Brno University of Technology
Faculty of Information Technology
Department of Information Systems

PhD thesis
Teze disertační práce

Query-driven Context-Aware Workflow
Dotazem řízené kontextově orientované workflow

Ing. et Ing. Vojtěch Mates

Supervisor: Prof. Ing. Tomáš Hruška, CSc.

Contents

Abstrakt (in Czech)

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.

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

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

1 Introduction

Companies tend to be more process-oriented in management style. On of advantages in this approach comparing to hierarchical model can be better sharing resources in company which also influence overall productivity. The other reason why can be process-oriented approach 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 also will increase productivity, especially for repeatable process. It also decrease error rate. There is very limited deviation from the process definition, because there have to be defined each step in advance.

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

Thus the goal of thesis is description how to add context-aware adaptability to process-aware systems, which current systems do not support. At the first, it will be describe how to get context to process-aware systems – description of creating and usage Dynamic Objects. Further, the architecture of adaptation of context-aware system using Dynamic Objects will be presented.

1.1 Advantages of Query-Driven Approach

High-level of flexibility is enabled due to mixture of several different approaches. It enables natural combination of declarative and imperative way in one process. The higher connection layer uses goal-based approach which actually can connect several proclets in one coherent process definition The whole approach can dynamically change part of process without affecting the main business logic structure. However changing the parts effects the performance of process and it enables fault tolerant mechanism similar to routing algorithm in network.

1.2 Context-Aware Adaptation System

At first, it is necessary to define what is context-aware system, it which will be mentioned in chapter How to build context-aware adaptation system. The research is structured to two main flows: Defining and representing context due to Dynamic Object and relevance of the context-aware approach and Adaption of business process logic according to changing context of system using algorithm which uses Dynamic Objects for heuristic proposes when select the most suitable variant of performing process.

The thesis is based mainly papers: Modification of Workflow Management System Focused on Human [12], 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[11].

The papers Process Mining in Manufacturing Company [19] for Predictions and Planning and Analysis Resource Performance and its application [20] are presenting importance of context supported research in manufacturing company, where hypothesis supported context relevance were tested on real historic data covering several millions records. The datamining methods were used for analysing of real data.

The paper Context-Based Adaptation of Process Definition [13] describes the idea of architecture of business process management system which is able to change the process logic according to changing context.

The application of context-adaptable process definition was also presented in area of Services-oriented architecture in paper Modelling of Context-adaptable Business Process and their implementation as Service-Oriented Architecture [10].

2 State of the Art

2.1 Process Aware Information Systems

The most current systems uses imperative approach of business process modelling, however the Declare also support the declarative approach, which I believe it is a important issue to handle loosely specified business process. Each solution can be suitable to different environment. Kissflow match the most requirements of small businesses and the cloud-based model also supports this market orientation. The Bizagi Suite is still provided as the desktop application, however, in my humble opinion, the user configuration setting is on the best level of user experience. The Yawl solution support some features which others still do not support, e.g. declarative and imperative approach in one environment. The SAP and Microsoft Solution are tighly connected to their software environment, however they are accessible via their own API and via ODATA in order work with data. The Oracle Based solution seems to be included all useful function, the process designer is accessible via web interface. The most business process solution are originated from Workflow Management System Archicture.

2.2 Workflow Management System

More information about WfMS architecture can be found in [3], and related terminology [2]. More information about BPM, which uses WfMS as the tool can be found in [23]. The Workflow Management System architecture is base architecture of other current business process management suite, there are also some extensions provided by other vendors, but the basic architecture still remains. However, I would add some extension which it 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.
- The architecture is not context-aware, which I believe is very important issue and the relevance was also shown in Case Study made in manufacture company, see parts Dynamic Objects. I believe that supporting context will lead to the new generation of the system. It can be observed some first context support in Bizagi software, see the manual [1].
- The architecture is not supporting dynamic process orchestration which I believe it will next important issue especially if the business process management system will be used. Because, there are a lot of problems with merging different version of business processes(see conformance checking [15]), because the current situation lack support for distributed business process architecture.

2.3 Approaches in Business Process Modelling

Each approach has specific attitude to model business process. I believe that the diversity of so many approaches shows that the problem is still open. The basic goal is the proper distribution of task to particular users, however there a problem with prediction of future context. There are also not known the all occurrences which will happen. There is also a problem to evaluate if the process was successful just after performance which much easier than in design time before process happens. There is also possible to make transformation between certain approaches in order to be supportable by one engine, the transformation is not often easy, thus the problematic processes, which are not easy prespecified are simply maintained by human outside the system or the complicated process is handled as the one complex task which is often known only the person that is performing it. If there is something out of the system then it is difficult to control it. The problem is 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 support several approaches, because each modelling approach is more handy for different process type. I believe there should also alignment tactical level with operational level of management. The alignment is often only in design time or in business process reengineering which prevent fast adaptation. The alignment has been show in [4], however the redesign has not been dynamic.

