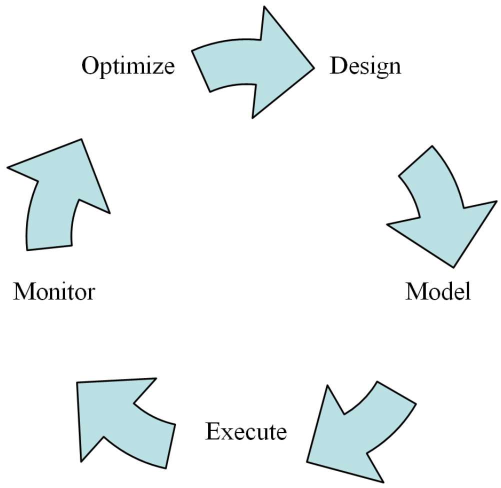

Figure 2.1: Business process management life cycle

- It is possible to generate up-to-date documentation describing the system functionality.
- It is easier to maintain, because a lot of common changes can be done by users simply by reconfiguring the system.
- All events in the system can be logged and the performance data can be used for further analysis leading to *business process optimizing* (see chapter 7).
- The business process management suites often provide the generating Graphic User Interface based on the task definitions.

2.1.1 Business Process Management Cycle

The implementation of particular business processes often follows the business process management cycle which is repeated with each iteration of *business process model* deployment[55]:

- *Design* – The design phase usually covers the collecting of information which should lead to providing the functionality desired by the specification of use cases. The design phase can result in the analysis of requirements for the business process.
- *Model* – It is possible to transform the specification from the design phase to the process model and other related components. The simulation tools are often used for supporting the designing of business process models without possible side effects .
- *Execute* – It is possible to execute the process model into a business process instance. The system starts the interaction with particular assigned users to *work items*, web services, etc.
- *Monitor* – During execution, data captured during a performance is stored, thus it can be used used for further analysis in the Optimize phase.
- *Optimize* – It is desired to compare the captured data to expected results. Based on the analysis of business process performances, new requirements for the next design phase of BPM cycle are created.

2.1.2 Relationship Between Business Process, Process and Instance

The *business process* is a specification of the behaviour related to part of the system. It is possible to extract a business process model as a simplification of business process behaviour which can be defined within the business process management suite. The model should adopt these changes in order to keep the model up-to-date because the real business process often changes.

The defining business process is one of the important steps in the standardization process. The process-oriented management helps in better sharing resources due to the standardization of processes. It is also easier to evaluate the process performance. The performance of the business processes is often connected with *key performance indicators* (KPIs) that will be described in section 7.1.

Process, in the context of business process management suite, often means the *business process model*. The *business process model* contains actions and rules defining the routing between actions. The *instance of business process model* is the particular execution based on logic defined in the *business process model*. In other words, the business process management can be described as the template for the *instance of the business process model*. The *instance of business process model* has to follow the rules defined in the *business process model*.

2.1.3 Components of Business Process Management Suite

The business process management suite can contain many components. However, the basic architecture concept of most business process management suites comes from *Workflow Reference Model* [8] [9]. However, there are also some extensions which are not covered by *workflow reference model*. The often used software components used in business process management system will be briefly introduced.

- *Process Designer Tool* – The tool provides creating a process definition using graphical notation in most cases. The definition of a business process model is the result of *Process Designer Tool*.
- *Workflow Engine* – The core component enabling executing the business process definition and managing the control flow within all *process instances* derived from *business process definition*.
- *Worklist Handler* – The workflow engine often creates a queue of tasks which are desired to be performed. The component *Worklist Handler* takes care about managing each *work item* (e.g. assigning the related resource to task).
- *User GUI Generator* – The component creates GUI (graphical user interface) based on the model of the *business process definition*.
- *Form Builder* – The Form Builder is the component specialized for generating form-based GUI providing user CRUD operations (Create, Read, Update, Delete) over data related to the task.
- *Databinding Component* – The component is used for binding data objects (providing application data) to *Form Builder*.
- *Management Console* – The component is used as an administration tool for managing processes (e.g. monitoring, starting, terminating processes, etc).

- *Reporting Component* – The component is used for creating reports based on the performance of processes.
- *Simulation Component* – The simulation component is used for simulation of processes in the modelling phase of the business process. It is often tightly connected to *Reporting Component*.
- *Resource Manager* – The *resource manager* provides managing records related to resources used in business processes.
- *Components for integration with other systems* – The business process management has to be connected often to other systems, mostly via web service interfaces. However, current information systems can support, for example ODATA interface (CRUD (Create, Read, Update, Delete) interface with other systems currently supporting SAP, Microsoft Dynamics, etc.

2.1.4 Selected Business Process Management Suite Solutions

Currently, there are several business process management suite products. The full specification can be found in many cases on the website of each company. However, some brief remarks related to each product will be presented.

Bizagi Process Management Suite

The Bizagi Process Management Suite covers the full functionality of business process management suites. The company, Bizagi, is also initiated in language standardization of BPMN (specification of notation in [39]).

I believe Bizagi has one of the best user interfaces and there is a lot of configuration options which can be set via the Bizagi Process Management Suite environment. Bizagi started dealing with the context-dependency management issue, however, it is not fully developed. The description of the system can be found in [5].

Oracle Business Process Management Suite

The Oracle Business Process Management Suite provides all necessary functionality; however, the graphic user interface is not as much intuitive as Bizagi's, but it supports accessing even to *process designer tool* via its website.

The process analytics can provide a lot of configurable options. Oracle uses BPMN as the modelling language. More information about the functionality of the product can be found in [40].

Microsoft Dynamics, Workflow Foundation, Nirex Workflow, Microsoft Sharepoint

The business process management suite solution can be provided by Microsoft. However, it is very tightly connected to several Microsoft technologies.

Microsoft Dynamic CRM can be used for modelling tasks and as the *Form Builder*; however, it does not directly support process modelling. Therefore, it is possible to use the combination with Workflow Foundation which is a primary software developer framework or with, for example, the Nirex Workflow. Microsoft Sharepoint is often used as the universal storage for unstructured data.

SAP

The SAP solution is quite popular for its robustness. However, the changes in the system can be very difficult and expensive.

The process modelling tool is ARIS, which is quite user friendly, however the part related to process deployment is not so intuitive as in for example Bizagi Process Management Suite. There is a significant dependency on a software development company, even for small changes in systems.

Kissflow

Kissflow is the cloud-based business process management suite solution. The different business model enables it to provide a quite expensive system as a service for a low fix payment. However, the system contains the basic functionality compared to others such as Oracle Business Process Management Suite or Bizagi.

Therefore, it is possible for it to be used for small-size companies. There is a support for integration with OneDrive and Google Apps which is not typically supported by other vendors. More information about the Kissflow functionality can be found in Kissflow User Manual [26].

YAWL

The business process management system has an academic background. It is primarily based on YAWL language which is an XML based language. However, it also supports Declare (which is described in section 6.4). The system is available to download on its website.

I see the main benefit that the system is well-documented because it is based on academic research. The support of the declarative approach of modelling is an another advantage; however, there is a lack of support for customization which is needed in case of commercial usage. It is based primarily on Java technologies, the installation package also contains web server Tomcat. More information about YAWL design can be found in [18]. The user guide is also available [13].

2.1.5 Business Process Management Suite Implementation

The implementation of the business process management suite has a very similar process compared to regular information system.

The mapping of a process is very difficult in the initial phase of the software analysis. However, the process structures are often mapped in some level of abstraction, at least in documentation related to ISO. The mapping of an existing system is also very complicated problem while there can be a problem with documentation. At best, there are implemented interfaces of web services which are well-documented.

After the initial software analysis, the designing of the business process is launched in cooperation with business process engineers. The prototypes, based on process definition validated by process engineers, are created. The advantage of the business process management suite consists in fast validation with user requirements. In parallel, the binding existing data sources is managed.

The potential cause of the problem lies in cases that people are not aware of process orientation, even in cases where the organization is already process-oriented. The documentation is not often up-to-date. The problem with integration with other low-documented systems is the same topic similar to regular information system development.

2.1.6 Summary and Discussion

The most current systems uses the imperative approach of business process modelling (e.g. BPMN). However, YAWL also supports the declarative approach by Declare, which is an important issue to handle a loosely specified business process. Each solution can be suitable to different business environments.

Kissflow matches most requirements of small businesses and the cloud-based business model also supports this market orientation.

The Bizagi Suite is still provided as the desktop application, however, in my opinion, the user configuration setting is at the best level of user experience.

The YAWL solution supports some features which others still do not support (e.g. declarative and imperative approach in one environment).

The SAP and Microsoft solutions are tightly connected to their software environment; however they are accessible via their own APIs and via ODATA in order work with data. The Oracle based solution seems to cover all useful functions and the process designer is accessible via web interface.

Most business process solutions are originated from Workflow Management System Architecture, see chapter 3.

In my opinion, the ideal business process management system should be accessible via web-based interface (universal access interface supported by many devices). It should be able to handle at least declarative and in imperative approaches of business modelling within one coherent *workflow engine*, because each modelling approach is more suitable for different kinds of processes.

It should provide context-sensitive management, because it adds possibility to handle each case more individually while managing automatically by defined rules.

The business process management system should be designed as a distributed system. In other words, business processes could be distributed via several nodes which could be easily added, thus state of all instances should be distributed in cloud-based storage. This feature enables the way of dealing with performance issues. Low performance can cause that only the core business processes are included in the system and the rest is done outside of the system so the simulation of change and the following reengineering of the whole system is much more difficult.

Chapter 3

Workflow Management System

This thesis discusses the problem of context-aware business process management systems based on the workflow management system architecture. Thus, in this chapter the workflow management system (WfMS) architecture will be briefly presented.

At the beginning, it should be mentioned why it can be useful to use the *workflow management system* (WfMS), for description of the workflow (see [59]). An important motivation for using WfMS is for supporting a process-oriented management style, where processes are more emphasized than the hierarchy in organization.

This attitude provides a better utilization of resources in organization. The *workflow management system* is a tool used for automation of the business process management. It can manage a mutual cooperation of the *WfMS participants* (including human and software interaction) in order to perform a *business process*.

The main purpose of WfMS lies in the coordination of resources according to the process definition which is usually created by the company management. Making changes in a business process is easier, because the process definition is not hard-coded into the system, thus the company can be more adaptable to changing conditions.

The workflow management system also allows for easier business process re-engineering as another of its functions is to provide valuable audit data (in addition to managing and monitoring of the processes) which can later be used for analysis. Information extracted from the audit data can also be used for better managing as will be illustrated below in the chapter. The architecture of the workflow management system will be described very briefly using of the reference model created by the Workflow Management Coalition [9].

In the part related to *Dynamic Objects*, some extension in WfMS architecture will be presented. These extensions can improve the system on the whole in particular areas (e.g. there will be a description of how to provide dynamic updating of certain properties of participants of WfMS using WfMS). The solution lies especially in setting up the *dynamic profiles* of *WfMS participants* and changes in the process definition.

3.1 Basic Components of Worfklow Management System

WfMS provides functions for managing processes according to *process definition*, assignment resources to tasks in *business processes*, *monitoring*, *auditing*, etc. WfMS should consist of *process definition tool*, *workflow management engine*, *worklist handler* and *user interface* [9].

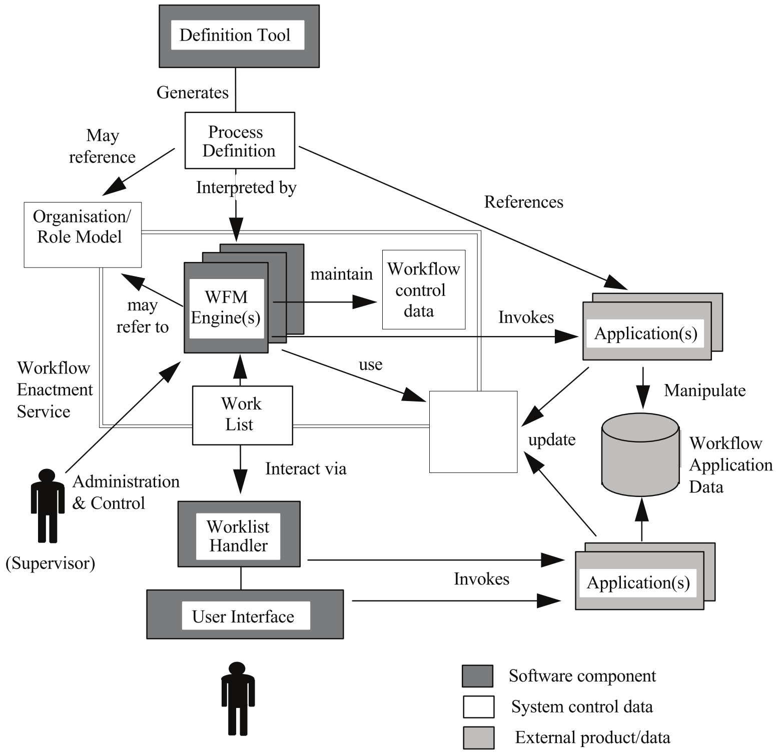

Figure 3.1: Architecture of the workflow management system [8]

The creation of the process definition (for example in BPMN, see [39]) is the first part of the process deployment in the *workflow management system*. This is usually done by the *process definition tool*. A process described by the WFDL (Workflow Definition Language) [7] is an output produced by this tool. XPDL (XML Process Definition Language) and BPEL (Business Process Execution Language) are usually used as WFDL. The description and the comparison of languages for the description of the process can be found in [42] for the modelling and transformation of the workflow (see [17]).

A *process definition* should include information about tasks such as who can perform individual tasks (mostly the role-based approach), and information about routing between the tasks. The business process execution logic can be described as routing that will be described in more detail in the following text.

The architecture of the WfMS will be described very briefly by means of the reference model created by the Workflow Management Coalition.

According to [8], the workflow management system should consist of:

- process definition tool,
- workflow management engine,
- worklist handler,
- user interface.

Instances (i.e. particular processes) originate from the *process definition* (described *business process model*). The *process definition* is forwarded to the *workflow management engine* where it can be launched as a *process instance*. *Worklists* are created by running particular *process instances* according to the *process definitions* with specific data. For instance during the execution, applications or people can update a workflow's relevant data. *Worklist handlers* then assign tasks to resources, for example people or software (mostly web services) after launching a process instance.

A *WfMS participant* is an object in the workflow management system taking part in a process instance (usually people or a web service). The user (typical *WfMS participant*) communicates with the system via several user interfaces (e.g. email, application interfaces, etc). All information about the performances of particular processes can be stored for monitoring and analysis purposes.

The WfMS produces data that can be used as input data to dynamic profiles of certain parts of the process definition. More information about *workflow management system* architecture can be found in [9].

3.2 Summary and Discussion

More information about WfMS architecture can be found in [9], and related terminology in [7]. More information about business process management in general, which uses WfMS as a tool can be found in [59].

The WfMS architecture is a base architecture of another current business process management suite (see chapter 2.1.4). There are also some extensions provided by vendors, but the basic architecture still remains.

However, I would add extensions which would be useful:

- *Worflow control data* and the *Process definitions* belongs together, especially if we would like to use support simulation based on history logs, or very useful for other analysis (e.g. see chapter about Process Mining 7.6).
- The architecture is not context-aware, which I believe is very important issue and the relevance was also shown in Case Study made in a manufacturing company (see chapter 10). I believe supporting the context will lead to a new generation of the system. It can be observed by first context support in Bizagi software (see the manual [5]).
- The architecture is not supporting dynamic process orchestration which I believe will be the next important issue, especially if the business process management system will be used. Because there are a lot of problems with merging different versions of business processes (see conformance checking [36]), because the current situation lacks support for distributed business process architecture.

Chapter 4

Approaches in Business Process Modelling

The aim of this chapter is to introduce concepts used for *business process modelling*. There are several approaches on how to model the business process. The most currently used approach is based on defining rules in an imperative way, thus every possible way on how to navigate in process is defined. One of advantages related to this approach is mapping to the formalism Petri Nets which enables the possibility to bind a certain pattern in language to a pattern in Petri Net.

Transforming languages to Petri Nets enables us some opportunities to use a wide variety of methods currently used on Petri Nets. As an example of this modelling approach, we can mention languages: BPMN, BPEL, XPDL, YAWL (see chapter 6).

However, there is also a different approach using a declarative way in modelling workflow in *business processes*. This approach is based on adding constraints to a set of tasks. Thus, the *business process definition* describes the constraints are in the process rather than enumerate all ways to navigate in the process. We can mention language Declare as an example (see [47]).

Because the set of tasks is countable, it is possible to transform from a declarative to an imperative way. However, because the declarative way of modelling is often used for *loosely specified process* (see section 4.1), the transformed model will be probably larger in many cases of transformation.

However, there are more perspectives in process modelling related to specifically used cases and so there will be more approaches to handle different points of view.

4.1 Process Categories

There are several categories of the processes, which can be found in an organization. The processes from the categories mentioned below also have a specific relation to modelling approaches. Each modelling approach has a specific usage and it is also supported by a different type of software. The categories are described from the level of freedom in the process point of view.

- *Ad hoc processes* – These kinds of processes are typically created in runtime. The usage of this kind of system is useful when everything has to be logged. However, it is impossible to predict how the processes should exactly behave in design time. This category of *business* *processes* is related to the *case-based approach*. This system is not used as much as systems based on prespecified processes.

- *Prespecified processes* – The prespecified processes are designed in detail during the design time. It provides full control over process managing. On the other hand, there is a problem with flexibility in order to adapt and control processes that are not well-specified in design time. The typical area of usage are production processes or administrative processes which have to be well secured. The prespecified processes are supported by the most used *business process management suites*. The prespecified processes are related to the *rules-based approach*.

- *Loosely specified processes* – The loosely specified processes are situated in between the prespecified and ad hoc processes. Thus, they need some level of management control, however, the processes do not have to be specified in every possible detail. The declarative approach is typically used for this kind of situation. The declarative approach defines constrains rather than the exact routing in process. The typical area of usage is the medical treatment [51].

4.2 Case-based Approach

The process was originally created for a repeated sequence of actions and the important aim was to standardize the process in order to keep a low error rate and good productivity. However, the concept was transferred from production level where the processes were almost invariant, which causes problems with flexibility issues related to reactions to unexpected events. Thus, the case-based approach was designed.

This approach was designed especially for ad hoc processes that have almost no structure and, therefore, is not easy to control and evaluate them. The *case-based approach* enables us to create new *process definition* for each *process instance* manually almost without limit. It is often designed in runtime. It is used for *loosely specified process*, however, there is sometimes a huge possibility to change even the created action which cannot be desired.

The creating of the process manually is also time consuming. The *case-based approach* is used much less than the *rule-based approach*.

4.3 Rule-based Approach

The rule-based approach is the most used approach, especially in the imperative form (defining each possible state and the route how to get to all states). There is good control over the process management and, therefore, bypassing rules is difficult which is critical, especially for banks or insurance companies.

I would recommend this approach at a later stage of company evolution where there is documentation and the processes are already set up. In the early stage, it is very difficult to create the rules which cover all possible cases in business process.

4.4 Data-centric Approach

This approach is based on routing using data attributes. The typical usage is adding routing rules navigating to the next task based on the values of certain attributes [10].

The typical use case of the approach can be seen in an administrative environment. Many users fill the forms in parallel that generates actions of filling in other forms. Thus, the synchronization is of users is done via an editing database. This can also be described as a specific form of the event-based approach where events are launched when a certain data attribute set is set up.

4.5 Subject-oriented Business Process Management

Subject orientation focuses on interactive perspective. It is form of a communication based on the views of actors. There are at least two subjects which are sending messages for each other. Each subject has an internal behaviour which is described as the control flow between internal states.

The basic abilities of each subject are:

- sending a message,
- receiving a message,
- making an unobservable action.

The advantage of this approach is a good level of flexibility, because it focuses primarily on the message-based approach. However, this approach is still not widely used. I believe that the most-suitable usage is for the lowest level of process abstraction, where high level of flexibility is welcome. The approach is based on *CCS Calculus* (calculus of communicating systems). The modelling concept is not difficult to use, it has only five basic symbols: *Subject*, *Message*, *Function*, *Send*, *Receive*.

Because, it is primarily designed as message-based, it is very flexible in the way of adding new subjects or messages in runtime. The messaging system also enables easier implementation as the distributed system. Each subject could be possible store on different node of system (see in [12]).

4.5.1 Subject Behaviour Diagram

With the SBD diagram, representing process steps from the actor's point of view, contributions can be identified based on subjects. Most often in strategic optimization projects, not all actors need to be examined in detail.

Instead of examining or explaining the entire process, the contributions of one single subject can be used to show strategy-relevant actions. Furthermore, limiting the matrix to the SBD of a single subject yields a smaller matrix with fewer entries.

4.5.2 Subject Interaction Diagram

In the Subject Interaction Diagram (SID), each message needs to be arranged on a unique horizontal level. Since both case studies in this contribution are using the SBD-based matrix, the modelling approach using SID will not be explained.

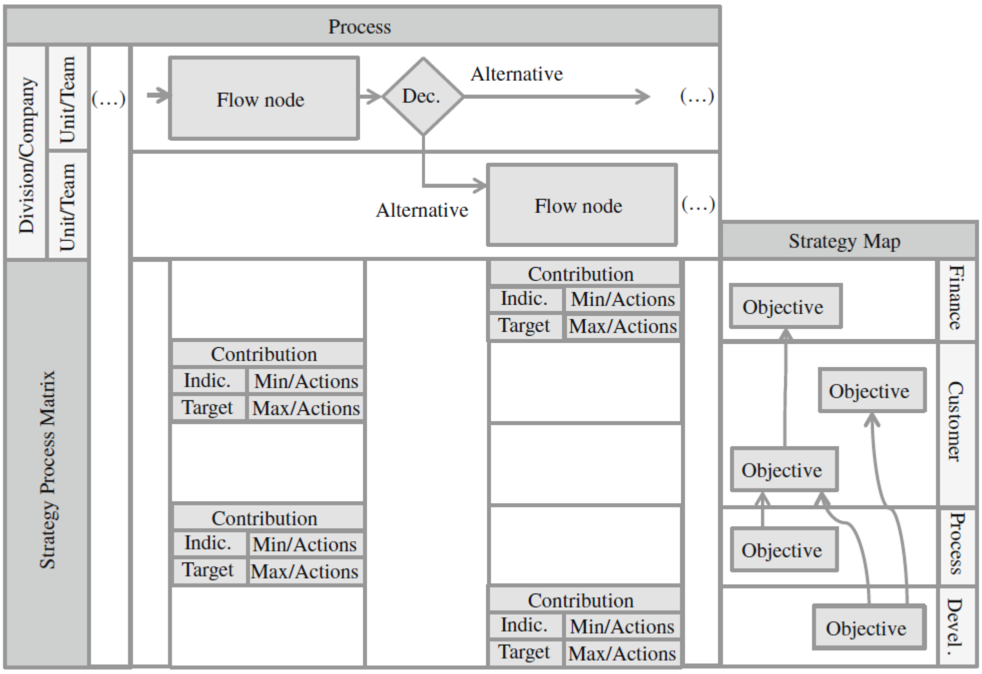

Figure 4.1: S-BPM and Strategy Map a book [12])

4.6 Strategy-oriented Business Process Modelling

The *strategy-oriented business process modelling* approach provides a linkage between models and strategy. The approach combines strategy objectives with the *business process models*. The Strategy Process Matrix is often the result of the *strategy-oriented business process modelling*.