I believe that top level orchestration should be management by goals and current environment which influence goals as well, the specific procedure how to perform some very specific goal is not important from the strategical point of view. The strategic approach has more declarative then imperative attitude. For example, I would expect the some consideration.

I know have to make 5000 products, should I take it? The goal 5000 has 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 advance. I will have to be fast enough in order to make our customers happy. I will to make our services more complex in order to satisfy our customers, which part of the process I should outsource and which part of the process should stay in house all time, or which part of the process should be changed according to current context of the situation.

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

2.4 Business Process Modelling

There was a a presented basic model structures which are mostly based on Petri Nets formalism, the section covers mostly imperative way of modelling, however, the basic construct are used widely in all modelling languages, even in declarative approach it is possible to model the basic patterns. The list of existing patterns is summarized in Workflow Controls Patterns, which also map every pattern to Petri Net. However, there are more models related to each approach. For example, S-BPM approach is based on Calculus of Communicating Systems (CCS Calculus), see [14] for more detail. Declare is based on Linear Temporal Logic(LTL) [18], etc. The declarative approach and S-BPM is still not widely used in spite of the use cases where these more flexible approaches are much easier to use for modelling.

2.5 Languages Used in Process Modelling

In this chapter, the most used language has been briefly presented. The BPMN is the most used process modelling language standard for many users. However, I see it quite complicated because of many symbols. The fact is supported even by vendors, including Bizagi, which implements only part of the standards, see documentation [1]. It is not easy use for more complex processes. I would prefer EPC style of modelling where role is associated with the task. The pool mechanism is quite complicated to use for more structured processes. The most software for modelling in imperative way has support at least for import of BPMN. XPDL is compatible with BPMN for storing BPMN diagram.

EPC is based on processing events. I prefer this way of modelling comparing with BPMN. It has much less symbols to remember, there is no pools.

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

The Declare language as the concept seems to be very interesting, however it is not so far supported by important vendor. However, the notation is not intuitive comparing to BPMN.

2.6 Analysis and Optimizing of Business Processes

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

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

Bottleneck analysis is very useful if there are available the data from historical performance which we can use. Because, if we chance process orchestration, we need to evaluate if there is no significant bottleneck.

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

I see the problem especially in gathering data for analysis and also most of the analysis are highly manually dependent which means they can not be done frequently. That implies the late recognition of the problem. I believe the core of the problem lies, that data for analysis is not collecting all the time and there are not sortable according to context which is very important. If the context of data is unknown, the data can be transfered to useful information with a lot of difficulties. In case study related to manufacture company, we had a lot of problem with quality and description of the data. We had to know context of environment when data was collected, otherwise the results would misleading. The data about part of the context was store only in memories of people (e.g. changing machine).

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

2.7 Conclusion

As it has been presented, there is several modelling approaches for modelling business process which are based on different ideas, especially declarative compared to imperative approach. Each approach is more suitable for usage in different use case, e.g. imperative way of modelling is better option in case it is desired to have precise control over all process and it 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 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). In latter case, it is better option the declarative way of modelling. 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 the way how to make orchestration of different modelling approaches. Other problem is flexibility in runtime, currently solved by manual intervention in process instance (cased-based system) which is quite time consuming, especially when there is a significant set of situation where the handling cases are very similar at higher level of abstraction, e.g. typical question which part of process should be outsourced? There are often several ways how to reach the goal, for example in many case, it is possible to make decision whether to buy or make product, however decision is often based on our previous experience and current context of the company.

I believe that, the ideal business process engine should be able to support several modelling approaches, it should support all level of abstraction of modelling (strategical, tactical, operational), it should be primary driven by goal (tactical level) in order to keep the right process alignment, however it should choose the best possible option how to reach the goal considering current context. The system should be able to learn from the historical interaction with human and system. The system should high-level of flexibility in order to quickly adapt to context changes and reconfiguration of tactical level should immediately impact operational level. Everything should cooperative as much automatically as possible.

3 Dynamic Objects

One of the largest disadvantages of using classic business process management system is the lack of flexibility of management of processes. However, the classic approach is so far widely used in many systems and the main benefit of this approach is the full control over the process execution. Therefore it is important enabling flexibility in process-aware information system. The performance of the process is influenced by logic structure (rules in process), resources, human and many other aspects that can be specific for each instance. The logic structure in process does not change so often as other context-depended variables that has significant influence to process performance, especially if there is a high variability in time of process performance. Adding context to managing of the process can significantly improve prediction of the performance of process instance. There is also possibility to add new context-depended rules to process definition which enable more precise managing of process instances and enabling functionality to differentiate behaviour according to specific product or assigned recourses.