4.6.1 Balanced Scorecard

The Balanced Scorecard is often used as a measurement framework of activities in organizations, especially from a strategic point of view.

This tool groups organization strategic objectives in perspectives:

- Finance (focusing on cash flow, sales),
- Customer (focusing on value for customers),
- Process (monitoring performance of business processes),
- Development (e.g. training and learning people).

The strategy map shows these objectives within dependencies.

4.7 Relation between Project Management and Process Management

The project is often described as unique configuration of tasks, resources and other conditions [21]. However, in certain circumstances, it is possible to see a similarity between project and process instance, because a lot of projects have to follow some rules based on the type of projects or organization. The main advantage of project management compared to process management is that it better supports changes in the runtime of projects.

However, project management is much more time-consuming comparing to process management. Thus, there is a need to add some features to process management in order to enable change management in similar levels of project management.

Thus, it is possible to model the framework of rules as a set of processes which allocate unique resources. The project usually has to deal with three basic dimensions:

- *Cost* – The budget influences primarily the scope of the project. The higher budget can also speed up the project time.
- *Scope* – The scope (the amount of work which has to be done) is directly related to cost and time dimension.
- *Time* – If there is a lot of time, it is possible to make some savings in costs.

The management activities based on triple constrains are discussed in [6]. The project usually adapts to changing conditions, for example, the detection of deviant behaviour (e.g. delay) means rescheduling. The rescheduling is usually done according to some priorities.

If a specific context is added to the process definition, then the managing of the processes is much closer to the project-related flexibility issues. Project planning is usually a more dynamic methodology that is well-designed to handle changes, because there is an assumption there is high probability there will be a change of plan in the process. However, the *business process* was designed rather as tool for standardization, the issues related to availability or changing the version of process in runtime are quite isolated area of process design.

Thus project management has rescheduling as an expected feature; however, a managing is highly dependent on skills of the manager. The manager has a lot of freedom to deal with a project compared to someone who is responsible for managing a process instance, where the basic structure of actions in process was designed. There are also some experiences with previous performances of the process. Thus, there is a demand to use the process structure as much as possible, because it is a much more effective way on how to achieve a goal.

The other problem of project management can be that the project manager is sometimes not completely aware of other activities which influence his projects. Another disadvantage can be a high-dependency of skills manager. The ideal project manager should evaluate a huge amount of information such detailed knowledge of skills of all persons related to the projects, perfect knowledge of every task in his projects and he has to track every step in projects. In other words, he has to be fully aware of the complete context of the projects.

Thus, I believe adding context and adaptability to the process management is a very important issue which should be worth dealing with.

4.8 Summary and Discussion

There are several approaches which have been presented so far, each of them has specific attitude to model business process. I believe that the diversity of so many approaches shows that the issue is still open.

The basic goal is the proper distribution of certain task to particular users; however there is a problem with predicting the future context. Also, all occurrences which will happen are not known. There is also a problem to evaluate if the process was successful just after its performance which is much easier than the evaluation in design time (before process happens).

It is also possible to make a transformation between certain approaches in order to be supportable by one engine, the transformation is not often easy, thus the processes which are not completely prespecified are often maintained by humans outside the system or the complicated process is handled as the one complex task which is often known only to the person who is performing it.

If there is something out of the system then it is difficult to control it. The problem is also related with substitutability of the resources because the process activities are not captured (the real atomic activities are outside the system).

Thus, I believe the system should support at least several approaches because each modelling approach is more suitable for a different process type.

I would suggest there should also be a tactical level alignment with an operational level of management. The alignment is often made only in design time or in business process reengineering which prevents fast adaptation. The alignment has been show in [12], however the redesign has not been dynamic.

I recommend that top level orchestration should be managed by goals and current environment which influence goals as well. The specific procedure on how to perform some very specific objective is not important from the strategical point of view. The strategic approach is more declarative than of an imperative nature. For example, I would expect the considerations below.

In such case I have a demand for making 5000 products, should I take it? The goal *5000 products* has a direct transformation to processes. I expect several possible expectations on the market: I will have to decrease the price in order to keep the longterm strategic advantage. I will have to be fast enough in order to make our customers happy. I will make our services more complex in order to satisfy our customers. Which parts of the process should be outsourced and which parts of the process should stay in house all the time or which parts of the process should be changed according to the current context of the situation?

Each modification of the goals means redesigning the whole process structure in order to fulfill the goals. The easiest way on how to make the change in the organization is changing process orchestration because from the operation level of view, almost no significant changes happen.

There are many events that can significantly influence the result of a process. Unexpected events are caused by changes in the environment of the process that are not covered by business rules. Thus, it is also necessary to deal with the context of a process. Since the environment of a process is variable, it is usually necessary to change the logic of the process. This is usually done by replacing one part of the process by another one that produces adequate results.

For example, if the resource inputs of a subprocess are temporary unavailable then a need will arise to change the logic structure (e.g., by outsourcing the particular part of the process). The process execution is also limited by certain constraints such as costs, time or success rate. Thus, the logic of the process should also be adapted for possible changes in these attributes that have an influence on the choice of variants of the process.

The presented solution in this thesis uses a *workflow management system* to collect data from previous process performances. At the same time, dynamic profiles of the processes are created from the collected data. The workflow management system thus works as a measurement tool.

There are more approaches for adaptive managing systems. The challenge is to provide flexibility and offer process support at the same time [51]. Another presented solution is the combination a rule-based, case-based, and agent-based approach [41]. My approach generates a structured process definition which can adapt according to the process context. Furthermore, the internal context variables are updated in runtime due to the existing workflow management system.

My approach is a hybrid among the declarative approach, rule-based approach and goal-based approach with the ability to monitor itself and use the information to optimize process performance by the user's preferences defined by a user query.

The approach is also a combination of the data-driven [10] and control flow approaches. It merges subprocess definitions using interfaces based on product compatibility. The algorithm uses dynamic profiles which describe historical behaviour of processes (e.g., time, success rate, costs) to choose the optimal way to reach the goal of a process. In order to create the dynamic profiles of processes, it would be almost impossible without automatic tracking of processes by the workflow management system.

The next important step is to build a product dependency tree. Its purpose is to map all possible ways to create the desired product. A route in the product dependency tree is used to build the final structure of a modified process. The product dependency tree represents a product hierarchy. In other words, it reflects which income products are needed to produce the desired product.

My method has several advantages. It allows switching over to another way of producing the product according to changing conditions. This can be done due to the declarative approach. Within this approach, it is necessary to define the final product and a set of conditions that are related to the context of the process. By using databases of processes that include process definitions and dynamic characteristics, it is possible to build new processes according to specific conditions. Thus, the process can be modified according to the current context (e.g., costs, time, success rate, resource problems, etc.). It enables us to have a great advantage because information about the context of a process is often not available before runtime.

Product substitutability is what makes my solution so flexible. The substitutability means that within a group of products needed for creating a final product, one product can replace another product so that the target product characteristics are not changed.

My approach is a hybrid of the rule-based and goal-based approaches. It deals with contexts using a top-down declarative approach using context variables. The process definition uses a structured definition of subprocesses and combines them according to the current context. The basic idea can be compared to a real-time decision process because managers compare the possible ways of reaching certain goals using the currently available process. If something important happens, then the process has to be rescheduled.

In the *case-based approach*, it is possible to adapt a process according to the current context, but each kind of change has to be defined, for example, for price changes, each supplier can use a different order process. Overloading due to continuous change can eliminate the advantage of case-based decisions.

The part Dynamic Adaption According to the Dynamic Objects aims to show how to optimize the process definition by using data captured by the workflow management system for managing *dynamic profiles* of processes, and how to use the declarative skill for *context-based online adaptation*.

Chapter 5

Business Process Modelling

In this section, we will focus on the level of the process model and their instances. At the beginning, elements and their relations necessary for the model will be defined. Therefore, the meta-model for creating business process models will be introduced.

5.1 Process metamodel, process model, process instance

There are several levels of abstraction in the modelling of business processes. At the highest level of abstraction, the semantics of language notation for creating process definition has to be created. After that, it is possible to use the notation based on a *metamodel* to create a specific *business process model* which describes specific rules and task for all performances of that business process model. The specific performance of the the process model is called *instance*. The specific performance requires then an assignment of the particular resources. The relation between meta-model, model and instance is depicted in Figure 5.1.

5.1.1 Business Process Metamodel

The basic concepts used for business process modelling is depicted in Figure 5.2. The base element of the *process model* is composed from an element *Node* and oriented arc *Edge*. Element *Node* can be implemented as *Activity Model*, *Event Model*, *Gateway Model* while constrained with two as the

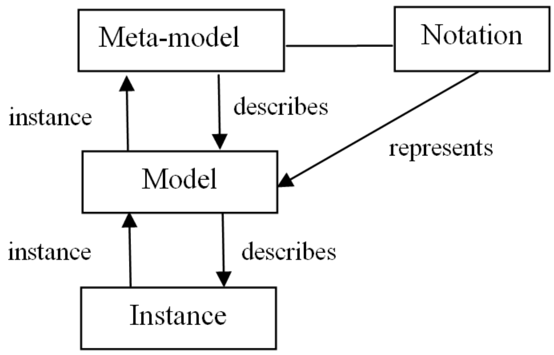

Figure 5.1: Relation between process metamodel, process model, process instance [59]).

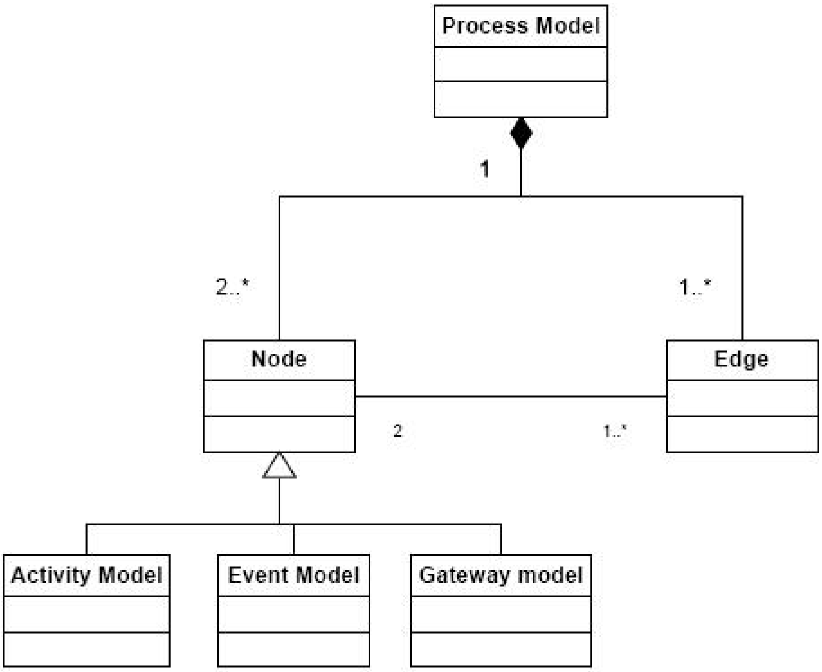

Figure 5.2: Business process meta-model (the example is adopted from [59]).

minimum number of node occurrences. Each edge represents the relation between nodes. Each edge connects just two nodes and represents their sequence order.

The *Activity Model* represents a basic work item which is supposed to be executed during the process performance. The node has just one incoming edge and one outgoing edge. The *Event model* represents the particular process state, which is significant for its performance.

The process beginning is represented by start event. The *Event Model* can be placed inside the process model.

The *Gateway Model* represents the flow control. The *Gateway node* is labeled as split node while it has one incoming edge and more outgoing edges. The split node is used in cases when the performing of the process can continue with several different activities while one or more has to be selected.

The function description will be explained later in the section. The *Gateway node* is join node while it has more incoming edges and just one outgoing edge, therefore, it represents joining several branches into one.

Business process meta-model can be defined as follows[59]:

Definition 5.1.1 (Process meta-model)**.** Assume that C is set of control flow constructs. $P = (N, E, type)$ is a process model where:

- N is the set of nodes with statement $N = N_A \cup N_E \cup N_G$, where N_A represents the set of activity models, N_E represents the set of event models and N_G represents the set of gateway models. These sets are disjunct with each other.
- E represents the set of edges such as $E \subseteq N \times N$, representing the control flow.
- type: $N_G \rightarrow C$ assigns one of the control flow patters to each gateway.

After setting the definition of the process, it is possible to represent the definition of the instance [59].

Definition 5.1.2 (Process instance)**.** Assume PI is a set of process instances, therefore a process instance $i = (Ei, <_i) \in PI$ based on a process model $P = (N, E, type)$ is defined by a partially ordered set of events E_i such as:

- E_i represents events of all activity instances $j \in AI$, such as $model(j) \in N$,
- $<_i$ is an event ordering in E_i satisfying the event ordering in all activity instances, in each gateway instances and event ordering between activities is restricted to constraints defined by process model P.

5.2 Workflow patterns

There are several different languages used for process modelling, therefore, there is a need for a transformation from one language to another. Thus, there was a need for creating a universal layer for transformation. The workflow patterns are describing widely used process fragment constructions modelled in Petri Nets. It enables the comparing of several languages in the expressiveness point of view and it also maps one notation to the notation of another language. The workflow control pattern and comparing expressiveness of selected business process notations were described in [42].

5.2.1 Control flow patterns

Control flow patterns represent basic control flow constructions during process performance that can be found in almost every commonly used process modelling language.

The basic patterns represent:

- Sequence execution,
- Selection (XOR split/XOR join),
- Parallel execution (AND split/AND join),
- Iterations.

More detail can be provided in [57][59]. The models of particular activities will be labeled in capitals letters A, B, C and the instances will be labeled in lowercase a, b, c. G denotes the gateway model, P denotes process model and p denotes its instance with event set ϵ_p.

Sequence

The simplest construction represents the *sequence pattern*. Assume that there are two *activity models* A and B, gateway model G and process model $P = (N, E, type)$ where $A, B \in N_A$, $G \in N_G$ and $type(G) = sequence$ and process model arcs are defined as $E \supseteq \{(A, G), (G, B)\}$. This situation is depicted in Figure 5.3.

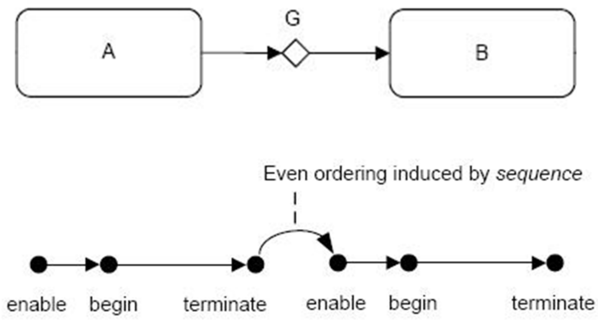

Figure 5.3: Sequence pattern [59]

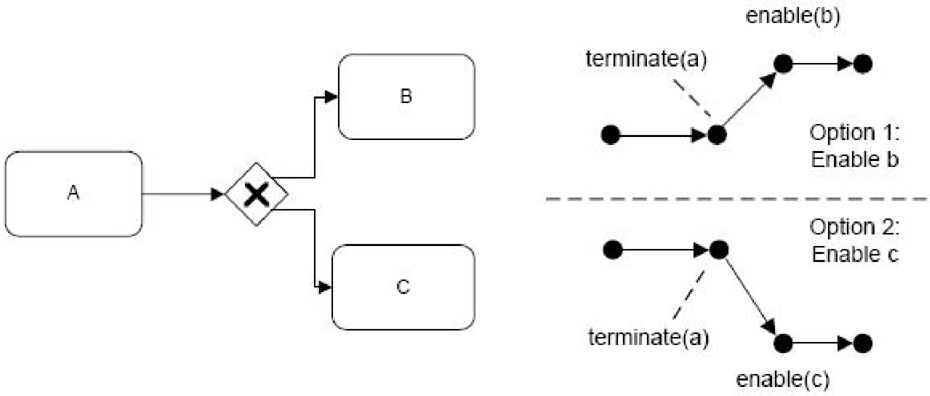

Figure 5.4: XOR Split pattern [59]

The gateway keeping the sequence patterns assures enabling the activity instance b after terminating the the activity instance a. The event ordering is depicted in the figure. For $\forall t_a \in \epsilon_p \exists e_b \in \epsilon_p : t_a < e_b$. The gateway keeping this sequence pattern is usually not explicitly mentioned. The activity order A and B remains the same, even after this simplification.

XOR split

The XOR split pattern represents the alternative between two or more branches of the process. Let us assume activity models A, B, C, gateway G. Model arcs of process P will be defined as $E \supseteq \{(A, G), (G, B), (G, C)\}$ and $type(G) = XORsplit$. The model with a related event diagram is depicted in Figure 5.4.

The gateway, which the control flow pattern is applied for, keeps the enabling activity instance b or enabling performing the instance activity c after terminating activity instance a. $\forall t_a \in \epsilon_p : e_b \in \epsilon_p \iff e_c \notin \epsilon_p$ such as $t_a < e_b$ or $t_a < e_c$.

XOR join

The pattern XOR join is a complement to XOR split pattern. It provides joining two or more branches of the process without synchronization. Let us assume activity model B, C, D, gateway G. Model arcs of process P will be defined as $E \supseteq \{(B, G), (C, G), (G, D)\}$ and $type(G) = XORjoin$. The pattern with the related event diagram is depicted in Figure 5.5. This patterns defines the behaviour of the gateway such as the enabling of activity instance d which is possible

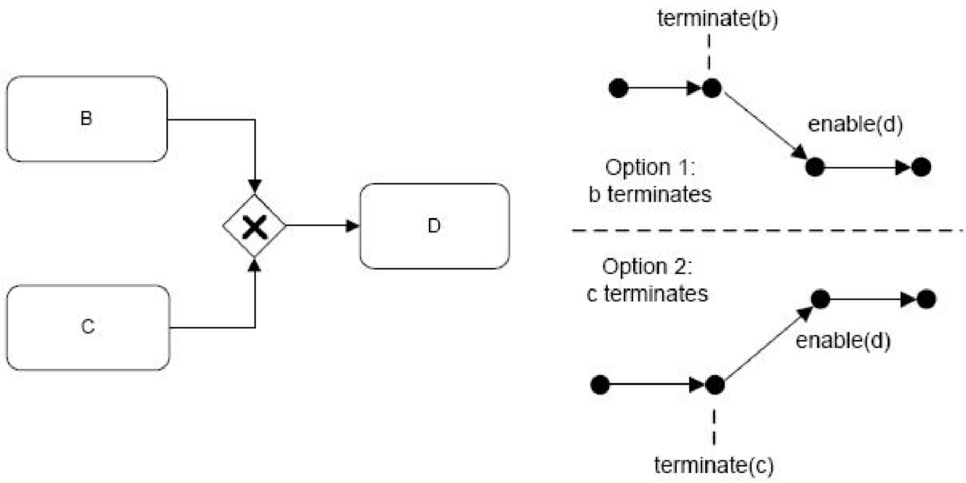

Figure 5.5: XOR join pattern [59]

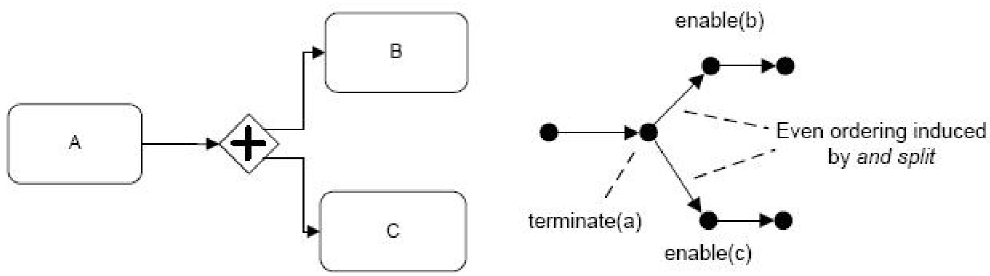

Figure 5.6: And Split pattern [59]

after terminating the instance activities b or c. It denotes $\forall t_i \in \epsilon_p$, so that $i \in \{b, c\}$ and there is a event $e_d \in \epsilon_p$ so that $t_i < e_d$.

AND split

The following pattern enables the modelling of parallel behaviour. The AND split pattern describes the place in the process, where one branch of the process is split into two or more parallel branches. It can be presented in activity models A, B, C, gateway G. The model arcs of process P will be defined as $E \supseteq \{(A, G), (G, B), (G, C), \}$ and $type(G) = ANDsplit$ (see figure Figure 5.6).

The gateway behaviour of AND split pattern ensures enabling the instance activity b and also at the same time activity c after finishing activity a. It denotes: $\forall t_a \in \epsilon_p \exists e_b, e_c \in \epsilon_p : t_a < e_b \wedge t_a < e_c$.

AND join

The AND join can be defined in a similar manner. The AND join keeps joining two or more parallel branches into just one branch. Let us assume the activity models B, C, D, gateway G. Model arcs of process P will be defined as follows: $E \supseteq \{(B, G), (C, G), (G, D)$ and $type(G) = ANDjoin$. The AND join model with a related diagram is depicted in Figure 5.7.)

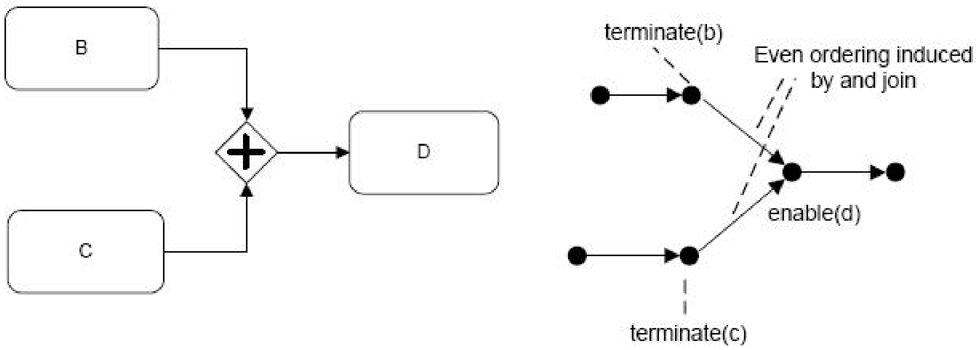

Figure 5.7: And join pattern [59]

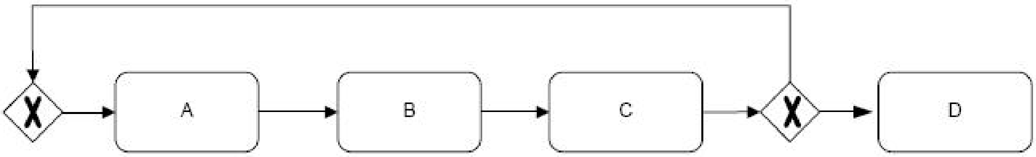

Figure 5.8: Iteration pattern [59]

The pattern applied to the gateway provides terminating instance activity b and c while enabling activity instance d, there $\forall e_d \in \epsilon_p \exists t_b, t_c \in \epsilon_p : t_b < e_d \wedge t_c < e_d$.