There are many events which can significantly influence the result of a process. Unexpected occurrences are mostly caused by changes in the environment of the process (which can be represented as the context state in time of execution). 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 a process. 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 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, success rate, cycle period, quality and added business value. 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. Another important issue aspect of performing a process is managing the resources. The human factor has a large impact to overall results of processes. Every business process management system user is different thus and that is the reason why the behaviour of the workflow management system should be adapted its behaviour to for use by each individual person. The adaptation of the workflow management system to suit individual users can positively influence on the performance of the workflow management system participant, which will be described later.

The presented solution uses as the workflow management system to collect 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. The workflow management system thus is thus used works as a measurement tool.

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

- Human acting in process instance,
- resource used in process instance,
- product flowing within process instance,
- process component also known as proclets,
- task in process instance
- objects representing relationship between objects mentioned above (e.g. relation human and task)

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

There are also context variables which are not directly related to the process instance and have influence to the context, for example performance of specific weekday. However, it is possible to analyse them as well and understanding their influence can be used for improving managing process structure and manageable context attributes mentioned above.

The importance of context related to user was presented in paper Modification of Workflow Management System Focused on a Human. The importance of analysing of product attribute as the set of context variables was shown in paper Process Mining in a Manufacturing Company for Predictions and Planning and Analysing Resource Performance and its Application in Company. Based on the analysis of this type of context there can be improve the overall predictability of the system.

Due to the workflow management system, it is possible to manage the dynamic profile of each object with minimum effort due to 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 especially for better man-aging of resources within the workflow management system. It is also very useful in for the assignment procedure.

The presented algorithm based on the dynamic process redesigning uses the dynamic profiles of processes to choose the optimal way to reach the goal of a process. 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 what income products are needed to produce the desired product.

The presented method approach 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. It is necessary to define the final product and a set of conditions which are related to the context of the process. By using databases of processes which 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. This means a great advantage because information about the context of a process is often not available before its runtime.

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

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

The methods are currently being tested on data provided by a manufacturing company.

4 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 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 situation when assignment 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 evaluation. 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 procedure which will lead to the desirable results (output). Each process should lead at least to one desired output, some process may lead to such similar output that it can be possible use one as the substitute for other process. Decision which substitute is the best is very often depended on context, e.g. backup plan is executed when certain circumstances reveal. For example, we choose cheaper solution in case 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 need for evaluation system for each substitutable component in order to choose the most suitable substitution for context-specific request. Evaluation should be done in moment of planned execution in order to get best possible data for decision which substitute of process is the most suitable in specific moment.

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

Thus, there 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 thus can work as an input data source for the component creating profile of behaviour of each important existing object in the system.

4.1 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 [21]. Another presented solution is the combination rule-based, case-based, and agent-based approach [22]. The presented approach generates a structured process definition which can adapt to process context. Furthermore, the internal context variables are updated in runtime due to existing workflow management system. Our approach is a hybrid between 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 data-driven [8] and control flow approach. It merges subprocess definitions using interfaces previously defined based on product compatibility.

4.1.1 Description of Purposed Context-Aware Management System

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. 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 be also result of analysis of value stream management), which track added value in 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 allows switching over to another way of producing the product according to changing conditions influencing context of the process. This can be done due to the declarative (query) approach. Within this approach, it is necessary to define the final products and a set of conditions, which are related to the context of the process. By using databases of processes that include process definitions and related 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 runtime (the context is constantly changing).

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

The approach is a hybrid of rule-based and goal-based approach. It deals with contexts using a top-down declarative (query) approach using context variables. The process definition uses structured definition of subprocesses and combines them according to the current context. The basic idea can be compared to a real-time decisions 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.

4.1.2 Example Illustrating a Context-Aware Approach

- Assume the desired final product is lunch. There are several substitutable ways how to get lunch, e.g.:
- going to a restaurant,
- making lunch at home,
- or hiring a cook, etc.

Each of these options (like each product) desired to perform a certain subprocess. For example, to have 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, lunch in a restaurant and lunch at home, including the subprocesses connected with both possibilities of getting lunch, can be substitutable.

The suitability of substitution related to process depends on the context of the process of creating the final product (lunch) because the cheapest or the fastest solution is not always the best 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, the algorithm chooses the second best option. The behaviour can be compared to computer networks routing algorithms, especially OSPF (Open Shortest Path First) algorithm. If only structure 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 delivery to home), it is necessary to redefine all related processes which use the subprocess and make evaluation of processes using the changed subprocess. The process can have different priority, thus it is not possible to simply change the subprocess in each related process.