Iteration

The last basic pattern which will be presented is the iteration pattern. The iteration pattern assures that one or more activities can be performed repetitively. Figure 5.8 presents the iteration of the sequence of activities: A, B, C. The pattern uses the XOR for modelling the decision if the sequence will be repeated again or that instance activity d will be performed. In the case of repeated iteration, the new activity instance a based on model A is created. There are more patterns similar to OR (e.g. multiple merge, discriminator, N-out-of-M Join etc.). Their semantics can be defined in a similar manner as already seen on basic control patterns.

5.3 Modelling Workflow Using Petri Nets

Petri Nets formalism is widely used to model dynamic aspects in the system. Petri Nets consist of places, transitions and edges. Places represent conditions in the workflow. Transitions represent the performing task. Oriented edges connect places with transitions. It is possible to simulate a performing process under conditions by moving tokens from place to place within Petri Net [14].

We use common workflow patterns in Petri Nets to model basic business process modelling constructs such as: *sequence pattern*, *AND-split*, *AND-join*, *OR- split*, *OR-join*, *iterations* [57]. Other constructs defined using Petri Nets were described in [42]. According to [57], there are four types of routing: sequential, parallel, selective, iteration.

For sequential routing, it is necessary to perform one task after another, because there are dependencies between them. All tasks run concurrently in parallel routing and a construction consists of AND-split and AND-join. In selective routing, just one branch is performed and the construction uses OR-split and OR-join. In iteration routing, certain parts of the Petri Net are repeated until the appropriate condition is passed.

More information about modelling business processes can be found in [59] and [57].

There several types of Petri Nets, for example, *P/E Petri Nets* (support up to one token in places), *P/T Petri Nets* (support more than one tokens in one places), *Hi-level Petri Nets* [14]. *Hi-level Petri Nets* formalism also supports:

- colour,
- hierarchy,
- time (time stamps).

Colour extension enables token holding several attributes of certain type (modeling data attributes flowing in business process model). Hierarchy enables building process from sub processes. Time extension can be used for time related conditions, described in further detail in [57].

The Petri Nets formalism is not commonly used by users (e.g. *BPMN*, *EPC*, etc.). However, the processes are modelled in Petri Nets formalism because of a good support of various methods used for verification.

The workflow control patterns modelled in Petri Nets were used for comparing basic constructs of various languages [42]. The problem of migrating a newer version of the process definition while running the process instances is described in [11].

5.4 Changing Process Definition in Runtime

Updating a version of the process definition is not complicated as long as the *process instances* are not already running. However, it is also desired to deal with the situation of having old and new versions of the *process definition* running at the same time. The easiest way is just to wait until the running *process instance* will end according to the old version of the process definition and a new *process instance* starts according to the new version of a process definition. However, the process instance can run for a long period of time. Thus, there is need for changing an already running process instance to a newer version. There are several approaches on how to deal with that problem.

- It is possible to make a mapping of state related to an old version into a new version (seen in [11]). The solution is shown on Petri Nets, therefore it is based on the imperative approach.
- It is possible to make a simulation from the event log of the old process according to the rule of the new process as seen in [29]. The advantage of the solution is that it is more independent to a specific process approach. Therefore, it could be also used in something other than imperative approaches.
- It could also be possible to split one process definition into smaller parts that do not last long, when communicated via interface. The solution has the advantage in better control of changes. The disadvantage of the above approach is uncontrolled merging, however it can be solved by managing approvals.

5.5 Summary and Discussion

There was presented basic model structures which cover the mostly imperative way of modelling. However, the basic constructs are used widely in all modelling languages and it is possible to model the basic patterns even in declarative approach. The list of existing patterns is summarized in Workflow Controls Patterns [42] that also map every pattern to Petri Net.

However, there are more formalisms related to each approach. For example, S-BPM approach is based on Calculus of Communicating Systems (CCS Calculus) [35]. Declare is based on Linear Temporal Logic (LTL) [47], etc. The declarative approach and S-BPM are still not widely used in spite of the use cases where these more flexible approaches are much easier to use for modelling.

Chapter 6

Languages Used in Process Modelling

There are a lot of languages used in business processes modelling. In the sections below, some of them will be introduced. The particular language is a specific implementation of *business process modelling* approach. Thus, the brief introduction can give us an insight into *business process modelling* as well.

6.1 BPMN (Business Process Modelling Notation)

The BPMN is the most used business process notation for modelling business process models. The current standard BPMN 2.0 [39] defines modelling and notation for:

- process diagrams,
- collaboration diagrams,
- choreography diagrams.

We will focus on business process diagrams, therefore BPMN basic elements in process diagrams will be briefly introduced:

- Event – It contains start, intermediate, end event elements (e.g. message, timer, interruption etc.)
- Activity – It contains task, subprocess, activity elements.
- Gateway – It is used as the decision point (e.g. OR (Inclusive), XOR (Exclusive choose), AND (Parallel split) ,etc.).
- Flow – Sequence, message, association, data association flow connects Event, Activity and data elements.
- Data – It models data objects, data collection or data store.
- Artifact – It contains a group of text annotation symbols.
- Swimlane – Swimlanes are used for organizing and categorizing activities. Pools represent major participants. Lanes are used to organize and categorize information within a pool.

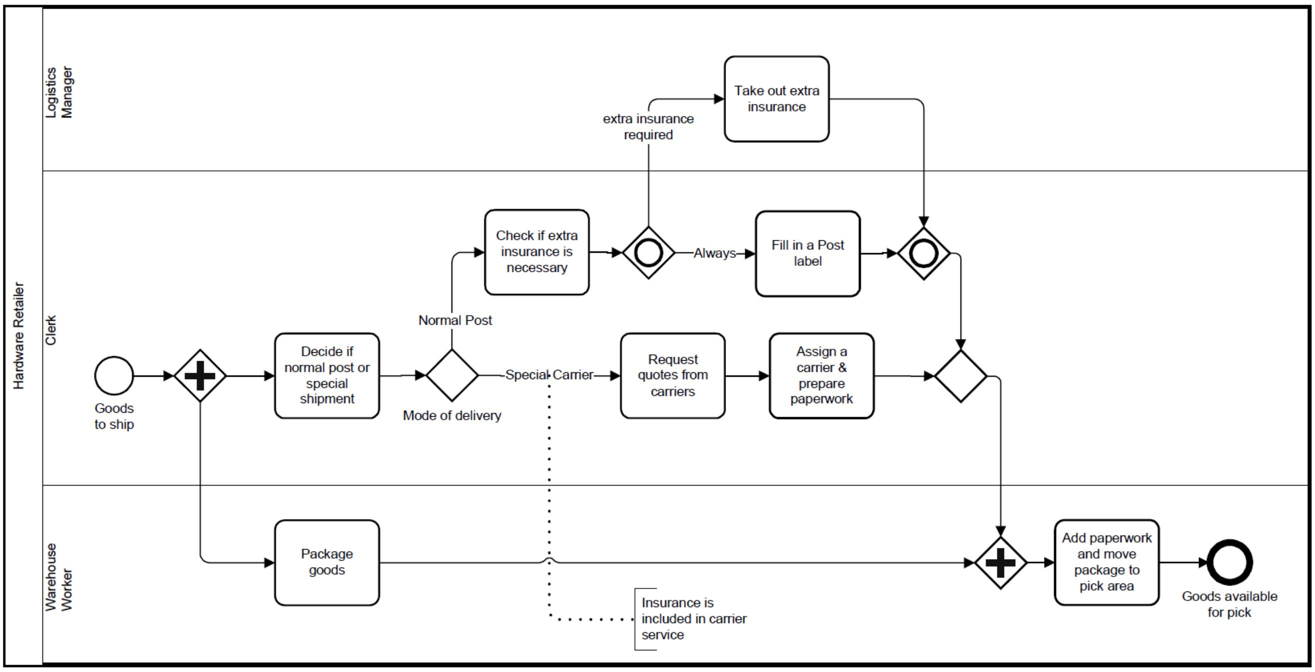

Figure 6.1: An example of BPMN diagram [38]

6.2 EPC (Event-driven Process Chain Notation)

EPC is a modelling language that can be used for description of business process . It contains basic elements [2]: *Activity*, *Event*, *Rules (gateways)* and *Flow*.

The following rules must be kept:

- An event must start and end the process.
- An event has to flow to a gateway or to an activity.
- Multiple functions can follow each event or multiple events can follow each function, but there must be rules in between. In EPC, such rules are called 'OR', 'AND', or 'XOR' and are represented as graphical connectors.

The Event-driven Process Chain notation is related especially to ARIS, SAP.

6.3 YAWL (Yet Another Workflow Language)

YAWL is the whole system based on the YAWL language. YAWL offers comprehensive support for the control flow patterns. It is a powerful process specification language for capturing control flow dependencies.

The data perspective in YAWL is captured through the use of XML Schema, XPath and XQuery.

YAWL offers comprehensive support for the resource patterns. It is a powerful process specification language for capturing resourcing requirements. YAWL also has a proper formal foundation. For its expressiveness, YAWL offers relatively few constructs compared to BPMN.

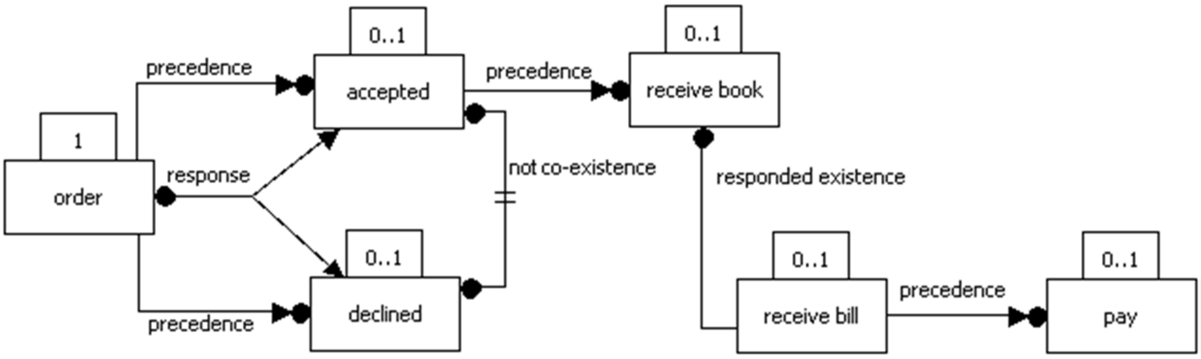

Figure 6.2: Example of process Purchasing a book in Declare notation [45]

YAWL offers support for exception handling, both for those that were and those that were not anticipated at design time. YAWL offers support for dynamic workflow through the worklets approach. Workflows can thus evolve over time to meet new and changing requirements. The idea's background and description can be found in [18] , the full specification can be found in the manual [13].

6.4 Declare

Declare is a prototype of a workflow management system. Declare is based on a declarative approach to business process modelling and execution. Unlike conventional systems, which use graph-like modelling languages (e.g. BPMN or Petri-nets), Declare uses logic to model and execute business processes, currently *Linear Temporal Logic* (LTL).

The connected graph contains several types of flow relation (e.g. *Precedence*, *Not co-existence*, etc.). The example and basic explanation to this approach can be found in [45], the example of the Declare notation is depicted in Figure 6.2. The table represents the relation between graphical notation and logic constructs as well as an example of the transformation of a model designed in declarative approach to the imperative modelling approach can be found [54].

The relationship between imperative and declarative approach is displayed in Figure 6.3. The white circle represents the state space of possible activities which would be normally defined in prespecified process, the grey object represents the state space of possible desired activities in the declarative approach, while the black circle represent the state space of the all activities, thus the orange area represents the forbidden state space of activities.

6.5 Proclets

The focus on traditional workflow management systems is on control flow within one process definition that describes how a single case (i.e., work-flow instance) is handled in isolation. For many applications this paradigm is not suitable.

Interaction between cases is as least as important. The Aalst [1] advocates the use of interacting proclets (i.e., light-weight workflow processes). By promoting interactions to first-class citizens, it is possible to model complex workflows in a more natural manner, with improved expressive power and flexibility.

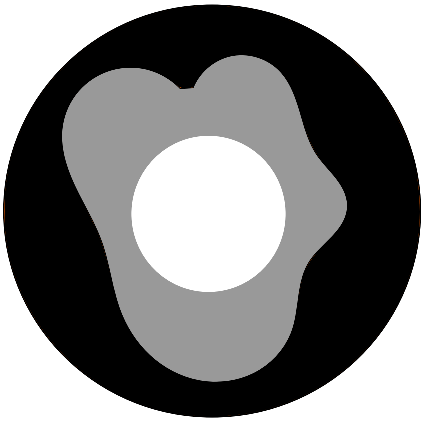

Figure 6.3: The relation of state space between the imperative and declarative approach

6.6 BPEL

BPEL (Business Process Execution Language) is an XML-based language and represents the block oriented language used for the business process definition. The language can describe the business process logic and specify the realization of a particular task using web services.

The model is composed of two basic parts [37]:

- *web services interface* – described by WSDL (Web Services Definition Language),
- *the process logic* – described in XML using basic modelling concepts (see bellow).

Interpreting the process description is provided by a web service orchestration. The process is executed when its execution is required by other related web services [37]. After this execution, the performance follows the defined activities while respecting the defined control flow. Performing of activities is realized by calling the particular web service.

The communication is provided by sending messages. The basic concepts of BPEL [37]:

- *Simple (Atomic) activities* covering the functionally of sending and receiving messages from web services (receive, invoke, reply), assigning a variable (assign), waiting until a time condition is met (wait) or process termination (terminate).
- *Structured activities* can be nested and can contain other nested or simple activities. By using this concept, it is possible to model the control flow.
- There are also *structured activities for sequential control flow* (*sequence*), *parallel execution of nested activities* (*flow*), *branching based on data-related conditions* (*if-else*), *receiving message* (*pick*) and *repeated performing of activities* (*while*, *forEach*).

Thus, BPEL implements the basic patterns.The definition of BPEL also contains variables that can store context during execution. The values of variables can be *control flow data* or data from web service messages. Variables have to be defined in the header of the definition of the BPEL process and the data type has to be selected from an attached scheme of web services interface (message types) from the XML schema.

Exceptions are handled by handlers. It is possible to model waiting for messages or a time event by using *event handlers*. There are also available *fault handlers* for handling the internal errors in process or it is possible to call *compensation handler* which can take back the effect of an action which causes the errors and return the process to the previous consistent state. Handling termination is provided by the*termination handler*.

Partners links represent the instrument for communication with web services used within the process. This graphical concept makes modelling easier.

The BPEL language is useful as a tool for the orchestration of web services. The messaging system supports especially well performing the parallel activities. I see the BPEL concept especially useful for software developers rather than business process engineers because of the difficulty of using the XML language in an imperative programming. I would say that difficulty of using BPEL can be comparable to a regular programming language while some other language constructs are missing. However, the usage as a connector for several distributed systems using web services for software developers is very useful.

6.7 Summary and Discussion

In this chapter, the most used language has been briefly presented. BPMN is the most used process modelling language standard for many users. However, I see it quite complicated because of many symbols; therefore, many vendors supports only part of the BPMN standard. For example, vendor Bizagi implements only part of the BPMN standard, although Bizagi contributes to the specification of the BPMN standard [5]. It is not easy to use BPMN for more complex processes. The pool mechanism is quite complicated to use for more structured processes. I would prefer the EPC style of modelling where a role is associated with the task. Most of the software used for modelling of business processes in the imperative way support at least import BPMN. XPDL is an xml-based format often used for storing the BPMN process diagram.

EPC is based on processing events. I would prefer the EPC way of modelling comparing to BPMN. EPC contains fewer symbols to remember and there are no pools.

The BPEL language is mostly used by software engineers rather than business users because the way of usage is not easy. However, it is widely used and there are also graphical designer tools enabling easier designing of the business process. However, it is still tightly connected with web services architecture which it was originally designed for.

The *Declare* as a concept seems to be very interesting; however, it is not supported so far by important vendors. However, the notation is not so intuitive as, for example, BPMN notation. However, adding constraints to the business process management notation can be more intuitive in capturing the process model based on the business process than enumerating each possible state of the business process.

Chapter 7

Analysis and Optimizing of Business Processes

There are several approaches related to business process optimizing which are usually combined together in order to increase overall performance and quality of the process outputs.

Optimizing is usually maintained by supporting processes. The supporting processes usually influence the context of *process instances*. The supporting process can cover the training of participants, investing in new equipment, etc. However, the changing properties of human or other resources can influence the process definition as well. The problem can be illustrated in the problem beginner vs. expert [32]. If both beginner and expert follow the same rules, it can lead to a decrease in overall performance.

The approaches mentioned below will be briefly presented. There are many perspectives and methods covered in the optimizing of the business process. A few of them will be briefly presented.

7.1 Key Performance Indicators

Setting key performance indicators (KPIs) is usually unique for a selected company, because it is connected with the overall business strategy. However, there are also general applicable economic KPIs.

- *Success rate* – This indicator is often used in the low level business process hierarchy and is used for indicating an area that can be searched for in the context of high probability of error.
- *ROI (Return on Investment)* – It represents the ratio of profit to investment, usually used at a higher level of abstraction.
- *EBIT (Earning Before Interest and Taxes)* – Profit that includes all expenses except interest and income tax expenses. It is the difference between operating revenues and operating expenses.

Key performance indicators are probably the most used systematic approach for the evaluation of business processes. The main disadvantage is difficulty in determining what is relevant and possible to measure at the moment. The KPIs are tightly connected to the overall strategy. However, it is possible to use the economic KPIs that have the disadvantage of reflecting the improvement very late, especially in the development phase when money is not coming in because product is not selling.

The necessity of using KPIs is the outlook over whole system performance. It is possible that only one part of the system which was improved could decrease the performance of the whole system.

7.2 Lean Six Sigma

A huge variety of mostly statistically based methods and tools are connected to *Lean Six Sigma*. These methods help to increase stability and predictability of the process (avoiding errors and other unpredictable issues) due to a process standardization. The method was originally designed primarily for production processes; however, it is also possible to use it partially in services [23].

I have focused on studying the impact of deviation in the performance of the process. We have tried to increase predictability of the process performance by taking context-depended variables into account which led to decreasing process deviation and increasing the predictability of behaviour [48] [49].

7.3 Value Stream Management

The method evaluates outputs of certain parts of the system with respect to value for a customer. There should be a correlation between the value of functionality for customer and effort (costs) spent on a part of the system. Thus, it is not supposed to invest much money to developing part of the systems which is not enough valuable for customers. The *value stream management* is popular especially in manufacturing enterprises; however, it can be applicable to the service area with some modifications. A dynamic approach to value stream mapping is presented in [3].

There is also the related term *customer process alignment* which is a method focusing on the transformation process according to customer needs.

The idea has a relation to the merging part of the systems according to inputs and outputs used in *context-aware approach* described later.

7.4 Bottleneck Analysis

It can be useful to predict a bottleneck using simulation which describes behaviour based on mathematical models. It is possible to model systems as a set of objects that produce or consume other objects with a certain probability of their occurrence. The analysis of general network flow was presented in [22].

In order to increase predictability in processes, it is helpful to have as small deviation as possible. Adding context-depended variables in order to automatically differentiate the specific behaviour is the way on how to improve predictability of the process. It helps to create a more complex model based on real-time data.

7.5 Business Process Reengineering

The method often relies on human creativity in finding ways on how to improve desired processes. The business process reengineering can be risky especially in the initial phases [4]. However, on average, *business process reengineering* positively affects the performance. The new versions of processes can be simulated and compared to the older versions of the process.

The query based approach mentioned in *context-aware approach*, described later in this thesis, is very helpful during the initial phase of business process reengineering because it helps to create all possible combinations of making the same output in different ways, or it can be used as an analytical tool. It is possible to find out the part of the process which is the most time and cost consuming and make a simple new version of process using existing subprocesses in order to achieve better performance.

I believe, it is very desirable to redesign a process in such a way that it tends to be context dependent, because for each context variable configuration can be a different optimal process configuration.

7.6 Process Mining

Process mining is one of the important issues related to the business process area. Process mining is the methodology of extracting information from an event log (for example audit data created by business process management system). Data are typically captured by the running of the system. The basic approach is focused on analysing relations between specific events [56].

Due to process mining, these kinds of problems can be solved:

- discovering the process model (typically from event logs),
- conformance checking (comparing business process models with each other),
- performance analysis (extracted statistics from event logs mapped to the process model),
- prediction of process evolution.

The aim of process mining is to make connections between events and model. The input is an event log. It can be extracted information such as:

- process model,
- model of organization structure,
- social network,
- sequence diagrams,
- bottleneck, etc.

Process discovery is able to analyse the process event log. An event log contains events that were executed during a run of the process. Every event corresponds to a task and a case. Events also have a start and end execution time (they must be ordered at least). Using this information, process discovery is able to find the process model of the task sequence.

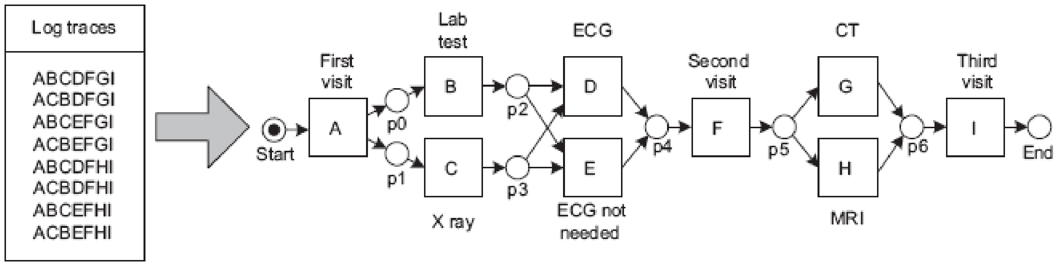

Figure 7.1: Process discovery [52]. It is possible to discover a process model from logs. The discovered process model must be able to replay most of the log traces

Process mining uses only the event log as the input. However, it can be necessary to do more complex analysis, for example, analysis taking into account properties of business process management system users participants. But it is not possible to provide this kind of analysis because such information is not typically stored. Thus, it is necessary to add information about the current configuration of business process management system users to an event log. An example of capturing context was presented in [31].

Another work in business process prediction was from Grigori [15][16]. She used classification over all case attributes for business process prediction. However, her work did not focus much on resource. Wetzstein [60] did related work where bottlenecks were identified using similar methods (classification). This work also covered resources (resources could be bottlenecks too).

7.7 Summary and Discussion

It this chapter, there were briefly presented the terms related to optimizing and analysis of business processes. In reality, a combination of the approaches is used. I believe that the evaluation of the process performance should also be connected with strategical level (goals), which is represented very often by key performance indicators.

The evaluation of historical data and finding deviations in performance is also a very important step in order to keep all processes standardized. Value stream mapping is one of the methods which is used, especially for lean business process development, because it is evaluating each process of value for the customer which is a kind of strategic metric.

Bottleneck analysis is very useful if the data from a historical performance is available to be used. If process orchestration is changed, it is necessary to evaluate it if there is no significant bottleneck.

Process mining is especially very useful in cases where there is already running some kind of information system which logs events. We can create process maps based on logs if we can assign all events to particular instances of the same process type. However, this cannot be easy applicable in many cases. Thus, a possible way on how to decode the process map is by following the data flow in order to find out the log instance and decision patterns.

I see the problem, especially in gathering data for analysis and most of the analysis needs a lot of effort which means they cannot be done frequently. This implies the late detection of the problem in process performance. I believe the core of the problem lies in, that data for analysis is not collected all the time and it is not sortable according to the context which is, in my opinion, very important. If the context of data is unknown, the data can be transfered to useful information with a lot of difficulties.

In the case study related to a manufacture company, we had a lot of problem with the quality and description of the data. We had to know the context of the environment when the data was collected, otherwise the results would be misleading. The data about part of the context was stored only in the memories of people (e.g. changing machine).

The datamining methods are very useful for analysis of context influence. In conclusion, I would say most problem of analysis are related to the unknown context of existing data. The business process reengineering is connected with a changing strategy, which means changing the goals in a very short time.

Chapter 8

Summary of State of the Art and Setting Goals of the Thesis

As already been presented, there are several modelling approaches for modelling business process that are based on different ideas, especially the declarative compared to the imperative approach. Each approach is more suitable for usage in different use cases, for example, imperative way of modelling is a better option in case it is desired to have precise control over all processes and is possible to prespecify the process (e.g. some production processes).

On the other hand, the imperative way of modelling is not suitable for the process where there is a huge number of possibilities of performance and it is not possible to have full specification of the process (e.g. processes related to medical treatment). The declarative way of modelling is better option in the case of loosely specified processes. As we can see, there is no better option for all cases. Thus, it is helpful to have architecture which supports multiple modelling options for modelling processes and subprocesses.

One of the goals of the thesis is to find a way on how to make orchestration of different modelling approaches. Another problem is flexibility in runtime, currently solved by manual intervention in the process instance (cased-based system) which is quite time consuming. Especially in cases when there is a significant set of situations where the handling cases are very similar at a higher level of abstraction (e.g. typical question which part of process should be outsourced).

There are often several ways of reaching this goal, for example, in many cases it is possible to make a decision whether to buy or make a product, however, the decision is often based on our previous experience and current context of the company.

I believe that the ideal business process engine architecture should meet these requirements:

- It should be able to support several modelling approaches, at least the imperative and declarative approach.
- It should support all levels of abstraction of modelling (strategical, tactical, operational).
- It should be context-aware.
- It should be primarily driven by a goal (tactical level) in order to keep the right process alignment, while selecting the best possible option on how to reach the goal considering current context.

- The system should be able to learn from historical interaction with human and system.
- The system should have a high-level of flexibility in order to quickly adapt to context changes and reconfiguration of tactical level should immediately impact the operational level.
- Everything should be able to cooperate as much automatically as possible.

The mentioned requirements are not present in most currently used business process management systems. I believe this meeting that requirement is important for a creating context sensitive business process management style with a systematic approach of dealing with the uniqueness of resources.

The most currently used systems do not support a process alignment according to the tactical level well. The alignment is often done manually by process engineers and this takes a long period of time. In order to utilize the resources at the best possible level, it is desirable to handle them in a systematic but individual manner and adapt the managing resources to the strategic objectives as fast as possible.

8.1 Description of Suggested Architecture

In this thesis, I propose the new architecture design which is based on WfMC reference model (see chapter 3), however I add several components which improve the system in context-aware adaptation point of view. The components placed into WfMC reference model is depicted in Figure 8.1.

I add these components:

- *Database of Processes*,
- *Process Orchestrator*,
- *Dynamic User Profiles*,
- *Dynamic Process Profiles*.

Database of Processes enriches the original process definitions with process interface enabling building the *Product Dependency Tree* (see section 12.8) and is described in more detail in section 12.6.1. The *Database of Processes* is the collection of *Process Definitions* with merging interfaces.

The *Process Orchestrator* uses *Dynamic Process Profiles* for heuristic analysis for a selection of optimal orchestrations based on desired products and constraints (see section 12.6) and also uses processes from *Database of Processes* as the building blocks for the process orchestration, more details related to the query-drive context aware process orchestration is described in chapter 12.The *Dynamic Process Profiles* can derived from Workflow Control Data, or can be created directly by *WFM Engine*.

The process orchestration in this thesis is also applicable to service-oriented architecture with small customization. Thus, it shows ability of a usage of the Process Orchestrator in distributed environment, for more details see chapter 13.

The component *Dynamic User Profiles* is used for providing ability of context-driven management based on user properties, see chapter 11. The Dynamic User Profiles uses *Organization/Role model* and *WFM Engine* for collecting data about user behaviour within WfMS.

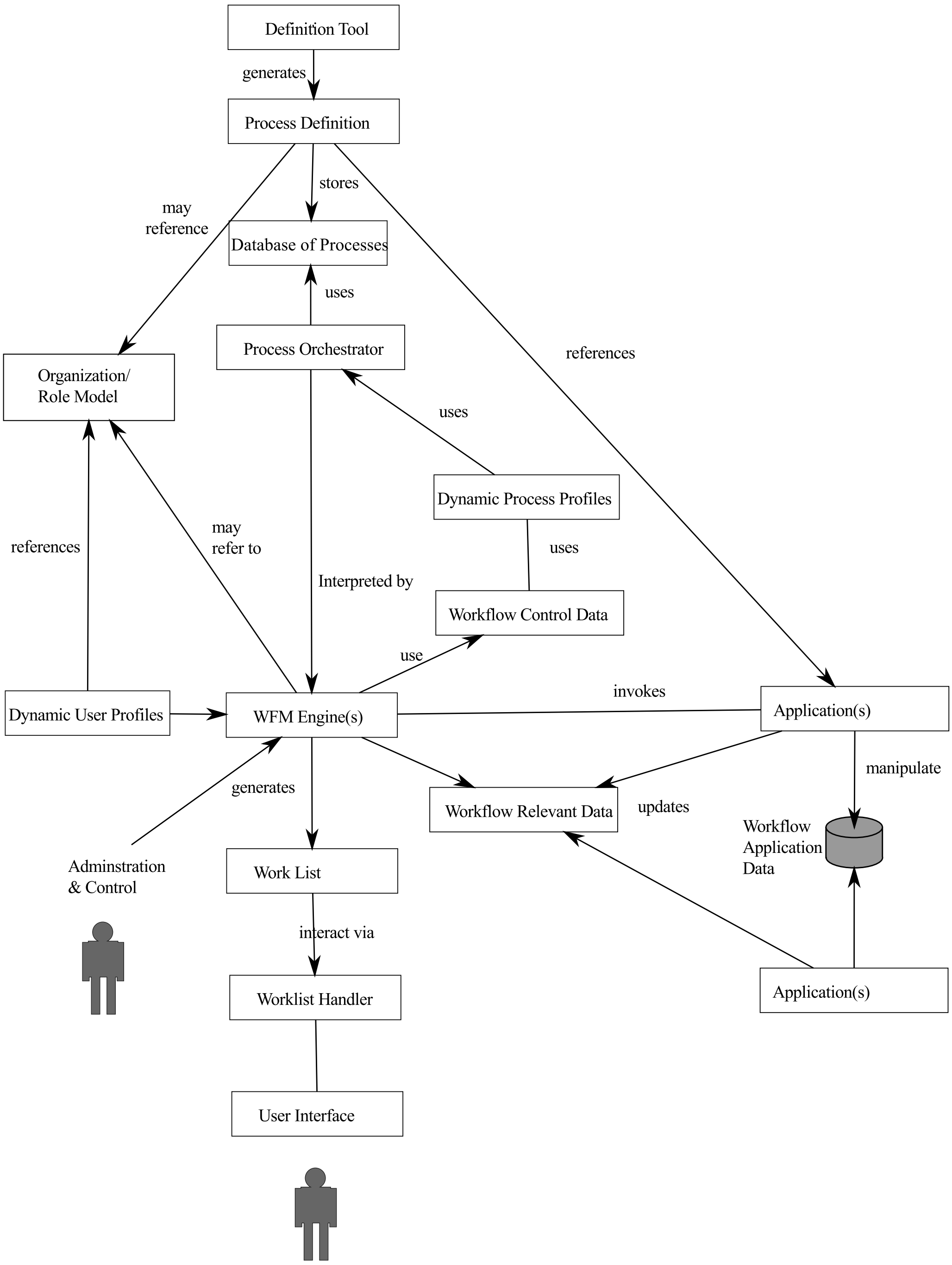

Figure 8.1: Suggested architecture of the workflow management system based on WfMC reference model

The components *Dynamic User Profiles* and *Dynamic Process Profiles* are specific usage of concept labeled as *Dynamic Objects* which is presented in chapter 10. The idea behind the *Dynamic Objects* was influenced by my research in manufacture. The importance of context management in process has been shown there, see chapter 9 for more details.

The suggested architecture supports dynamic process orchestration of distributed subprocesses. Each subprocess can run on different *WFM Engine* and it can be described in different business process modelling approach because of unified interfaces added to subprocess. It uses data captured from performing of the processes by WfMS labeled as the *Dynamic Objects* for providing information about context to the overall system.

The dynamic dynamic orchestration is driven by setting objectives (desired products) and constraints. Thus, if we add Dynamic Objects, we get the Context-Aware Workflow Management System as the final result. The dynamic process orchestration is driven by query that declares information about desired products and related constraints.

Chapter 9

Impact of Context to Performance

In this chapter, we will present the way how real performance is influenced by the context variables. The case study was done in a manufacturing environment where a high deviation of the process was not expected. Many current *business process management systems* store a lot of information about processes, data flow, resources and execution time.

If data about the performance of every activity is combined with data describing context, then interesting results will be revealed. In the case study, the hypothesis against several millions of records of historical data describing a process of production were tested.

9.1 Outline of the Manufacturing Company

The manufacturing company has a network of machines and workplaces where workers make operations manually. The product is routed between machines by humans according to requests for the production operation.

The production process is built according to the parameters that describe the final products. Almost each product is unique, because the company is able to make several million variations of its product by choosing a possible variant of production operation.

There is standardized performance time for each operation, however, standardized performance time does not distinguish between variances of the products, thus the estimated standardized performance time does not correspond to the real time needed for performance of the operation. There is a significant deviation comparing standardized performance time to real performance time.

9.2 Research Results

According to the data analysis related to manufacturing, there is a significant variance of performance time in each operation even if they should fit to the standardized performance time. Thus, one of the goals was to find a dependency between specific attributes in products and performance time of the operations.

We used mainly a datamining approach to analyse the problem. We have found significant dependencies. In papers [49] [48], we are describe some of the results.

The main motivation for an analysis of context dependency in a process was the optimizing of the process. The optimizing context variables of the process is one of the most cost effective ways of business process optimizing. A change in the production process would be very expensive in the case study.

Base of the analysis, it is possible to find hidden dependences related to the context of the production process. If we add context to the process model, we gain a more precise model for prediction which can be used in logistics. We have found the deviation of performance time while using context extension can be decreased approximately by up to 50% of the performance time which is only computed as average performance time even in a standardized environment. We also have tested several approaches from a performance and accuracy point of view.

The decreasing expected deviation can be used (e.g. in the prediction model) for a more precise *bottleneck analysis*. The *bottleneck analysis* is often used for designing better logistics (e.g. designing of temporal buffers). Better logistics is often optimized for *Just in time* approach – it is optimal to have material available exactly at the time when it is needed. The better short-time context based prediction has a direct influence on optimized planning, because lower deviation from the plan supports better performance of the process instance.

There was also an analysis related to manual work. The goal was to find out if someone is more productive in a specific task. The other goal was a better prediction of performance time based on a combination of a set of specific tasks and particular worker. This information can be used for better planning of shifts. The problem of planning while not using a performance profile of workers is that the deviation of performance time of different workers with a different task is so significant that it is very difficult to plan such an amount of work to be done in order to perfectly fit to a planned shift. If the work is not done during the shift, the work is often done till the next day which causes an unplanned delay. In the case of underestimating the amount of work, there is unused capacity of workers. The job for workers has be prepared in the previous steps of the production process, thus the amount of work has to be planned.

We have also tried the analysis of the general context. We tried to find out the dependency of error and keeping a deadline of suppliers on a specific week day. We have found that there is a significant dependency. The highest error rate was on Monday.

Based on the results of a case study which took several years, we have supported the hypothesis about relevance to the context. However, most of the current system is still not context-aware which can cause inefficiency of management, because this is an easier way to make a better management plan when we are able to make better predictions of the future.

Chapter 10

Dynamic Objects

One of the largest disadvantages while using the classic business process management system is the lack of flexibility of process management. However, the classic *imperative approach* is widely used so far in many business process management systems and the main benefit of the *imperative approach* is the full control over the process execution.

Therefore, it is important to enable flexibility in a *process-aware information system*. The performance of the process is influenced by the logic structure (defined rules in a process), resources, human and many other aspects that can be specific for each *process instance*. The logic structure in the process does not change as often as other context-depended variables do. It has significant influence in a process process, especially if there is a high variability of time in the process executions.

Adding context to process management can significantly improve prediction in *process instance* performance. There is also a possibility to add new context-dependent rules to process definition. It enables a more precise managing of *process instances* and it also provides functionality to differentiate behaviour according to a specific product or assigned recourses.

There are many events that can significantly influence the result of process performance. Unexpected occurrences are mostly caused by changes in the environment of the process. It can be represented as the context state of time in the process performance. A set of business rules is only a matter of the business process management.

Nevertheless, it is also necessary to deal with the context of process performance. Since the environment of a process is variable, it is usually necessary to adapt the logic of the process. This is usually done by replacing one part of the process by another one that produces adequate results. For example, if the input resources of a subprocess are temporarily unavailable, a need may arise to change the logic structure (e.g. by outsourcing the particular part of the process).

The process execution can be limited by certain constraints such as:

- *costs* – The amount of costs can influence the quality and speed;
- *time* – The abnormal deviation can cause the synchronization problems;
- *success rate* – The sufficient success rate influence cost and time dimensions;
- *quality* – Assuring the quality that has to be contained in the process;
- *added business value* – Added business value has to correlate with costs, etc.

Thus, the logic of the process should also be adapted for possible changes in context attributes that have an influence on the choice of variants of a particular process.

Resource management is another important aspect of process performance. The human factor has a large impact on overall results of processes. Every business process management system user is different, thus this is the reason why the behaviour of the workflow management system should adapt to requirements related to each individual person. The adaptation of the workflow management system suited for individual users can positively influence the performance of the workflow management system user. A more detailed discussion related to the problem will be described in next chapter.

The presented solution uses the workflow management system as a collector of data from the previous process performances. At the same time, *dynamic profiles of the objects* in the *workflow management system* are created from the collected data. Thus, the *workflow management system* is used as a measurement tool.

The context of the process can be described by a set of variables related to several *dynamic objects*:

- People acting in *process instance*,
- resources used in *process instance*,
- products flowing within *process instance*,
- process components,
- tasks in *process instance*,
- objects representing the relationship between objects mentioned above (e.g. human relations and tasks).

These kinds of objects can be influenced by the management of the company. Therefore, understanding the influence of certain attributes can help to improve overall performance.

There are also context variables that are not directly related to the *process instance* and have an influence in the context, for example, performance during specific weekday. However, it is also possible to analyse them. Understanding their influence can be used for improving the managing process structure and manageable context attributes mentioned above.

The importance of context related to user was presented in the paper [32]. The importance of analysing product attributes as the set of context variables was shown in the papers [48][49]. Based on the analysis described in the papers, there the overall predictability of the system can be improved.

It is possible to manage the dynamic profile of each object with minimum effort due to the workflow management system. A dynamic profile is a set of properties which has a dynamic part which is continuously changing.

The dynamic profiles of resources are especially used for better managing of resources within the workflow management system. It is also very useful in an assignment procedure.

The presented algorithm in this thesis is based on the dynamic process redesigning which uses the *dynamic profiles of processes* for selecting the optimal way on how to reach the desired goals of a process. The next important step is to build a *product dependency tree*. Its purpose is to map all possible ways of creating the desired products. A route in the *product dependency tree* is used to build the final structure of a modified process. The *product dependency tree* represents a product hierarchy. In other words, it reflects what income products are needed in the production of desired products.

The presented approach in this thesis has several advantages. It allows us to switch over to another way of producing the product according to changing conditions. This can be done due to the declarative approach. It is necessary to define the final products and a set of conditions that are related to the context of the process.

By using database of processes which includes process definitions and dynamic characteristics, it is possible to build new processes according to specific conditions. Thus, the process can be modified according to the current context (e.g. costs, time, success rate, resource problems, etc.). This means a great advantage because information about the context of a process is often not available before its runtime.

The managing approach is also suitable for use in the *ad hoc workflow management system* where it adds some level of managing and programming rules, even to the process structure which is being built in runtime.

Product substitutability makes the presented solution flexible and feasible at the same time. The substitutability means that within a group of products needed for creating the final products, one product can replace another product so that the target product characteristics are not changed.

Chapter 11

Context Based on User Experience

This chapter discusses the problem of managing resources by utilizing the *workflow management system architecture* taking into account the user properties of *business process management suite*.

A problem will be presented here using architecture according to *Workflow Management Coalition* – no distinction is made between participants of *workflow management system* with the same role. A possible solution lies in the modification of the *process definition*. The chapter shows how to use the *workflow management system* for setting up *dynamic profiles* of *WfMS participants* which significantly supports the solution for a more precise managing of resources in the *workflow management system*. This modification of the *workflow management system* also enables other ways for an analysis of the business process. The solution can also improve assigning users of *workflow management system* to the task [32].

11.1 Basic Attitudes to Assignment User to the Task

There are two basic attitudes to assigning the user of *business process management system* to the task:

- *manual assignment* (assigning user to the task is done manually denoted person),
- *automatic assignment* based on role mentioned in the *process definition*.

There is a person who is responsible for the assignment of the item from the worklist to the user of the business process management system in a manual assignment. The manual assignment has the main advantage of taking into account *WfMS participant* properties.

The person is able to distinguish between particular persons, but there are also some disadvantages:

- The person who makes an assignment can be overloaded (queues).
- The person has to know properties of all *WfMS participants* very well in order to utilize the main advantage.

The manual assignment usage is more expensive and slower than an automatic assignment. Another way is to automate the assignment items from the worklist to *WfMS participants*.

There is information about which group of the *WfMS participants* can manage the task within the workflow (role-based approach) in a *process definition*.

The *WfMS participant* is chosen from a group which is capable of performing a task just before performing the work item.

The automatic assignment has the advantage of cost savings. Performance of the system is also higher. On the other hand, the system loses the ability to choose resources more individually. An improvement of the automatic assignment is a decision based on more suitable parameters of user of the *business process management system*.

11.2 A Problem with Handling Users in Business Process Management Suite

There are usually several ways how to perform certain work, but *workflow management system* cannot distinguish between BPMS users from the same group and thus it only deals with them according to participation in certain groups. Therefore, the system forces an interaction with a beginner in the same way as with an expert. However, it could cause some problems. Let us try to answer a very simple question: "How many checkpoints are needed for making a subprocess perfectly suitable for a beginner and an expert in a particular task at the same time?"

It is necessary to keep some level of quality of the process in order to check if everything is going well in certain states of performing the task in order to reveal mistakes as soon as possible (at the worst case at the end of subprocess).

Let us suppose the *process definition* perfectly fits the beginner. Now the *process definition* is perfectly set up for the beginner, but the *process definition* has just been made less suitable for an expert at the same time. The expert will not be comfortable with the system and his productivity can even decrease as a result of it. The reason is too much interactivity with the system or losing freedom in performing a subprocess. The expert naturally does not need such care of *business process management system* as the beginner. The expert can keep to a low level of making mistakes while performing the task.

Let us try setting up the process according to the expert (the process will probably look much simpler, for example, less checkpoints), but we have just made another and maybe a worse problem by decreasing checkpoints. The beginner now has too much freedom, thus mistakes made by him will be revealed much later. It can significantly decrease the performance of the beginner, because he will find the mistake at the end of his job at the worst case, which means that this part of the process has to be repeated. It also implies another allocation of resources during the performing part of the process, because mistakes revealed later mean larger costs.

Another problem can be, for example, related to different nationalities. Every nationality has a different cultural background; therefore *business process management suite* should be able to deal with these characteristics as well. The age and education of the human participant can also be important in certain cases, etc.

11.3 Modification of the Business Process Management Suite Behaviour

In order to solve the preceding problem, we need to change some parts of the *business process management system*. At first, it is necessary to change the *process definition* in order to support the distinction between users of *business process management systems*, even if they are placed in the same group. Placing users of *business process management systems* to a group represents that a user of *business process management system* has a certain qualification for performing certain tasks.

Let us suppose an assignment of the participant to the subprocess has just been done, all his properties can be read and the logic structure of process can be set according to its current set of properties (similar to a *manual assignment*).

It is possible to enrich the *process definition* with the rules using certain properties of the user of *business process management system*. Thus, it is possible to choose the right route in the subprocess according to their particular properties of the *WfMS participant*.

11.4 Managing Profile of Business Process Management System User

In order to support the *process definition* with an extension containing rules based on properties of the *WfMS participant*, an up-to-date profile (the set of properties typical for the particular user of *business process management system*) has to be managed. Let us separate groups of properties:

- *static properties*,
- *dynamic properties*.

In the case of *static properties*, it is possible to manage them manually (editing records manually); these properties do not have the specific relation to an existing activity of the company, for example, age, sex, education, nationality, etc.

In the case of *dynamic properties*, it is very difficult to keep the proper values up-to-date.

Examples of *dynamic properties* are:

- experience with a particular task,
- error rate,
- experience with a group of tasks, etc.

Therefore, *dynamic properties* describe the behaviour of the user of *business process management system* in an organization. It would be very helpful if it was possible to distinguish between a beginner and a more experienced user, thus *dynamic properties* are needed. But updating *dynamic properties* manually would be almost impossible without automatic support.

11.5 How to Get Dynamic Properties of User

The problem with updating of *dynamic profiles* can be solved by using the *business process management system*. While running the *business process management system*, the data about behaviour of the *business process management system* user can be stored.

It is often needed for the review of history related to a particular process. Thus, information can be extracted about:

- the error rate,
- the experience with a particular task,
- the experience in his profession, etc.

The *error rate* can be found out by counting positive and negative results of the confirmation test at the checkpoint. In the case of positive or negative results, the particular counter is incremented. Of course, the result closer to present time has a higher weight than a result which was done more in the past.

The *experience with the particular task* can increment the specific counter for the participant and the task. It should also be taken into account on how far data were created in the past. Data created in the present are probably closer to reality. Experience with a certain group of tasks is the example of the monitored indicators of overall quality of the *business process management system* user.

In order to manage *dynamic properties*, updating specific values is triggered by events which are created by *business process management systems*. Events are happening during the performance of a *instance* of the particular process.

11.6 Profile of Business Process Management System User

The *static profile* of *business process management system user* is a set of properties with declaration of *static properties* which are valuable for the company. The information can be updated by filling-in some forms by users of *business process management system* mostly at the beginning of user participation in the company and later manageable by office administrators or by themselves.

The *dynamic profile* is created by interactions between the *business process management system* and its users. Events created by performing the *process instance* trigger the updating of *dynamic profiles*. The structure holds a collection of items related to particular tasks. Every item holds the set of monitored indicators (error rate, level of experience, etc.) for each task. It also contains a collection of items containing data on a group of tasks (error rate, experience) which represents overall experience, see [31].

11.7 Modification of the Process Definition

The motivation for the modification of the *process definition* consists in better managing users within *business process management system*.

The *process definition* contains in general:

- *information* about the task in the process (task properties, role of resource),
- and *routing* (rules used to specify an order of the performing task sequential, parallel, selective, iterative).

In order to solve the above mentioned problem, an extension of routing information is needed (especially *guard functions* and *arc expression functions* to include variables based on properties of resource). If all important user properties (e.g. error rate, experience with certain task) are known due to managing of *dynamic profiles* by *business process management system* then a change of the definition of the business process will be possible as well.

It is possible to model such an extension in *Hi-level Petri Nets* (commonly used as process modeling formalism):

- I use a current model of *process definition* built from patterns (see [42]).
- A token has to carry data necessary for performing a task and routing information. It is necessary to include information containing a set of properties of *business process management system user* assigned to the task in a token in order to use it for the routing.
- It is also necessary to add *guard functions*, *arc expression functions* which will use *business process management system* properties as the input.
- New subnets of Petri Net can be built according to the preceding rules.

As it has been shown, I still use *Hi-level Petri Nets* even if the extension enabling interaction with the *business process management system user* based on his personal properties was added. It implies that the same methods of analysis can be used as it was possible before the extension, but I gain the *process definition* which can describe different behaviour depending on properties of the *business process management system user*. Thus, new rules which can work with properties of the *business process management system* user can be added.

11.8 Example of Modified Process Definition

In Figure 11.1, the process of creating a report is presented. The expert is able to make a report as one single step. He can combine gathering information and writing. He can recognize correct data. But the less experienced user will be checked at the first checkpoint to see if he has correct and relevant data before he will continue writing the report. Because the user has less experience the data has to be checked before he starts with writing. In the case of failure in the first step, performing the second one is not necessary and the subprocess can be immediately restarted.

By repeating the creating of the report, the user gains knowledge and constantly improves his performance. Getting experience simulates updating his profile after every step he makes. There is a test at the end of each step. In the case of a success or a failure, a particular counter is increased. In the case of a successful performance, a particular counter is increased.

In cases where a certain condition is met, he becomes a more experienced user and he can skip one checkpoint. In our case, the goal is to get the error rate in gathering information under 10.

In the case of performing *process instance*, the engine of *business process management system* checks if the condition based on the profile of the assigned business process management user is met in order to choose the right route.

Due to the extension *business process system management system engine* can interact with an expert in a different way than with a beginner, even if they are in the same group that qualifies them for the performing certain tasks. For example, the more experienced user can have a different route with a different number of checkpoints.

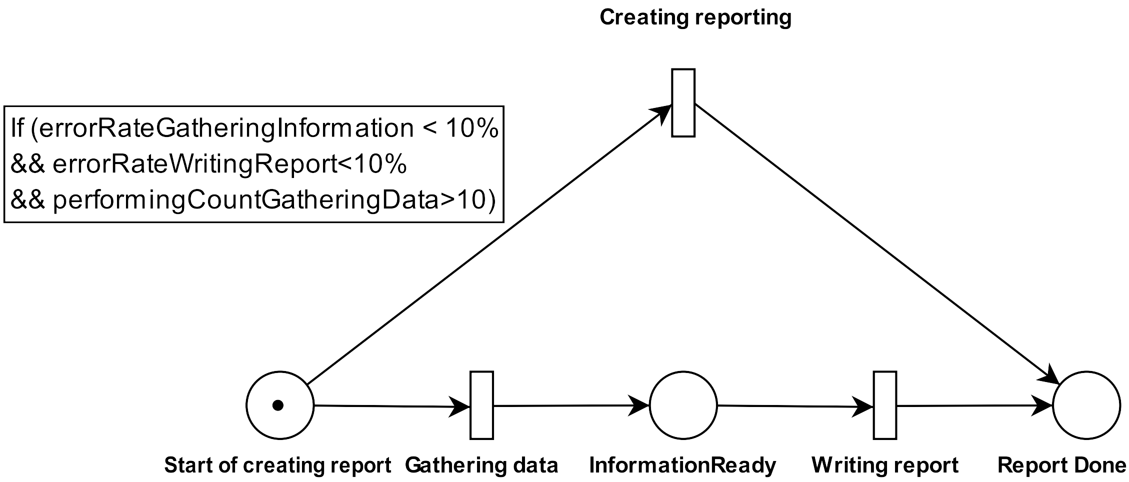

Figure 11.1: Example of changed process definition [32]).

Thus, *business process management system* changes its interaction according to each particular user dynamically according to his *dynamic profiles*. For example, if the user managed to reach some number of performances of the subprocess and has an acceptable error rate at the same time, then the *business process management system* would deal with him in a different way than before.

11.9 Example of Dynamic Profile of User

The dynamic profile of user represents his current state by configuration described as a nested structure. Thus, the the most recent state of the particular user can be described by static and dynamic properties. The configuration is constantly update by interaction user within workflow management system.

The static properties of user profile can contain:

- unique identification,
- age,
- nationality,
- sex,
- education, described for example by:
  - university degree
  - certificates

However, dynamic properties can be described by:

- collection of tasks, each task can hold:
  - error rate,
  - average time for performing task,
  - minimum time for performing task,

    - maximum time for performing task,
    - experience with task, represented by number of performing task;

- collection of task related to specific group (based often on roles of WfMS participant) , e.g. network administration, each record can also hold:

    - error rate,
    - average time for performing task,
    - minimum time for performing task,
    - maximum time for performing task,
    - experience with task, represented by number of performing task.

11.10 Adaptation of Business Process Management Engine

From the implementation point of view, meta-data of the *process definition* has to be changed and adaptation of the *business process management system engine* for performing the modified process definition is necessary as well. If the *business process management system engine* is supporting *Hi-level Petri Nets*, no significant changes will be necessary, because the *business process management system engine* will just work with more parameters and with larger *Petri Nets*.

Even if the *Hi-level Petri Nets* were changed in order to support an extension enabling user context-aware management, it would be described by Hi-level Petri Nets which could be interpreted in the same way as it was before.

11.11 Adaptation of Worklist Handler

By creating *dynamic profiles* of *business process management system user*, I had a benefit because I gained an option to distinguish between system users from the same group (for example, an expert vs. a beginner). I can also improve the assignment algorithm using the extended model. According to the priority setting of the *business process* (e.g. cost, time, quality), it is possible to make an automatic assignment based on properties of the user of the *business process management system*.

I can set up our priorities in the automatic assignment. It is desirable to prefer an expert in cases when it is necessary to decrease the error rate. It is desirable to prefer a cheaper resource if we are not in hurry, thus we can save on costs. It can be a good strategy to assign the beginner to the task in the case it is affordable it, because we also need to make regular updating skills of the *business process management system user* and save experts for more difficult situations.

Business process management system users have an impact on the performance of the process and process impacts the *business process management system users* as the feedback of their activity (performing the process also updates skills). For example, if we are expected to wait at the synchronization point, we will be able to use a less experienced user with a smaller impact on the overall performance of the process.

Parameters of the process can be static or dynamic. Static parameters can be set up at the beginning as the input to the assignment algorithm. Let us suppose these basic parameters:

- cost,

- speed,
- quality.

It is possible to detect divergence by currently running a *process instance* from casual performing of the process, because the history of the preceding performance of the particular process is known. Thus, it is possible to reconfigure parameters of the *process instance* which is dynamically based on the current performing of the *process instance*. Changing parameters impacts the assigning resources to the tasks.

11.12 Modification of Audit Data

The usage of an extended model based on properties of the *business process management system user* is very useful for *business process analysis* as well, because it is possible to record the current state of *business process management system user* during the running of a *process instance*. The *dynamic profile* at the moment of the assignment represents the current state of the *business process management system user* (set of properties).

Due to the triggering mechanism of *business process management system*, the profile holds up-to-date information about the current state of the user. Because *business process management system* knows the up-to-date profile of all its users, it can also record the exact state of its users due to auditing data at the moment of recording event. Thus, I gain the option to make an analysis which works with properties of the *business process management system users*.

For example, by changing a structure of the audit data, it is possible to answer the following questions:

- "How many times has the user had to re-do a particular task in order to get the error rate less than five percent?"
- "Which properties of the user have the best results in particular task?"
- "Which of the set of properties has the largest impact on the best performance?"

There are several reasons why we need to analyse properties related to particular tasks:

- It is possible to manage changing properties of *business process management system users* in order to adapt to changing conditions, because we can educate them, manage participation in certain kinds of processes, etc. Thus, we can find the relevant user context by adapting the training process according to the context we find.
- It is possible to find out, if someone has worse results in certain groups of the task and if someone has better results in other groups of tasks related to others. As a result of an analysis, we can change membership in qualification groups.
- It is possible change the *extended process definition* based on the analysis. For example, we can change a value in conditions important for dealing with a user as with an expert.

There are more reasons why it is helpful to analyse the properties of *business process management system users*.

The need for this kind of analysis is obvious, because the results of processes are significantly impacted by *business process management system users*. The problem of having less knowledge about particular *business process management system users* is even worse in process-oriented management, because we try to reduce the hierarchy as much as possible which means that managers are losing feedback to workers.

11.13 Summary and Discussion

The goal of this section has been to describe an idea of extension of the *business process management system* based on *workflow management system architecture* in order to make better management of its users.

The extension improves modelling abilities by describing *business processes* more precisely and thus the *business process management system* deals with its users more individually.

The presented solution lies mainly in changing the *process definition* and managing *dynamic profiles* of *business process management system users*. The way of using the current *business process management system* to manage *dynamic profiles* was also presented which is important for the presented extension.

Using *dynamic profiles* also enable us to change the algorithm for the assignment which supports the dynamic reconfiguration of priorities related to the process. The extended model also gives us the possibility of analysis which can answer several important questions for the future evolution of *business process management system users* in a company. The solution was presented as a modification of the architecture presented by the Workflow Management Coalition. The changes in the *process definition* were briefly shown on *Petri Nets*.

Chapter 12

Optimizing Business Processes Using Workflow Management System

There are many events that can significantly influence the result of a process. Unexpected events are caused by changes in the environment of the process, which are not covered by *business rules*. Thus, it is also necessary to deal with the context of a process. Since the environment of a process is variable, it is usually necessary to change the logic (routing rules and related tasks) of the process.

This is usually done by replacing one part of the process by another one that produces adequate results. For example, if the input resources of a subprocess are temporarily unavailable (it may be possible simple switch to another resources, however, there are situations when performing is limited to unique resource), a need will arise to change the logic structure (e.g. by outsourcing the particular part of the process).

The *process execution* is also limited by certain constraints such as costs, time or success rate which may be different in every moment of the evaluation. Thus, the logic of the *process instance* should also be adapted to possible changes in these attributes that have an influence on the selection of *process variants* which will lead to the desired results (outputs). Each *process model* should lead at least to one desired output, some processes may lead to similar outputs that it can be possible to use one as the substitute for other process. A decision, which substitute is the best, very often depends on related context (e.g. backup plan is executed when certain circumstances are revealed).

For example, we choose a cheaper solution in cases where we have more time to create desirable results; however, the final result can be the same or substitutable. In order to choose the optimal substation of process, there is a need for system evaluation for each substitutable component in order to choose the most suitable substitution for a context-specific request. Evaluation should be done at the moment of a planned execution in order to get the best possible data for a decision which process substitute is the most suitable in a specific time period.

The data used for the modelling of predicted behaviour of a future execution of the process substitute producing desired outputs that is possible to gain from historical performance logs.

Thus, it is possible to use a *workflow management system* to collect data from previous process performances. At the same time, *dynamic profiles of the processes* are created from the collected data. The *workflow management system* can thus work as an input data source for the component creating a *dynamic profile* of behaviour of each important existing object in the system.

12.1 How to Make Context-Aware Business Process Management System?

There are more approaches for adaptive managing systems. The challenge is to provide flexibility and offer process support at the same time [53]. Another presented solution is a combination of the *rule-based*, *case-based* and *agent-based* approaches [58].

The presented approach generates a structured process definition which can adapt to the process context. Furthermore, the internal context variables are updated in runtime due to the existing *workflow management system*. My approach is a hybrid between the *declarative approach*, *rule-based approach* and *objective oriented approach* with the ability to monitor itself and use the information to optimize process performance by the user's preferences defined by a *user query*.

The approach is also a combination of the data-driven [27] and control flow approaches. It merges subprocess definitions using interfaces based on product compatibility.

12.1.1 Description of Context-Aware Management System

The algorithm uses *dynamic profiles* describing the historical behaviour of processes (e.g., time, success rate, costs) to choose the optimal way to reach the goal of a process. In order to create the *dynamic profiles of processes* would be almost impossible without automatic tracking of processes by the *workflow management system*.

The next important step is to build a *product dependency tree*. Its purpose is to map all possible ways leading to creating the *desired product*. The *product dependency tree* can also be the result of an analysis of the *value stream management*, which tracks added value in the process.

A route in the *product dependency tree* is used to build the final structure of a modified process. The *product dependency tree* represents a product hierarchy. In other words, it reflects which income products are needed to produce the *desired product*.

The presented method has several advantages. It allow us to switch over to another way of producing the product according to changing conditions that influence the context of the process. This can be declared due to the *query approach*. Within this approach, it is necessary to define the *final desired products* and a set of conditions, which are related to the context of the process. By using *databases of processes* containing *process definitions* and related *dynamic profiles* (dynamic characteristics), it is possible to build new processes according to specific conditions. Thus, the *process instance* can be modified according to the current context (e.g., costs, time, success rate, resource problems, etc.).

All this means a great advantage for us because information about the *context of a process* is often not available before runtime, because the context is constantly changing.

Product substitutability is what makes the presented solution so flexible. The substitutability means that within a group of products needed to create a *final desired products*, one product can replace another product so that the target product characteristics are not changed.

The approach is a hybrid of the rule-based and goal-based approaches. It deals with contexts using a *top-down declarative (query) approach* using context variables. The process definition uses a structured definition of subprocesses and combines them according to the current context. The basic idea can be compared to real-time decisions in a process because managers compare the possible ways of reaching certain goals using the currently available process. If something important happens, then the process has to be rescheduled.

12.1.2 Example Illustrating a Context-Aware Approach

- Let us assume the *desired final product* is lunch. There are several substitutable ways how to get lunch, for example:
- going to a restaurant,
- making lunch at home,
- or hiring a cook, etc.

Each of these options (like each product) desires performing certain subprocesses. For example, having lunch in a restaurant requires making a reservation and going to the restaurant. To make lunch at home requires buying the ingredients at a shop, and preparing the meal. Under certain circumstances, having lunch in a restaurant or lunch at home, including the subprocesses connected with both possibilities of getting lunch, can be substitutable.

The suitability of substitution related to a process depends on the context of the process related to creating the final product (e.g. lunch) because the cheapest or the fastest solution is not always the best option for each case.

Current location should be considered as one of the factors in the choice of the optimal solution because it influences the price and time depending on transport conditions. The process of shopping can also involve some subprocesses. For example, the ingredients can be bought at different shops that have different prices. If the optimal way according to the current context is not applicable (e.g. temporary technical issues or resource unavailability), the algorithm chooses the second best option.

The behaviour can be compared to computer networks routing algorithms, especially the OSPF (Open Shortest Path First) algorithm. This kind of approach adds fault-tolerant mechanism at a level of the business process logic.

If only the *rule based approach* is used then all the possible combinations of variables have to be defined. If a variant is added (e.g. buying ingredients with home delivery), it is necessary to redefine all related processes which use the subprocess and make an evaluation of the processes using the changed subprocess. The process can have a different priority, thus it is not possible to simply change the subprocess in each related process.

In the *case based approach*, it is possible to adapt a *process instance* according to the current context, but each kind of change has to be defined manually, for example for price changes, each supplier can use a different order process. Overloading due to continuous change can eliminate the advantage of the *case-based approach*.

The presented approach shows us the way to optimize the *process definition* by using data captured by the *workflow management system* for managing *dynamic profiles of processes*, and on how to use the *declarative(query) approach* for context-based online adaptation.

12.2 Top-down declarative (query) approach to business process management

The *top-down declarative (query) approach* to *process definition* is similar to real-life planning. It is necessary to set the objectives (goals) of a project first and then find out the way of reaching the objectives. Thus, the final objectives generates other objectives. Each objective can be fulfilled by performing the process.

There are three basic levels of management: *strategical*, *tactical* and *operational levels*. First, the main objectives have to be set by user's decision. The other objectives should support the primary goals. Thus, the final objectives generates other objectives which can generate other objectives, etc. Finally, the particular tasks are handled by particular users at the *operational level* of management.

The approach is also frequently used in the field of manufactured production. The production process related to production of final products is decomposed into several subprocesses (e.g. ordering material is a typical subprocess). The approach is usually used for the calculation of expected costs. Different strategies can be chosen in the production of desired products – all the processes needed to produce the product that can be performed within a company or certain processes or their parts can be outsourced. There are several strategies in making a product, at least it is often possible to produce or outsource it. A combination of both of the approaches is usually used.

The *top-down declarative (query) approach* is useful for online adaptation of business process management because a process can be defined as a set of conditions related to the changing environment and the final structure of the process is composed according to the conditions in a particular context. The new process is feasible as there is at least one realization of subprocesses generating interchangeable products (a subprocess generating interchangable products can substitute each other) for each required input needed for the process of the highest level of abstraction. The process of the highest level of abstraction is derived from the desired *final products*.

It is also possible to use the *top-down declarative (query) approach* even for analysing the constraints based on resource properties. As a result, a process structure can be composed from such subprocesses that are custom-made with minimum effort. The properties of resources influence the performance of the process. For example, the best resources are usually the most reliable with respect to a successfully finished accomplishment of the desired task, but in most cases, they are not usually the cheapest.

The commonly used declarative approach in process modelling is described in [20][44] [46]. However, the *top-down declarative (query-based) approach* is focused more on a hierarchy of objectives. The approach is similar to database queries. The generated process can be launched in a rule-based engine *workflow management system*. The *top-down declarative (query) approach* is useful as an analytic tool for evaluating the variants of reaching the desired objectives.

12.3 Prerequisites for business process optimizing

In order to keep the logic of a whole process, it is necessary to keep the transformation of the process structure according to certain constraints. The optimizing method requires that each process has to produce at least one product. This is not an unusual requirement because processes are primarily optimized to produce a specific product (see *value stream management*). The product created by the process can also be information (e.g. where to find a shop offering the lowest price or at least confirmation that something is done). A process can contain several subprocesses. The process often needs some other products to run.

There are several ways of obtaining a certain kind of a product, because there is a possibility of obtaining a substitute for the desired products in many cases. Nevertheless, each way of obtaining the substitute of the desired product can be different in certain properties. Examples of such properties are the time needed to get the product, costs paid to get the product, need for other resources and special requirements for the product.

It also has to be mentioned that there are several ways to make the same product in organizations. It is possible to demonstrate the situation by means of a simple example. A new table is desired. There are several options to get a table. It can be bought at a shop, or someone can be hired to get it. Another option is to make a table according to a table construction scheme.

The first option will probably be more cost-effective than the second, but it will also be more time-consuming as some time has to be spent on choosing a table and comparing prices before the ideal table is found. The second option is the least time-consuming but someone (e.g. an interior decorator) has to be paid to find the best table according to the requirements. The third option is preferable in cases where no concrete product is needed to be found because it can be made according to the requirements. However, some wood is needed as well as tools for work with wood, some nails, etc.

There is no general answer to the question on which of the option is best. It depends on the context of performing the process called *get the table*.

The first option is probably the most cost-effective provided there is nobody specialized in making tables to produce one. A disadvantage of this option is that some time has to be spent in order to save money.

The second option is probably the fastest if the product has been specified, but it is necessary to pay for searching, table transfer, etc.

The third option is the most cost-effective if the process is launched in a company specialized in making tables because it already has the equipment for making tables. Otherwise, buying all the equipment and material for making one single table would cost more than payment to a joiner for making the table.

The process *get a table* can be a part of a more complicated process, for example, the process of arranging a new house or business office. Therefore, one option is preferably chosen according to the context of the process performance.

The question arises about which constraints are supposed to be the most suitable for the iterative composition of the parts of the process? Let us suppose that there is a database of all possible ways to make some products. The database also contains information about the properties of each of the ways.

12.3.1 Examples of the Properties

There are many properties which can influence the performing of the process, some of them may be monitored in most cases:

- time to accomplish the process,
- process costs,
- success rate of the process.

These properties are changing according to actual performance of the process. The time to accomplish the process can be changed as a result of redesigning the process or changing resources. The cost of the product is calculated as a sum of fixed and variable costs, while part of the variable costs also depends on the working time of a particular worker.

Reliability can also be captured as the success rate of the process. It is almost impossible for the staff to manage every single part of the database of processes, but it is possible to use the *workflow management system* to do all the work.

12.4 Optimizing Using Product Dependency Tree

The aim of this task is to generate a complete process structure by specifying the final products of the process which are desired to be created. It is also possible to set some specific additional constraints that reduce the number of feasible solutions.

When a query about the final products is made, the tool tries to find the optimal solution to the process definition according to the launched query.

12.5 Main Advantages

There are several advantages of using this approach compared to commonly used solutions.

- It is easy to use. The tool only needs to have a set of desired products and constraints.
- As the process definition can be composed during runtime, it is possible to get a better solution because more up-to-date information is available.
- The process can be custom-made so that it fits the particular purpose.
- It is possible to modify the process structure according to temporary constraints which are only known in runtime (e.g. it is not possible to launch a particular task when the process is running, which may be caused by missing resources or a limited amount of time).

The problem can be defined as how to create the optimal process structure while keeping the process logic which implies that the process has to produce the same result after its transformation.

Let us analyse what can be changed within the process definition. Is it possible to make changes in the business logic of a process that will lead to making the desired product? The answer depends on the particular definition of the process. If a part of the process is changed randomly, the logic of the process will be destroyed. What can be changed dynamically in order to avoid destroying the general logic?

Which two processes can be substituted with each other? It is possible to replace the original process with a process that produces results, which can substitute for the results of the original process. A product does not necessarily have to be a physical object, but it can also be information. A process can be described as a stream of changing products which leads to the final product. An input product can be the result of an agreement process as well.

In order to make any queries, it is necessary to have a database containing sufficient information. The approach of the following product value chain is typically used in *value stream management*.

12.6 Database of the Dynamic Profiles of Processes

The database of processes has to be provided with some specific additional information that is required for an automatic change in the composition of the original *process definition*. As the *process definition* is continuously updated, the optimal solution may be changed from the previous version during runtime. The costs and time variables can be changed due to better training or equipment.

There are many possible changes in the process environment that consequently influence the optimal solution, such as changes in the inner structure of the process, constant changes in the process characteristics (time, costs, success rate, etc.), choice of a different alternative of making the same product, updated requirements on the final products, etc. It is also very useful to have several scenarios prepared that can be used for a simulation of a potential evolution of the system, for example, an increasing amount of products dependent on an increasing demand for them.

12.6.1 Content of the Database of Processes

The *process item* has to be enriched information which enables possibility of creating and updating a process orchestration with *process items*. Thus the *process item* contains several records:

- The process definition, which was designed for producing the desired product.
- A set of places in the process definition which is used for the purpose of merging several processes into one.
- A set of input products/events is important for finding other dependent processes.
- A set of output products is necessary for finding processes, which are using products as an input.
- *Dynamic profile of the process* such as runtime details likes the performance time, costs (calculated from fixed and variable costs depending on time), success rate, etc.

The *process definition* can be described by high-level Petri nets. The *process definition* contains a plan for performing tasks. The hierarchy of subprocesses can be modelled by a hierarchy extension in high-level Petri nets.

It is necessary to specify a set of places which enables the process to be bound with other processes modelled by Petri nets. The set of places will be used for the synchronization of the output and input products.

A set of input products represents the prerequisites which have to be satisfied as a condition for successful completion of the process. Input dependencies can be demonstrated in following example.

It is necessary to *buy a table* and then to *take it home*. *Buying the table* is an input event for the *transfer of the table*. An input product for the process called *transfer of the table* is the *purchased table*. An output product of the process *purchase of the table* is the *table*. The product *table* has to be used for the purposes of synchronization.

The *dynamic profile of process item* describes the previous behaviour of the *process item*. The runtime details represent a set of general properties which can be used for optimizing purposes (e.g., costs, time, success/failure rate). The *workflow management system* can be used for managing this kind of information. It will be described in more detail in the next section.

12.7 Managing the Dynamic Properties of the Activities

In order to support the *dynamic updating properties* of the *database of processes*, it is possible to use the *workflow management system*, which can collect input data for processing in order to obtain the up-to-date profiles (sets of properties) of processes.

12.7.1 Static and Dynamic Properties of the Process Profile

In the case of *static properties*, it is possible to manage them manually. The information can be updated by filling in some forms and can be managed by administrators responsible for process performances. These properties do not rely on process performances. Examples of such kinds of properties are costs per time unit or fixed costs related to the process.

In the case of *dynamic properties*, it is very difficult to keep the correct values up-to-date. Examples of *dynamic properties* are the time to accomplish the process, process costs, or reliability of the process. In order to manage the *dynamic properties*, updating of the specific values is triggered by events, which are created by the *workflow management system*. The events happen when an *instance* of a particular process is being performed. Therefore, the *dynamic profile* of the process is updated continuously.

However, the profiles could be extracted from execution logs (if logs would contain context-based information). On the other hand, using triggers is more suitable from the performance point of view, because redesigning processes in runtime requires fast responses. The structure contains a collection of items related to a particular process. Every item holds a set of monitoring indicators, the time to accomplish the process, process costs, reliability of the process, etc., for each process. The process profile should contain at least the time to accomplish the process, process costs, and reliability of the process. The time to accomplish the process can be calculated from the historical performance of the process.

As can be understood, the *dynamic properties* describe the performance of a particular process. They can reflect changes in the process performance. A combination of the static and dynamic properties can provide, for example, the calculation of the current process costs.

The costs depend on the time and value of the fixed and variable costs for each task in the process. The reliability of the process can be found out by counting positive and negative results of the test in the checkpoint. Results closer to the present time certainly have higher significance than a result that was retrieved a longer time in the past.

Another important property is the time needed to perform the process. It can be directly calculated from the historical data captured by the *workflow management system*.

However, to update the dynamic properties manually would be almost impossible without the automatic support of the *workflow management system*. The procedure of dealing with the *dynamic properties* is similar to concept of dynamic user profiles [31]. *Dynamic properties* can be used in the decision process as presented in [32].

12.8 Product Dependency Tree

Each activity usually has some inputs that are necessary for producing some products that can be used as inputs for other processes. The *product dependency tree* represents all possible routes on

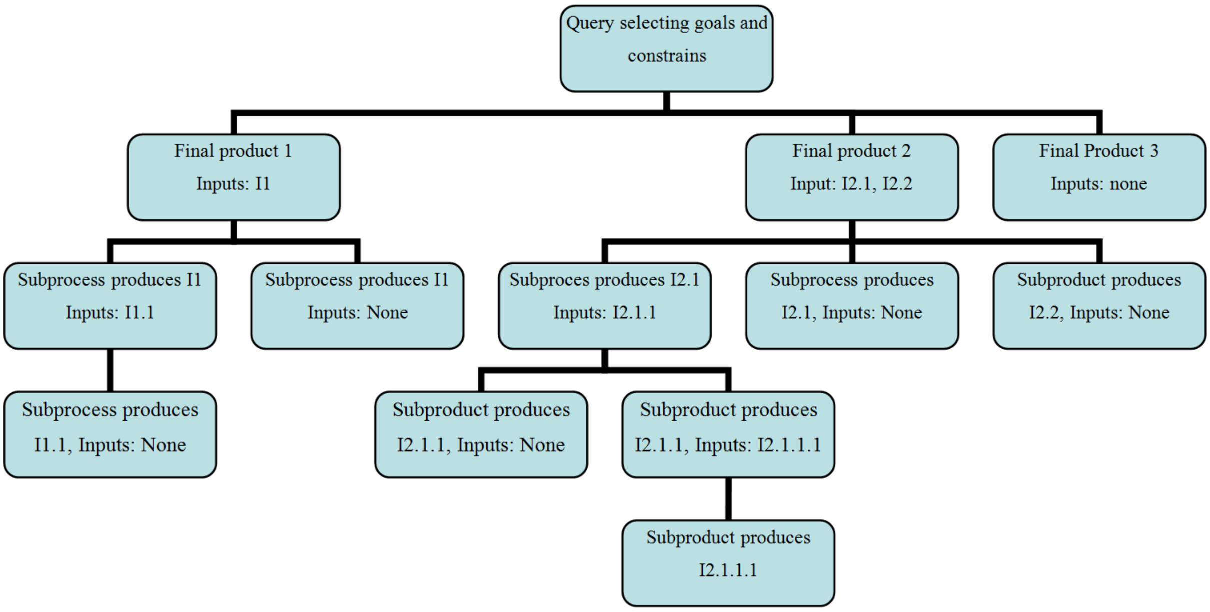

Figure 12.1: Example of construction of Product Dependency Tree

how to reach the desired final products. If we have more desired products, the query node becomes the root node in the *product dependency tree* which is depicted in Figure 12.1.

The structure of the *product dependency tree* dynamically changes according the availability of a possible solution at a specific moment.

12.8.1 Phases of the Optimizing Process Structure

The whole continuous optimizing process can be described in the following steps:

- It is necessary to build and keep an up-to-date database of processes, including their *dynamic profiles*.
- The *product dependency tree* based on a database of processes has to be built. It can be done in runtime.
- When the previous steps are accomplished, it is possible to find the optimal solution respecting the selected constraints. The constraints are declared by user query.
- The optimal solution is then to find all processes and their subprocesses within the *product dependency tree* that are needed to create the *desired final products*. These are then merged into the final process that produces the same results as the original process.

The main idea of the algorithm is based on connecting related input and output events. Each event has to belong to a particular group. The events, which are placed in the same group, can be substituted with each other.

12.8.2 Types of Processes in Product Dependency Tree

The *product dependency tree* contains several types of process in its nodes:

- Leaf process – the process does not depend on any other process, it has no input event that has to be produced by any other process. The process only uses resources directly.
- Root process – the process is derived from the final desired product. In the case of more desired products, there are more root process nodes.
- Regular process – the process that has subprocesses, but is not the root process at the same time.

12.8.3 Algorithm of Building the Product Dependency Tree

The algorithm uses items from Database of Processes for building the Product Dependency Tree. At first, it is necessary to select the desired products, these products can be produced by processes generated the desired products as their outputs. These processes will represent the highest level of Product Dependency Tree. The next step is to find all configuration of input subprocesses which produces the inputs need for the process layer created in previous step. We have to continue with searching process that fulfill the input request until there is no input request which mean we have produces the tree to the leaf level. In other words, the Product Dependency Tree is generated while there is empty set of the input request.

Thus, the algorithm can be summarized in following steps:

1. Select the desired products.
2. Find all processes that can be used to produce the products selected.
3. You will get a set of processes as the result. Search for all subprocesses that can produce inputs for the previous set of processes. Exclude the processes that were used in the previous steps (prevention of cycling in the tree).
4. Repeat until the activity is not a leaf process.

12.9 Searching the Process Definition in Product Dependency Tree

The *product dependency tree* covers all possible ways of getting the desired product. Each node of the *product dependency tree* represents one process. The relationship between nodes represents the relationship between the input and output products of the processes. The nodes in the *product dependency tree* are synchronization points and contain additional data about the historical process performances.

The algorithm follows the route from the *root process* to *leaf process*. The route is marked according to the additional properties of the process (e.g. dynamic characteristics like time, cost, etc.).

For example, there are two routes representing two options of reaching the desired final product. One option is more cost-effective but slower, another is faster but more expensive. If the slower way meets the constraints, both routes will be followed to the lower levels until the leaf processes are reached.

The higher level of the tree also covers related subprocesses so that there are already aggregated values in the higher level of the tree due to the *workflow management system*. The aggregated

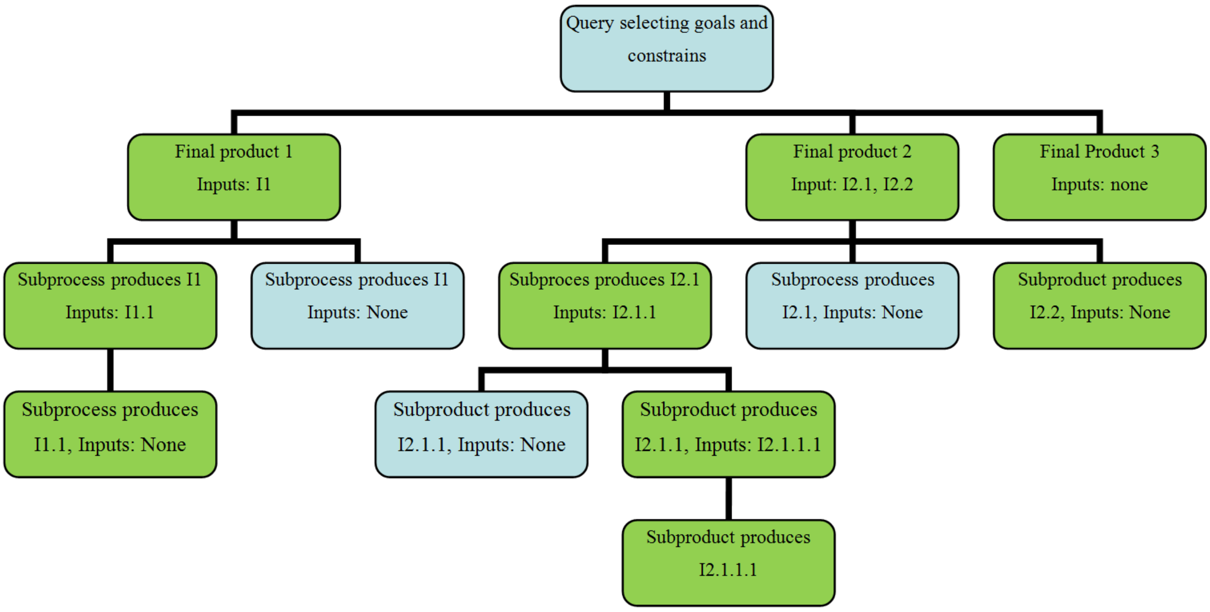

Figure 12.2: Example of searching in Product Dependency Tree

values can be used in the decision process of selecting the optimal variant of reaching the desired final product.

If several solutions are available fitting the query, the solution having the priority (e.g., costs, time, success rate, GPS of resources or priority set by the user, etc.) given by the query has to be chosen. The user query defines what the process should look like in order to be optimal (e.g., cost, time, quality).

An example of searching is depicted in Figure 12.2. The set of desired final products contains: *Final Product 1*, *Final Product 2*, *Final Product 3*.

The *Final Product 1* needs input *I1* which can be obtained via two subprocesses. The first one needs input *I1.1*, the other one is already the *leaf process*.

The *Final Product 2* needs inputs *I2.1* and *I2.2*. The output *I2.1* is produced by two processes. The output *I2.2* is already produced by *leaf process*. The output *I2.1.1* can be produced by two subprocesses. The first one is already the *leaf process*, the second one needs the input *I2.1.1.1* which can be provided by one *leaf process*.

The *Final Product 3* is already the *leaf process*.

The searching algorithm selects just one variant (marked as green) leading to getting the desired final products. The decision rule for variant could be time, cost, etc. An example of query will be described in next section.

12.9.1 Example of a Query for Selecting the Optimized Process Solution

The process can be built using process items from Database of Processes. It is necessary to describe the desired products with all constrains reducing the number of process orchestration variants.

The query can look as follows – select a process in which the desired product is 1 bread or 10 rolls, the time to perform the process is less than 10 days, and the maximum costs are less than 2 Euro. Choose the cheapest process realization.

In order to accomplish this aim, these steps need to be followed:

- Focus on the *root process*, which is derived from the product (1 bread or 10 rolls).
- Select the routes that satisfy the constraints of time shorter than 10 days and costs not higher than 2.
- The process must contain the subprocess with product "wheat."
- Order all the processes by costs.
- Choose the first process from the list.
- Make the dynamic substitution of the substitutable parts in the process definition.
- Launch the modified process by the *workflow management engine*.

The query can be described in a language which is similar to regular database queries, thus a SQL-like language illustrates how it looks like. We use that kind of language because we query the Database of the Processes where we get items for the process configuration specified by the query.

```
SELECT p /* a process variable name */
FROM processDatabase /* the source for the query */
/* final product definition includes the amount */
WHERE p.finalProducts IN (1 bread, 10 rolls)
/* constraits for context variable */
/* based on historical performance of process p */
  AND (p.term <= 10 days)
  AND (p.cost <= 2)
/* defining constraints for subProduct */
/* which the desired product should use */
  AND p.subProduct IN (wheat)
/* order the results which meet the previously specified */
/* conditions  according to their costs */
ORDER BY p.cost
LIMIT 1./* limiting the number of results */.
```

The result of a query is ready for the launching process for getting the desired products. There are many context variables which could be used in a query, the variable can be user-defined and depend on specialization of the process (e.g., data from sensors - temperature, humidity, etc.), but time, costs, location, role of person starting the process are general variables.

If it is necessary to change the priority of a process, then it should be done simply by changing the query. If the variant of making a substitutable product is chosen, then the algorithm has to re-evaluate the process just before launching the subprocess. The final process will be generated according to the context values in runtime.

Even if the number of variants of making the desired product is the same and the query remains the same, the historical performance data is still changing. Each change means checking the process if it is optimal according to the current situation.

12.10 Experiments

The *workflow management system* which will also support the adaptation of processes for changing conditions is currently being implemented.

Research has been made based on data which describes changes in the project scheduling. The data retrieved within the research covered only a low number of projects, but a lot of causes of rescheduling of individual projects were similar and the following solutions were similar too.

The outcome of the research is that almost every project is rescheduled due to missing the deadlines of the subprocesses in the projects or due to a temporary unavailability of resources. After detecting these kinds of problems, the solution was to outsource part of the projects.

Depending on the context and priority of the projects included in the research, the suppliers were usually selected according to their success rate of meeting the deadlines, sufficient quality, experience and cost prices depending on the context and priority of the projects.

The integration of the adaptation into the *workflow management system* is, therefore, expected to reflect and react to natural changes in processes which occur in practice.

The concept of the *product dependency tree* is currently being tested on the dynamic selecting of the most suitable supplier. The concept of the *dynamic profile* is being tested for scheduling purposes on data of over one million records. The *dynamic profiles* are solving the problem of individuality of the objects in the system (e.g. people in the construction section have a different productivity in making some operations).

The work for a shift should be planned with respect to awareness of the particular individual person's shift workers and their productivity. The solution uses the dynamic profiles of a particular person. The profile is created from data that is collected by a bar code reader that monitors the amount of submitting work done by the person by the bar code reader. The method is more precise than a calculation based on standards. Another important side effect function of the method is that it is possible to monitor abnormalities or trends in the behaviour of a particular person.

One of the further useful usage implementations of the *dynamic profiles* is monitoring the relationship between a production machine and the properties of its products. The problem lies in the huge dispersion of time for similar operations. Creating the *dynamic profile* for each relation between a task and its input product helps to improve the process due to splitting one step in the process into several steps, depending on the input product for the machine.

Due to the manufacturing execution system, data about the operations tasks will be continuously processed so that the process model will still be kept up-to date with all the runtime characteristics. The *dynamic profiles* of tasks and processes also make it possible to report abnormalities immediately during production. The detection of a possible failure, before it occurs saves a lot of money. More information about research in manufacture can found in [49] and [48].

12.11 Conclusion

The process management system should consider the context of a process. The presented approach aims to describe the idea of a context-aware extension of the *workflow management system*.

Adaptation of the *process definition* is based on replacing the substitutable parts of the process with its other parts. The *process definition* can also be adapted for the particular *workflow management participant* according to its properties.

The original *workflow management system* was used for maintaining the *dynamic profiles*. The *dynamic process profiles* were used for making decisions in the *product dependency tree*. The extension can solve the problem of the automatic redesigning process according to the continuous changes in the context.

The *dynamic profiles* of a *workflow management participant*, with an extension of the process definition, were used for adaptation of the process logic for the particular *workflow management participant* according to his previous behaviour in the *workflow management system*.

The context-based approach is very helpful in dynamic rescheduling. The generated process is applicable because only substitutable parts can be replaced with each other which have merging interfaces. The solution also supports fault tolerant managing processes and includes risk management into the casual managing processes. The solution is also suitable for *business process reengineering* because it can simulate and compare the old and new versions of a process because changing one part can influence the whole process.

The dynamic process structure allows for higher flexibility of the process management in runtime. The online adaptation is important, especially for processes that are not easy to manage because of a high level of changes in scheduling during runtime. But the *dynamic profiles* can also solve the problem with their unique behaviour in manufacturing processes, as was briefly described in [49].

The *dynamic profiles* of *workflow management participants* can be used for an extension of assignment tasks to the users. An assignment procedure can be precise because of the possibility of comparing the requirements with the values in the *dynamic profiles*.

It also allows some analysis of the *workflow management system*. It will enable a specific action to be triggered if the object starts to behave unusually. The *dynamic profile* of the relationship between a product and a task can be used for an improvement in time scheduling.

Chapter 13

Modelling of Context-Adaptable Business Processes and their Implementation as Service-Oriented Architecture

The goal of the chapter is to provide a description of a system capable of automatic adaptation of managing logic. Its logic is continuously adapting to optimal performance according to the user-driven and environmental context. The system observes itself and is based on these observations that it changes realizations of implemented business processes in order to keep the optimal and fault-tolerant performance of the system. The concept is presented on service-oriented architecture which demonstrates the applicability my approach to the area of distributed systems.

The context-adaptability is an important and desired feature, especially for processes situated in dynamic environments [58]. Its adaptability to the environmental changes allows it to reschedule a process (i.e. to use its new realisation which is optimal for the environment and meets the same process goals and produces the same outputs as the original realisation). The rescheduling helps to keep fault tolerant processes with an optimal configuration.

Business processes automation and IT support is often implemented by means of service-oriented architecture (SOA) [28]. A business process can be managed via a business process management system which can use SOA as an implementation architecture of the business process or its part. In SOA, a software system implementing the business process or its part is decomposed and distributed in the form of autonomous, but cooperative components, known as services. This architecture has many advantages, such as better scalability and fault-tolerance. Moreover, SOA principles, such as loose coupling, statelessness, or reusability, allows its easy runtime modifications of a composed system by changing its particular components [25].

However, realisation of the context-adaptive processes in SOA brings up several issues that need to be addressed. The rescheduling of a business process requires a strong knowledge of its context at both the business and implementation levels, which includes an administration cost of the rescheduling; QoS requirements for the process, its sub-processes, and activities; performance indicator values for current and potential realisations of the process; etc.

The goal of this chapter is to provide a conceptual approach to description and realisation of context-adaptable business processes by the service-oriented architecture. In the chapter, a model is proposed for both description of the adaptable business processes as well as description of their potential realisations and implementations, including performance metrics. I also propose a method of automatic evaluation and assignment of process definitions to their best process realisations based on an observation of the actual context and historical and current values of the performance indicators.

13.1 Context-Adaptability in Business Processes

In business process modelling, there is a strong need for the context adaptability in order to build flexible processes. These processes should be able to adapt to changing conditions during their performance time, to keep their optimal performance despite unreliable resources, or to provide secure fault-tolerant solutions. The adaptability can mean rescheduling a *business process* completely or to select one of many process variants on how to achieve process goals, which may be very complicated and time consuming.

The context of a process has a significant influence on its performance. According to previous research [49], [50], the context-dependency is very important for production processes as well as for project-oriented processes. In a factory production line implementing a production process, particular information is often available in runtime only or just before runtime of the process (e.g. an actual delay or availability of resources of which a specific product will be processed). The project-oriented processes (i.e. the processes controlling a project) are highly context-dependent on availability of resources, time, and cost, with a significant influence on process performance.

In both cases, potential rescheduling of the processes may be very expensive in time and resources. Therefore, it is useful to have a predefined set of variant solutions on how to achieve specific process goals. In order to decide which variant is the best for achieving particular process goals, all available data about the previous performance of a process has to be gathered continually and interpreted as fast as possible, to be able to represent the actual state of the context, and to be evaluated periodically according to many aspects. This data processing and evaluation can be done automatically, which will be described later in this chapter; however, data types, sources, and implementation methods, as well as aspects for evaluation, have to be defined manually.

The related work typically addresses different phases of a business process life-cycle, namely modelling, execution/management and measurement [34]. The presented approach combines these phases in order to perform evaluation as soon as possible and to use results of the evaluation for adaptation of a process definition at runtime.

There are several approaches in modelling adaptive business processes. The challenge is to provide flexibility and offer process support at the same time [53]. Most of the approaches (e.g. BPMN, XPDL, Petri Nets, etc.) focus on the modelling of processes in an imperative way which is not as suitable for changing process structure in runtime as declarative approaches (see [19, 43]). There are also declarative approaches (e.g. Declare [46]) which can provide a certain amount of freedom in the performance of the process, because the rules can be easily added even during performance time.

However, in Declare, the rules are not very suitable for processes which have to be controlled because of too many variations on how to perform the process (e.g. in banks or insurance companies), while avoiding the rules that may have significant consequences.

Execution of business processes in the second phase of their life-cycle can be controlled by two types of business process management systems: rule-based and case-based. The *rule-based systems* manage business processes by predefined rules describing the way to react in particular situations (e.g. in the case of a new production order). These systems are based on best practices and supported by many well-established approaches (e.g. for monitoring of the rule-based managed processes in the measurement phase).

However, the *rule-based systems* are not very suitable for dynamic modifications of the processes at runtime according to context changes which cannot be described by the predefined rules. Existing approaches (e.g. [11]) focus mostly on the updating of a process to its new version during its runtime, which could take days or even years without any possibility of stopping the process for an update.

The *case-based systems*, the second class of process management systems, manage business processes on the basis of previous experiences in the form of cases describing the process variants. These systems are very flexible, because the process definition is fully-specified in all its variants of individual cases and it is always possible to introduce a new case according to current needs.

However, there may be a problem with monitoring and controlling of the process performance in the measurement phase which may be very different in each case. This can lead to breaking business rules, problems with performance evaluation, and an unpredictable quality because of too much freedom in the processes.

13.1.1 Outline of the Proposed Approach

In the presented approach, the way to describe a business process is proposed, including its adaptability to control the process automatically taking into account its context information, and to adapt the process at runtime based on continuous measuring of the process performance.

In comparison with the related work which has been described in the previous section, way is proposed *a hybrid modelling approach* to model business processes in both a declarative and imperative way. The approach allows for declarative modelling of business processes as high-level abstractions describing process goals; however, components representing realisations and implementations of the business processes can be modelled in an imperative way (e.g. by workflow or orchestrations).

The presented approach is also *a combination of rule-based and case-based process management systems*. The approach allows us to control variants (the cases) of possible realisations of the business processes by predefined business rules. The realisations corresponding to a given business process are found by the *goal-based approach* when each realisation has to fulfill goals required by the business process.

The following sections describe business process modelling and management at two different levels for two different groups of users: managers or process analysts, and business process engineers, IT analysts, or IT architects.

Managers or process analysts often look at a business process in terms of what will be the result (a product) of the process and what is necessary to have in order to accomplish the process (resources). They may ask questions such as “Is it possible to make the certain product under these conditions?” or “Is it possible to make the product better in these specific external context conditions?”. On the other hand, *business process engineers* are responsible for well-designed process definitions and are usually focused only on building solutions which solve particular problem, for example, by a workflow or an orchestration of IT services in the case of IT analysts or IT architects.

The basic idea can be represented by automatic real-time decisions process which will compare the possible ways of reaching certain business process goals using the currently available processes realisations and context information. The goals, which represent business process high-level abstractions, should be defined by managers, and the available process realisations should be described by business process engineers. Both goals, as the high-level process abstractions and the processes realisations form a model of context-adaptive business processes, should be described at designtime while context information is provided at runtime by an existing *workflow management system*.

Finally, the process management system in the approach is driven by *queries* defined by its users at runtime. Each query describes goals of the main business process, which produces outputs required directly by the users, not by another business process as a part of its input resources. Moreover, the query describes particular properties of preferred realisations of the process (e.g. preferred cost of its product or time constraints for its performance). The approach allows for process management system to monitor business processes at runtime and use their model, runtime context information, and the queries to adapt the processes in order to optimise their performance according to actual needs.

13.2 Modelling of Context-Adaptable Business Processes

In this section, a conceptual model of context-adaptable business processes is introduced. It defines a collection of terms (e.g. "process", "process realisation", or "resource") and their possible relationships (e.g. "the process is realised by ...") which can be used by a process analyst to describe the adaptable business processes. In this way, the model can be used as a meta-model for modelling of context-adaptable business processes.

Moreover, it defines data supporting and operations implementing the adaptability of business processes in general (i.e. the dynamic modification of the business processes according to various aspects, for example, current values of performance metrics). This section serves as an introduction and a conceptual base for Sections 13.3 and 13.4 dealing with the context-adaptability and the implementation, respectively, in more detail.

13.2.1 A Conceptual Model of Context-Adaptable Business Processes

Modelling of context-adaptable business processes has to address four aspects: process definition, process realisation, process implementation and process performance. The *process definition* describes a process from an abstract point of view and according to the process goals as a set of output resources produced by the process. The *process realisation* follows up with a description of realisation of the process by a single activity or an orchestration of sub-processes transforming input resources to the previously described output resources. The *process implementation* describes a particular implementation of the process realisation, for example, as a Web service implementing the process activity or a BPEL process implementing the orchestration of sub-processes. Finally, the *process performance* describes definitions and values of metrics related to the process realisation.

The conceptual model is depicted by the UML class diagram in Figure 13.1 and is based on the aspects described above. The process definition is represented by class `Process` and the "should produce" association with class `Resource` describing output resources produced by the process. The process definition can have one process realisation assigned.

The process realisation with its input resources is represented by class `Realisation` with class `Resource` linked by the "consumes" association, respectively. Class `Realisation` has

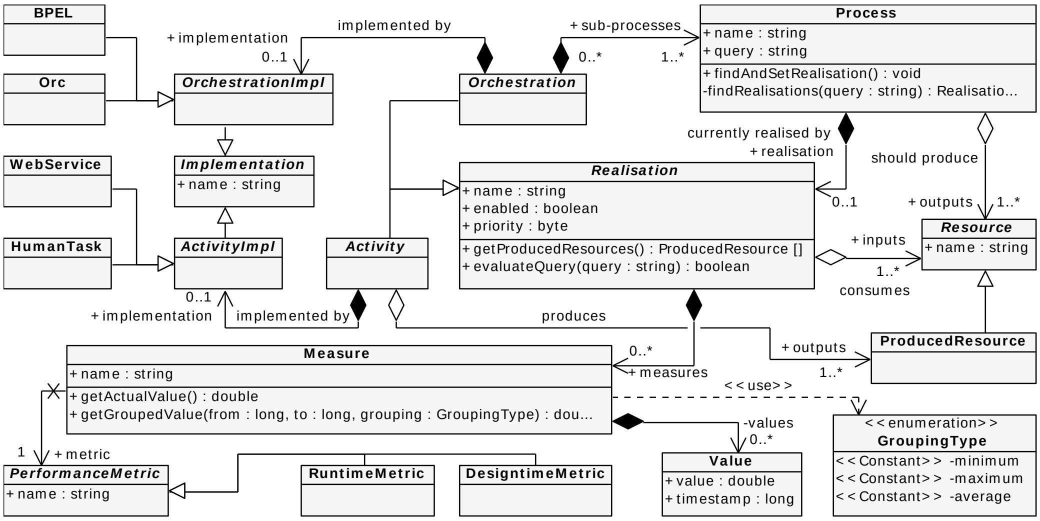

Figure 13.1: The conceptual model of the process definition, realisation, implementation, and measures, for context-adapable business process modelling.

to be specialised to class `Activity` representing a single activity, or to class `Orchestration` representing an orchestration of sub-processes (instances of class `Process`).

The process realisation also produces output resources, more specifically, the same output resources as the corresponding process definition describes. In the case of the activity (i.e. class `Activity`), these particular output resources are represented by the "produces" association with class `ProducedResource`. In the case of the process orchestration (i.e. class `Orchestration`), the output resources of the orchestration are the resources produced by its all sub-processes, that is instances of class `Resource` associated with an instance of class `Process` for each sub-process of the orchestration.

The process implementation is represented by class `Implementation` and its specialisations to classes `ActivityImpl` and `OrchestrationImpl` for implementations of activities and orchestrations, respectively. The conceptual model takes into consideration orchestrations described in BPEL or Orc language and activities implemented as Web services or as human tasks[1]. These implementations are represented by classes `BPEL`, `Orc`, `WebService`, and `HumanTask`, respectively.

Finally, performance of the process realisations can be measured by class `Measure` which is linked to information on a metric represented by an instance of classes `RuntimeMetric` (e.g. a supplier's response time) or `DesigntimeMetric` (e.g. a set-up cost of the first production cycle) and a list of measured values represented by instances of class `Value`.

13.2.2 Modelling of Context Adaptability

In the conceptual model from the previous section and Figure 13.1, the context adaptability is represented by the separation of the (abstract) process definition and the (concrete) process realisation. A business process (class `Process`) can adapt to the context by selection of its appropriate realisation (class `Realisation`), so there can be different realisations for different contexts. Furthermore, the selection of a particular realisation appropriate to a given process definition is

[1] The conceptual model is not limited to the listed implementations of orchestrations and activities (i.e. other implementations are possible, for example, an orchestration by the SCXML description, a "business rule" activity, etc.).

done on the basis of output resources. More specifically, output resources produced by the process realisation have to include output resources prescribed by the process definition. The produced resources can be obtained by the method `Realisation.getProducedResources()`, while the prescribed resources are linked by the "should produce" association between classes `Process` and `Resource`. Method `Realisation.getProducedResources()` gives all instances of `Resources` which are produced by an activity (for a process realised by the activity) or by all sub-processes participating in an orchestration (for a process realised as the orchestration of its sub-processes).

The mapping of a process definition to an appropriate process realisation is carried out for each single execution of the process. The appropriate process realisation has to produce prescribed output resources and to meet a user-defined query indicating preferred values of given metrics. The process realisation selection and execution is represented by method `Process.findAndSetRealisation()` and proceeds as follows. At first, all convenient process realisations are found by the method `Process.findRealisations()`. These realisations have to be active (attribute `Realisation.enabled`), must produce satisfactory output resources (method `Realisation.getProducedResources()`), and their measures have to match actual or grouped values stated in the query (method `Realisation.evaluateQuery()`). Finally, the convenient process realisations are ordered by their priority (`Realisation.priority`) and the first of them is selected and executed.

13.3 Process Realisations and a Product Dependency Tree

In the case of more variants as a way of achieving a business process's goals by its various process realisations, which is especially typical for project-oriented processes, one of the variants has to be selected according to the desired conditions. However, it is often a problem to evaluate which variant is better, because the decision could be very complex. In order to make the decision as good as possible, we need to evaluate the possibility of mapping the business process to all its possible realisations, and also to all possible realisations of all their sub-processes in the case of decomposition by orchestrations. This can take a lot time which could cause a delay in the process performance.

In this section, we describe the construction of a product dependency tree which maps business processes to their best realisations through the whole hierarchy of process decomposition. The tree is based on a query (see Sections 13.1.1 and 13.2.2) and on process realisations defined by business process engineers. The best realisation is selected by its produced resources (the desired products) which meet the goals of a given business process and by preferences stated in the query.

13.3.1 A Product Dependency Tree

The *product dependency tree* is defined by a directed acyclic graph $G = (N, E)$ where each node $n \in N$ contains a set $n(p)$ of business process abstractions and a set $n(a)$ of process realisation that are activities (the realisations which cannot be decomposed, contrary to orchestrations).

The root node of the *product dependency tree* contains an empty set of activities, $n(a) = \emptyset$, and an elementary set with an abstraction of the main business process only, $|n(p)| = 1$, i.e. the process which produces output resources consumed directly by its user, not by other processes (it is the top process in the hierarchy of process decomposition). Each leaf node of the tree contains an empty set of business process abstractions, $n(p) = \emptyset$, and a non-empty set of activities, $n(a) \neq \emptyset$.

Each edge $(n_p, n_c) \in E$ in the tree, which connects a parent node n_p to one of its children nodes n_c, represents a set of mappings of all process abstractions from the parent node's set $n_p(p)$ to a particular combination of their realisations, which have business process abstractions of all their sub-processes in set $n_c(p)$ of child node n_c, in the case of realisations by orchestrations, and all activities in set $n_c(a)$ of child node n_c in the case of realisations by activities.

In other words, each non-leaf node of the product dependency tree represent mappings of all process abstractions at a particular level of the hierarchy of process decomposition to a particular combination of their realisations. Then, each branch in the tree (i.e. a maximal path in the tree from its root to some of its leaf nodes) represents a particular and complete realisation of the main business process through its whole hierarchy of process decomposition, the desired result of the process management system. The mapping on each edge of the tree is done by matching, or a dependency of, goals of business process abstractions and produced output resources (the products) of process realisations. Because of the dependencies of the products, the tree is called "the product dependency tree".

13.3.2 Building the Product Dependency Tree Based on a Query

The query declares the final products of the main business process whose realisation is required (e.g. a product shipping order), available input resources which can be consumed by the process, and preferences and constraints for the resulting realisation (e.g. shipping cost limits and a maximal delivery time).

The resulting *product dependency tree* should provide, in one of its branches representing all possible ways on how to achieve the goals of the products, the best realisation of the main business process through its whole hierarchy of process decomposition. The realisation has to provide the required products, consume input resources defined in the query only, and to meet desired preferences and constraints according to the query (e.g. an optimal shipping order at low cost and minimal delivery time).

In order to build the *product dependency tree*, each business process participating in a hierarchy of decomposition of the main process, including the main process itself, has to be defined as a process abstraction with a set of produced output resources (products delivered by the process, or at the least, information that the process is finished).

Moreover, each process realisation which may participate in the *product dependency tree* has to be described by its activity or orchestration, a set of produced output resources (products of the realisation), and a set of consumed input resources. If a process realisation needs some inputs which have to be provided by other processes, there is a dependency between the current process realisation and other processes providing inputs as their products.

Finally, each process realisation has its performance profile assigned with metrics (e.g. the performance time, costs calculated from fixed and variable costs depending on time, success rate, performance indicators of Web service implementations, etc.). The performance profiles are continuously monitored by the process management system and describe behavioural history of all process realisations in the system.

The dependency tree contains all possible ways of getting the desired output resource produced by the main business process. The tree $G = (N, E)$ is built from a query by a top-down approach as it follows:

1. The root node is created and its $n(p)$ contains an abstraction of the main business process as declared in the query.

2. Each leaf node n with a non-empty set of process abstractions, $n(p) \neq \emptyset$, is analysed. For each possible combination of process realisations which can be mapped to the process abstractions in set $n(p)$ and meets constraints and conditions in the query, a new child node n_c is created.

   A process realisation can be mapped to a process abstraction iff output resources which should be produced by the process abstraction are a subset of output resources which are produced by the process realisation (i.e. the process realisation provides products defined by the process abstraction).

   For each process realisation which is an orchestration, process abstractions of all its sub-processes are added into set $n_c(p)$. Moreover, each process realisation which is an activity is added into set $n_c(a)$.

3. The second step is repeated until each leaf node n in the tree has $n(p) = \emptyset$.

13.3.3 Evaluating the Query

In order to find the best realisation of the main business process through its whole hierarchy of process decomposition, we need to evaluate the previously built *product dependency tree* and the query together.

In the tree, each branch (i.e. maximal paths in the tree from its root to some of its leaf nodes) represents a particular and complete realisation of the main business process through its whole hierarchy of process decomposition. Then, all branches in the tree are evaluated according to criteria set in the query (the branch which fulfils the criteria best gets the highest value). Finally, the evaluated branches are sorted according to their value and the branch with the highest value represents the best realisation of the main business process according to the query. In the case of several branches with the highest value, the choice is done by a user-defined priority of the realisations or randomly.

The above mentioned evaluation criteria are mostly related to the process performance and to the available context information. The decision tree may be constant as long as there is no modification of the query and no changes in process abstractions and available process realisations.

However, the query has to be re-evaluated when a decision on which variant is currently the most suitable is needed, based on the actual process performance and context information.

Nevertheless, it is not necessary to re-evaluate either all branches or every single process realisation in a branch. We need to re-evaluate only those branches which are affected by changes in the process performance or actual context information. Moreover, the re-evaluation needs to be performed only for those nodes in the affected branches which are directly related to the change or precedes in the path of other nodes related to the change. Therefore, continuous re-evaluation of a query, which is necessary for business process context adaptability, may be considerably optimised.

13.3.4 An Example of the Query

A general business process usually contains a human task and Web services. A user who needs results of the process need not know the process realisation in detail and to understand the complete process structure. The realisation often depends on particular user requirements (e.g. on a response time, cost, and the quality of the process product).

In the presented approach, the requirements of a user for process realisation are described by a query, which is written in an SQL-like language. By the query, the user describes the business process he is looking for in terms of its origin (there can be several "databases" of the processes), its products, and performance indicators of its eventual realisation. The result of processing the query is the best realisation of the business process according to the query.

In order to demonstrate the presented approach, let us consider the following query resulting from a simple use case. A user would like to run a high demand computational application (e.g. a large-scale real-time network simulation which can be implemented in several different ways) by different distributed algorithms orchestrating many services of the different providers.

For example, a simulation can be performed by a group of commodity low-cost computers, however, progress will be slow and the results may be inaccurate. On the other hand, the same simulation can be performed by a super-computer which is quite expensive, however, the user will get a fast and precise response.

Therefore, based on particular requirements and available possibilities, the user can make the following query "select the least expensive implementation of the simulation as an orchestration of available Web services which produces the final product where the response time is less than 1 second, maximum cost is 2 Euro, and the precision level is 90". Formally, the query can be described in the SQL-like language as follows:

```
/* a process variable name, the source for the query */
/* final product definition (a process abstraction) */
SELECT p FROM processDatabase
WHERE p.finalProducts="the_large-scale_real-time_network_simulation"
/* constraints of context variables */
/* based on historical performance of p */
AND (p.responseTime < 1 s) AND (p.cost <= 2 Euro)
/* order the results according to costs and return the first */
ORDER BY p.cost LIMIT 1;
```

In order to accomplish the query and execute the resulting process (its best realisation), these steps have to be followed:

- Focus on the main process which is derived from the product "the large-scale real-time network simulation" and build the *product dependency tree*, as described in Section 13.3.2.
- Select a branch of the tree so that it satisfies the constraints of the response time shorter than 1 second and maximum cost is lower than or equal to 2 Euro.
- Order all suitable branches by costs (the branches represent hierarchies of suitable process realisations).
- Choose the first branch from the list by the above order.
- Assign the first process realisation in the branch to the main process and the rest of process realisations in the branch to its sub-processes according to their level in the hierarchy, as described in Section 13.3.3.
- Launch the main process (its realisation).

Various context variables can be used in a query. These can be defined by a user or a process specialisation (e.g. parameters of the desired product). However, the most important are response time, cost, location and quality of service.

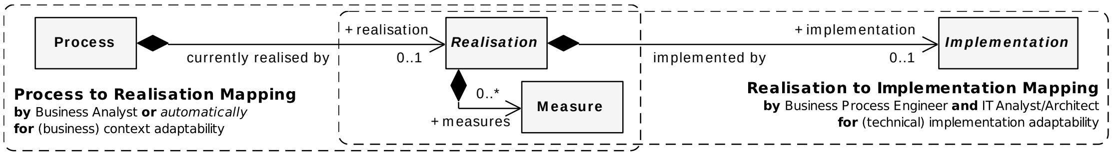

Figure 13.2: Context adaptability and implementation adaptability in the conceptual model for context-adaptable business process modelling.

13.4 Context-Adaptive Business Processes and Service-Oriented Architecture

In the previous sections, the context adaptability of business processes has been described as a dynamic assignment of a particular process realisation to a given process definition driven by user-defined criteria and by features and performance indicators of available process realisations.

So far, we have thought in terms of business processes and their decomposition and performance. In this section, implementation details of business process realisations and the discussed mapping of context-adaptive business processes to services of Service-Oriented Architecture will be taken into account.

13.4.1 Implementation of Context-Adaptive Business Processes

The context-adaptive business processes are implemented by *process realisations* which "realise" *process definitions* as described in Section 13.2.1. In the other words, each process realisation can have a process implementation assigned. The process realisation and the process implementation are different views of exactly the same business process.

The first one takes into account business perspectives (e.g. process decomposition and input/output resources), while the second one focuses on implementation techniques (e.g. service and human-task implementations).

Both views are interconnected as each process realisation specifies the purpose of the corresponding process implementation. Moreover, the type of process realisation, which can be an orchestration or an activity, affects its implementation, which can be a BPEL/Orc orchestration or a Web service/human task, respectively. Analogously, the implementation determines the resources required by the process realisation (e.g. a human task may require qualified persons).

Based on the two views, we can distinguish two types of adaptability, namely *(business) context adaptability* and *(technical) implementation adaptability*, as depicted in Figure 13.2 (for the context, see also Figure 13.1).

Generally, the context adaptability is managed by a business analyst, while the implementation adaptability is solved by a business process engineer or an IT analyst/architect. In the presented approach, it focuses on the context adaptability so as to allow optimisation at the business level by automatic mapping of process definitions to process realisations. Moreover, at the business level, each realisation also represents its implementation which affects its resources and measured performance.

Therefore, for multiple realisations of the same process with different implementations, the context adaptability decides also which of the implementations will be utilised in the process because each realisation has assigned, at most, one implementation. In this way, the mapping of process realisations to their implementations does not need to be considered.

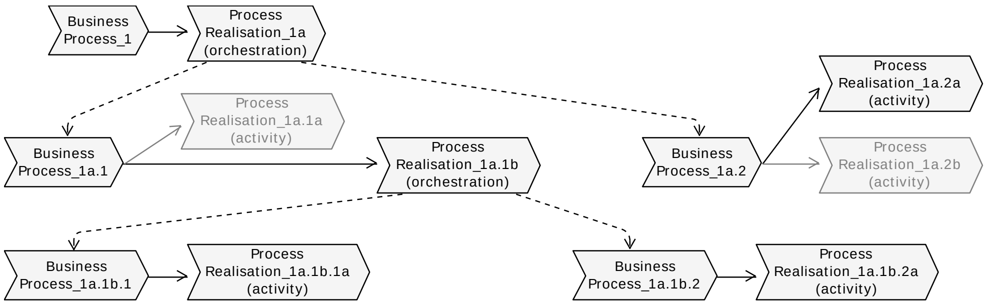

Figure 13.3: Hierarchy of business processes with mappings to process realisations (denoted by rightwards full arrows) and decomposition to orchestrations of sub-processes (downwards dashed arrows). Alternative mappings to realisations 1a.1a and 1a.2b are shaded in grey.

The implementation of context-adaptive business processes proceeds as follows. In the beginning, a business analyst defines a set of business processes and their realisations, including output (produced) and input (consumed) resources.

For each process realisation, (a set of) the output resources produced by the realisation have to include (to be a superset of) the output resources that should be produced by the corresponding business process according to its definition (see also Section 13.2.2).

The business process realisations can be found by decomposition of the main business process (a top-down approach) or by a description of all possible realisations of partial business processes and their composition (a bottom-up approach).

Each realisation should include definitions and values of its design-time metrics (e.g. set-up costs) and also definitions and initial, current, or historical values of run-time metrics (e.g. a response time) if they are available. With this information, current mappings of process definitions and their realisations can be automatically re-evaluated to find an optimal initial decomposition of the main business process. After this step, each business process is realised by a hierarchically decomposed and orchestrating sub-processes to the level of individual activities.

In the second step, business process engineers and IT analysts/architects implement the previously defined process realisations. Each realisation is mapped to its (one) implementation, which can extend or affect the previously described input resources and metrics. After that, service-oriented architecture of the system is defined, including implementation details of orchestrations, individual services and human tasks. The system is ready for deployment, testing, and a trial run.

Finally, the implemented business process realisations are continuously re-evaluated at runtime, based on actual values of relevant metrics. Before each single run of a business process in the system, its process realisation can be automatically re-assigned according to the actual needs and values of performance indicators (measured metrics).

During each time, the resulting hierarchy of mappings of business processes to their realisations (for example, see Figure 13.3) is *correct* because output resources produced by each process realisation always include all output resources that should be produced by the corresponding (mapped) business process according to its definition.

Moreover, the hierarchy is always *optimal* because each process realisation in the mappings has been selected from all possible realisations (producing the required output resources) according to their performance and user-defined priorities. The performance of the process realisation is determined not only by the performance of its current implementation, in the case of an activity, but also by performance of all sub-processes (their particular realisations) in the case of an orchestration.

13.4.2 Quality of Service and Implementation of Performance Metrics

The quality of service (QoS) and the performance metrics play key roles in context-adaptive business processes. The adaptability of business processes does not change the process goals; however it affects their QoS by its maintenance or improvement of process performance. In the presented approach, *runtime metrics* and *designtime metrics* are utilised.

The first is used to measure performance of process realisations at runtime and to evaluate their suitability as final realisations of defined processes, while the second allows us to also include into the evaluation hidden costs, such as resource utilisation or set-up costs (i.e. changeover costs associated with switching from the current process realisation to another). Based on the metrics' values, each process is able to adapt to a process realisation with the best QoS and affordable costs.

Performance metrics of process realisations are defined at design time. Together with definitions of business processes, a business analyst defines *process performance metrics* based on business goals. These process performance metrics are then translated by a business process engineer and an IT analyst/architect into performance metrics of process realisations.

Moreover, in the case of runtime metrics, business process monitoring tools have to be implemented to support runtime metrics by continuous or event-driven/triggered measurements of process realisations.

Finally, the process performance metrics are used as evaluation criteria in business process adaptability (i.e. during mapping of business processes to their optimal realisations). For example, a process performance metric states that "average duration for order processing has to be less than 60 minutes", which can restrict mapping to "automatic order processing" realisations because manual order processing cannot meet the 60-minute limit due to working hours or holidays.

In service-oriented architecture, the runtime metrics of process realisations should measure performance and QoS of individual Web services and their orchestrations. The following Web service performance metrics have been adopted from [24]: availability and reliability of the service, price, throughput, response time, latency, performance, security, accessibility, regulatory, robustness/flexibility, accuracy, servability, integrity and reputation.

13.5 Evaluation and Discussion

Potential applications providing a fault tolerant and an adaptive solution in the environment of distributed systems should address many issues, such as the lack of possibility to have some influence in order to control every part of a system which is caused by the adaptability or fault-tolerance of service providers which can be solved by redundancy in providers or alternative service realisations.

However, the process adaptability and resource sharing are mainly limited by user requirements. For example, one group of users needs fast responses at any cost (e.g. for life-critical systems) while other users may prefer lower cost. Therefore, the resulting system has to implement both process realisations, which is different for each user group according to their needs, even if both realisations provide the same products. The presented approach provides the solution which can automatically adapt to a changing environment based on user requirements.

The concept of adaptation at the business process level has been previously analysed in [33] as rescheduling of projects with a focus on causes of the rescheduling. Each sub-process has been modelled as an object with its own dynamic profiles which describe its possible behaviour.

The presented approach extends the concept of dynamic profiles from [33] to dynamic process realisations. For example, in the realisation of a business process by means of service-oriented architecture, the process would be realised as a service orchestration where orchestrated services would be realisations of individual sub-processes.

Continuous observation of the performance of these services and other user-defined metrics representing the context of the process allows us to change possible process realisations (i.e. to switch service orchestrations and implementations). The resulting system would be very flexible (adaptation to the context, easy to add new services at runtime, etc.) and would immediately be able to find how to keep optimal performance based on a user-defined query which describes the priorities for desired optimal behaviour.

13.6 Conclusion

The main contribution of the presented approach is an automatically controlled and optimal dynamic orchestration of processes in accordance with context dependencies. By means of the presented approach, the processes can adapt to changing environmental conditions by the selection of particular process realisations while keeping performance requirements. The approach deals with both the business and implementation levels of the processes. At the implementation level, service-oriented architecture is utilised, so that the processes can adapt (e.g. to changing response time, cost, or availability of orchestrated services).

The current approach can be improved. Future work will include an extension of the approach with the ability to analyse combinations of service performance properties and process input resources, because it is expected that a service may have a different performance according to its different input configurations (the current approach evaluates suitability of a service as a process representation by its performance in producing desired outputs, i.e. without any knowledge of the actual service inputs).

Chapter 14

Conclusion

In the thesis, the approach for architecture design of intelligent context-aware business process management system has been proposed. The thesis meets the objectives set out in Chapter 8. The objectives has been accomplished by adding components to the WfMC reference model.

I add these components:

- Database of Processes,
- Process Orchestrator,
- Dynamic User Profiles,
- Dynamic Process Profiles.

The thesis is based mainly on research papers: *Modification of Workflow Management System Focused on Human* [32], which describes briefly the problem of missing support of context in current Workflow Management Systems, the idea of usage Dynamic User Profiles for users is mentioned in *Using Workflow Management System for Analysis Based on Properties of Resources*[31]. These papers describe the component Dynamic User Profiles, however concept is generalized to Dynamic Objects that provides description for context-relevant subjects in the system.

The research related to importance of context was published in the papers *Process Mining in Manufacturing Company for Predictions and Planning*[48] and *Analysis Resource Performance and its Application* [49] where the importance of context was supported by research in a manufacturing company, where an hypothesis supported context relevance was tested on real historic data covering several million records. The datamining methods were used for analysing of real data.

The paper *Context-Based Adaptation of Process Definition* [33] describes the idea of architecture of business process management system which is able to change the overall process logic according to a changing context in runtime. This paper describes componets Dynamic Process Profiles, Database of Processes and query-driven Process Orchestrator.

The application of context-adaptable process definition was also presented in the area of services-oriented architecture in paper *Modelling of Context-adaptable Business Process and their implementation as Service-Oriented Architecture* [30]. This paper adapt concepts of Query-driven process orchestration to Service-Oriented Architecture which shown that architecture can be used for distributed process orchestration.

The architecture designed in this thesis has several advantages compared to current business process management systems architecture.

- The architecture provides automatic fault-tolerant mechanism.
- The architecture also supports several modelling approaches.
- The architecture has natural support for alignment with strategic level of management where there is a possibility to create to several possible strategies simply declaring the objectives and constrains related to them and the algorithm creates the Product Dependency Tree.
- The architecture supports context-awareness, the process logic can be re-evaluated during the time of binding each subprocess.
- The designed architecture supports the usage of historic data and uses them for heuristic analysis.
- The architecture can be used as an adaptive context-aware planner. This can be helpful especially in short-term predictions where there are several context variants of performance.

14.1 Supporting several modelling approaches

The most used modelling approach so far is the imperative way of modelling. However, there are other approaches (e.g. declarative way of modelling, based on symbol grammar, S-BPM, agent-based etc.).

Each approach is suitable for different kinds of processes, especially depending on the level of looseness of managing.

For example, it is very difficult to model process describing the treatment of illness in the imperative approach, because there is a huge state explosion of combinations of all activities. However, using the declarative approach would enable easier modelling such kind of processes.

Because there is no simple answer on which modelling approach is the best, the architecture should support most of them.

The designed system deals with process definition on the level of input and output interface. Thus, it simply enables us to join one modelling approach on others into a shared business process management architecture.

14.2 The fault-tolerant mechanism

In many situations, there is a need for a backup plan especially if risk issue occurs. Thus, it is necessary to continuously watch the changing environment of the process. Currently, there is a situation solved by using exception handling; however, these exceptions have to be implemented in the process definition which is very time consuming. The unified strategy of managing changing logic of the process which is context-aware is very helpful.

Changes in context influence the logic of performing. The internal and external context is constantly changing while each performing instance is in its unique state. Thus manual handling would be very complicated.

Because the system has access to historical objects from a dynamic object, there will be a high potential for a good resource usage prediction in near future. It can automatically choose the most suitable subprocesses which are able to fulfill the desired objectives.

There is an analogy to OSPF algorithm in the network routing area. If the commonly used routing in business process is temporary unavailable, it is desired to choose the other feasible routing in the business process.

14.3 Alignment with Strategic Objectives

The alignment with strategic objectives is a very important issue from the whole system performance point of view.

However, the most used methodology for alignment is often used in design time and this does not reflect changes of environment in runtime. Thus significant set up time is needed in order to make strategic changes. The huge effort in change management causes that the alignment is not done with frequency corresponding to frequent changes in environment.

The environment is changing even faster than it used to be a few years ago, thus even the strategic objectives need to change faster. If the company changes the strategy objectives then it has a huge influence on overall organization because the strategy objectives have a direct relation to the processes.

The approach presented in the thesis provides alignment to strategic objectives automatically, because it is possible to change the strategic goals in runtime or at least to make a simulation of the overall system based on projected objectives.

The query selects the most suitable variants of process orchestration while the user declares the final products and constrains related to producing the final products. The user does not need to know all process definitions at the strategic level.

However, defining the query influences the process orchestration not only in design time. The query can be changed in even in runtime. Each running instance of processes can be in a different state or it is possible to add new subprocess to the system while other process instances will be performing.

14.4 Context-Aware Adaptive Scheduler

The architecture is designed to deal with many aspects that other planning systems do not take into account.

Therefore, there is a solid base for adding support of short term planning which is very difficult. The presented architecture uses transformed historic data of performance into *dynamic objects*, it is designed to change the process logic orchestration according to the changing context.

Thus, it can improve short term prediction of outgoing actions. Each action within system is connected to particular resources. Because the system is context-driven and uses historical performance date for heuristic analysis, the prediction of an outgoing action is much closer to reality compared to systems that do not use the context-aware approach.

In papers [49][48], I tested the influence of context dependencies on real data set from the manufacture covering several millions records and it was possible to decrease the prediction deviation of performance time up to approximately 50% when the prediction was taking into account the attributes of the product. Thus, there was an important context dependency, even in the production process where every action is supposed to be standardized. The standardization, in general, should imply a low level of performance time deviation.

14.5 Future Work

The future research will be especially focused on improvement in scheduling which will be based on the current context-aware approach. Related to this topic the location dimension should also be taken into account.

The presented approach is quite general, thus it could be transferred to different areas with modification of heuristic analysis. It was applied to the Service-oriented Architecture in [30]. The approach is also suitable for distributed architecture where each node can publish the list of business objectives.

However, each node in distributed architecture can fulfill the objectives in different ways. Thus each node can provide different dynamic properties (e.g. reliability, cost, time, etc.) of production related to the desired products.

Bibliography

[1] Wil M. P. van der Aalst, Paulo Barthelmess, Clarence A. Ellis, and Jacques Wainer. Workflow modeling using proclets. In *Proceedings of the 7th International Conference on Cooperative Information Systems*, CooplS '02, pages 198–209, London, UK, UK, 2000. Springer-Verlag.

[2] Software AG. Event-driven process chain (epc).

[3] K. Agyapong-Kodua, J. O. Ajaefobi, and R. H. Weston. Modelling dynamic value streams in support of process design and evaluation. *Int. J. Comput. Integr. Manuf.*, 22(5):411–427, May 2009.

[4] Kemal Altinkemer, Yasin Ozcelik, and Zafer Ozdemir. Productivity and performance effects of business process reengineering: A firm-level analysis. *J. Manage. Inf. Syst.*, 27(4):129–162, April 2011.

[5] Bizagi. Bizagi bpm suite v10.6 - user guide.

[6] Joseph T. Catanio, Gary Armstrong, and Joanne Tucker. The effects of project management certification on the triple constraint. *Int. J. Inf. Technol. Proj. Manag.*, 4(4):93–111, October 2013.

[7] Workflow Management Coalition. Workflow management coalition terminology & glossary.

[8] Workflow Management Coalition. The workflow reference model.

[9] Hollingsworth D. Wfmc the workflow reference model 10 years on.

[10] Julian Dolby, Christian Hammer, Daniel Marino, Frank Tip, Mandana Vaziri, and Jan Vitek. A data-centric approach to synchronization. *ACM Trans. Program. Lang. Syst.*, 34(1):4:1–4:48, May 2012.

[11] Clarence Ellis, Karim Keddara, and Grzegorz Rozenberg. Dynamic change within workflow systems. In *Proceedings of conference on Organizational computing systems*, COCS '95, pages 10–21, New York, NY, USA, 1995. ACM.

[12] Albert Fleischmann, Werner Schmidt, and Christian Stary. *S-BPM in the Wild: Practical Value Creation*. Springer Publishing Company, Incorporated, 2015.

[13] The YAWL Foundation. Yawl user manual 3.0.

[14] Claude Girault and Rudiger Valk. *Petri Nets for System Engineering: A Guide to Modeling, Verification, and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2001.

[15] Daniela Grigori, Fabio Casati, Malu Castellanos, Umeshwar Dayal, Mehmet Sayal, and Ming-Chien Shan. Business process intelligence. *Comput. Ind.*, 53(3):321–343, April 2004.

[16] Daniela Grigori, Fabio Casati, Umeshwar Dayal, and Ming-Chien Shan. Improving business process quality through exception understanding, prediction, and prevention. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 159–168, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[17] W. Gruber. *Modeling and Transformation of Workflows With Temporal Constraints*. IOS Press, Inc., 2004.

[18] Arthur Hofstede, Wil van der Aalst, Michael Adams, and Nick Russell. *Modern Business Process Automation: YAWL and Its Support Environment*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[19] R. Hull, F. Llirbat, E. Siman, J. Su, G. Dong, B. Kumar, and G. Zhou. Declarative workflows that support easy modification and dynamic browsing. In *Int'l Joint Conference on Work activities coordination and collaboration*, pages 69–78, 1999.

[20] Richard Hull, Francois Llirbat, Eric Siman, Jianwen Su, Guozhu Dong, Bharat Kumar, and Gang Zhou. Declarative workflows that support easy modification and dynamic browsing. In *Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration*, WACC '99, pages 69–78, New York, NY, USA, 1999. ACM.

[21] Project Management Institute. *A Guide to the Project Management Body of Knowledge: PMBOK(R) Guide*. Project Management Institute, 5th edition, 2013.

[22] Yong-Heng Jiang, Ling Wang, and Yi-Hui Jin. Bottleneck analysis for network flow model. *Adv. Eng. Softw.*, 34(10):641–651, October 2003.

[23] Florian Johannsen, Susanne Leist, and Gregor Zellner. Six sigma as a business process management method in services: Analysis of the key application problems. *Inf. Syst. E-bus. Manag.*, 9(3):307–332, September 2011.

[24] Sravanthi Kalepu, Shonali Krishnaswamy, and Seng Wai Loke. Verity: a QoS metric for selecting web services and providers. In *Proceedings of the Fourth international conference on Web information systems engineering workshops*, WISEW'03, pages 131–139, Washington, DC, USA, 2003. IEEE Computer Society.

[25] D. Karastoyanova, A. Houspanossian, M. Cilia, F. Leymann, and A. Buchmann. Extending BPEL for run time adaptability. In *Ninth IEEE International EDOC Enterprise Computing Conference*, pages 15–26, September 2005.

[26] Kissflow. Knowledge base.

[27] V. Künzle and M. Reichert. Philharmonicflows: Towards a framework for object-aware process management. *J. Softw. Maint. Evol.*, 23(4):205–244, June 2011.

[28] Fernando Lins, Julio Damasceno, Andre Souza, Bruno Silva, David Aragão, Robson Medeiros, Erica Sousa, and Nelson Rosa. Towards automation of soa-based business processes. *International Journal of Computer Science, Engineering and Applications*, 2(2), April 2012.

[29] Lukáš Máčel and Tomáš Hruška. Bringing flexibility into dynamic process change:the process re-execution approach. In *BUSTECH 2014: The Fourth International Conference on Business Intelligence and Technology*, pages 31–38. The International Academy, Research and Industry Association, 2014.

[30] Vojtěch Mates, Marek Rychlý, and Tomáš Hruška. Modelling of context-adaptable business processes and their implementation as service-oriented architecture. *Procedia Economics and Finance*, 12(2014):412–421, 2014.

[31] Vojtěch Mates. Using workflow management system pro analysis based on properties of resources. In *DATAKON 2010 Proceedings*, DATAKOM 2010, pages 161–168, 2010.

[32] Vojtěch Mates. Modification of workflow management system focused on human. In *Modification of Workflow Management System Focused on Human*, ISD 2010, pages 479–490. Springer, 2011.

[33] Vojtěch Mates and Tomáš Hruška. Context-based adaptation of process definition. In *Proceedings of the Fourth international conference on Web information systems engineering workshops*, BUSTECH 2012, pages 25–31. IARIA, 2012.

[34] Jan Mendling. *Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, volume 6 of *Lecture Notes in Business Information Processing*. Springer, 2008.

[35] R. Milner. *A Calculus of Communicating Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1982.

[36] Thomas Molka, David Redlich, Marc Drobek, Artur Caetano, Xiao-Jun Zeng, and Wasif Gilani. Conformance checking for bpmn-based process models. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 1406–1413, New York, NY, USA, 2014. ACM.

[37] OASIS. Web services business process execution language version 2.0.

[38] Inc. (OMG) Object Management Group. Bpmn 2.0 by example.

[39] Inc. (OMG) Object Management Group. Business process model and notation (bpmn) version 2.0.

[40] Oracle. Developing business processes with oracle business process management studio.

[41] Egon Ostrosi, Alain-JéRôMe FougèRes, and Michel Ferney. Fuzzy agents for product configuration in collaborative and distributed design process. *Appl. Soft Comput.*, 12(8):2091–2105, August 2012.

[42] Workflow Control-Flow Patterns. A revised view, bpm center report bpm-06-22.

[43] M. Pesic, H. Schonenberg, N. Sidorova, and W. M. P. van der Aalst. Constraint-based workflow models: Change made easy. In *OTM Conferences (1)*, pages 77–94, 2007.

[44] M. Pesic, M. H. Schonenberg, N. Sidorova, and W. M. P. Van Der Aalst. Constraint-based workflow models: Change made easy. In *Proceedings of the 2007 OTM Confederated International Conference on On the Move to Meaningful Internet Systems: CoopIS, DOA, ODBASE, GADA, and IS - Volume Part I*, OTM'07, pages 77–94, Berlin, Heidelberg, 2007. Springer-Verlag.

[45] M. Pesic and W. M. P. van der Aalst. A declarative approach for flexible business processes management. In *Proceedings of the 2006 International Conference on Business Process Management Workshops*, BPM'06, pages 169–180, Berlin, Heidelberg, 2006. Springer-Verlag.

[46] M. Pesic and W. M. P. van der Aalst. A declarative approach for flexible business processes management. In *Business Process Management Workshops*, pages 169–180. Springer, 2006.

[47] Maja Pesic, Helen Schonenberg, and Wil M. P. van der Aalst. Declare: Full support for loosely-structured processes. In *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference*, EDOC '07, pages 287–300, Washington, DC, USA, 2007. IEEE Computer Society.

[48] Milan Pospíšil, Vojtěch Mates, Tomáš Hruška, and Vladimír Bartík. Process mining in a manufacturing company for predictions and planning. *International Journal on Advances in Software*, 2013(3):283–297, 2013.

[49] Milan Pospíšil, Vojtěch Mates, and Tomáš Hruška. Analysing resource performance and its application. In *The Fifth International Conference on Information, Process, and Knowledge Management*, eKnow 2013, pages 143–148, Washington, DC, USA, 2013. IARIA.

[50] Milan Pospíšil, Vojtěch Mates, and Tomáš Hruška. Process mining in manufacturing company. In *The Fifth International Conference on Information, Process, and Knowledge Management*, eKnow 2013, pages 143–148, Washington, DC, USA, 2013. IARIA.

[51] Manfred Reichert and Barbara Weber. *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Springer Publishing Company, Incorporated, 2012.

[52] A. Rozinat, R. S. Mans, M. Song, and W. M. P. van der Aalst. Discovering simulation models. *Inf. Syst.*, 34(3):305–327, May 2009.

[53] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. M. P. van der Aalst. Process flexibility: A survey of contemporary approaches. In *Advances in Enterprise Engineering I*, pages 16–30, 2008.

[54] Dennis M. M. Schunselaar, Fabrizio Maria Maggi, Natalia Sidorova, and Wil M. P. van der Aalst. Configurable declare: Designing customisable flexible process models. In Robert Meersman, Hervé Panetto, Tharam S. Dillon, Stefanie Rinderle-Ma, Peter Dadam, Xiaofang Zhou, Siani Pearson, Alois Ferscha, Sonia Bergamaschi, and Isabel F. Cruz, editors, *OTM Conferences (1)*, volume 7565 of *Lecture Notes in Computer Science*, pages 20–37. Springer, 2012.

[55] Alec Sharp and Patrick McDermott. *Workflow Modeling: Tools for Process Improvement and Application Development*. Artech House, Inc., Norwood, MA, USA, 1st edition, 2001.

[56] Wil M. P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer Publishing Company, Incorporated, 1st edition, 2011.

[57] Wil vanderAalst and Kees vanHee. *Workflow Management: Models, Methods, and Systems*. MIT Press, Cambridge, MA, USA, 2004.

[58] B. Weber, W. Werner, and R. Breu. Ccbr–enabled adaptive workflow management. In *Proc. European Conf. on Case-Based Reasoning*, ECCBR'04, pages 434–448, 2004.

[59] Mathias Weske. *Business Process Management: Concepts, Languages, Architectures*. Springer Publishing Company, Incorporated, 2nd edition, 2012.

[60] Branimir Wetzstein, Philipp Leitner, Florian Rosenberg, Ivona Brandic, Schahram Dustdar, and Frank Leymann. Monitoring and analyzing influential factors of business process performance. In *Proceedings of the 2009 IEEE International Enterprise Distributed Object Computing Conference (Edoc 2009)*, EDOC '09, pages 141–150, Washington, DC, USA, 2009. IEEE Computer Society.