In case management, it is possible to adapt a process according to 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 case-based decisions.

The presented approach 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(query) skill for context-based online adaptation.

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

The declarative approach to process definition is quite similar to real-life planning. It is necessary to set the goal of a project first and then to find a how to reach the goal. Thus, the final goal generates other goals. Each goal can be fulfill by the performing process.

There are three basic levels of management: strategy, tactical, and operational level. At beginning First, the main goals have to be set. The primary goals should be supported by The other goals should support the primary goals. Thus, the final goal generates other goals. 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 production. The production process of the final product is decomposed into several subprocesses with ordering material being the basic subprocess. The approach is usually used for the calculation of costs. Different strategies can be chosen to create a product – all the processes needed to produce the product can be performed within a company or certain processes or their parts can be outsourced. There are several strategies to make a product, at least it is often possible to produce it or outsource it. A combination of both of the approaches is usually used.

The declarative approach is useful for online adaptation because a process can be defined as a set of conditions related to a changing environment and the final structure of the process is created according to the conditions in a particular context. The new process is feasible as its subprocesses generating interchangeable products can be modified.

It is also possible to use the declarative approach even for making constraints for on the properties of resources. As a result, a process structure can be obtained that 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 in respect to a successful finish accomplishment of the desired task, but in most cases, they are not usually the cheapest at the same time.

The common declarative approach in process modelling is described in [5][16] [17]. However, the top-down declarative (query-based) approach is focused more on hierarchy of goals. The approach is more similar to database queries. The generated process can be launched in a rule-based workflow management system. The query approach is useful as an analytic tool.

4.3 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 something is done. A process can contain several subprocesses. The process often needs some other products to run. There are several ways to obtain a certain kind of a product because in reality, there is a possibility of obtaining a substitute for the desired product. 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 a process. 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 favourable in that no concrete product has 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 which of the options is the 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 some 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 equipment for making tables. Otherwise, buying all equipment and material for making one single table would cost more than a joiner would be paid to make a 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 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.

4.3.1 Examples of the Properties

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

These properties are changing according to actual performing of the process. The time to accomplish the process can be changed as a result of redesign of the process or changing resources. The cost of a product is calculated as the sum of fixed and variable costs while the 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.

4.4 Optimizing Using Dependency Tree

The aim of this task is to generate a complete process structure by specifying of 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 the feasible solutions. When a query about the final product is made, the tool tries to find the optimal solution to the process definition according to the launched query.

4.5 Main Advantages

- It is easy to use. The tool only needs to have a set of desired products and constraints.
- As the process definition can be built 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 limited amount of time).

The problem can be defined as how to create the optimal process structure while keeping the process logic (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 automatic product creation? 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 automatically to avoid destroying of 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 a result of some agreement process, too. In order to make any queries, it is necessary to have a database containing sufficient information.

The approach of following the product value chain is typically used in value based management.

4.6 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 factors 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 how to make 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 potential evolution of the system, for example an increasing amount of products dependent on an increasing demand.

4.6.1 Content of the Database of Processes

- The process definition, which was designed for producing the desired product.
- A set of places in the process definition 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.

- Runtime details such as the performance time, costs (calculated from fixed and variable costs depending on time), success rate, etc.

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

It is necessary to specify a set of places, which enable the process to be connected 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 finishing of the process. For 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 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 such kind of information. It will be described in more detail in the next section of the text.

4.7 Managing the Dynamic Properties of the Activities

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 to obtain the up-to-date profiles (sets of properties) of processes.

4.7.1 Static and Dynamic Properties of the Process Profile

In the case of the 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 the 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 the dynamic properties, it is very difficult to keep the correct values up-to-date. Examples of the dynamic properties are time to accomplish the process, process costs, or reliability of the process. To manage the dynamic properties, updating of the specific values is triggered by events, which are created by the workflow management system. The events are happening 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. 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 holds a collection of items related to a particular process. Every item holds a set of monitoring indicators, time to accomplish the process, process costs, reliability of the process, etc. for each process.

As can be understood, the dynamic properties describe performing of a particular process. They can reflect changes in the process performance. A combination of the static and dynamic properties can provide the calculation of the current process costs. 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 process of dealing with the dynamic properties is similar to [11]. Dynamic properties can be used in the decision process see [12]. 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. 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 ago.

4.8 Product Dependency Tree

Each activity has some inputs that are necessary for producing some products that can be used as input for other processes.

4.8.1 Phases of the Optimizing Process Structure

- It is necessary to build and keep an up-to-date database of processes.
- The dependency tree based on the database of processes has to be built.
- When the previous steps are accomplished, it is possible to find the optimal solution respecting the selected constraints.
- The optimal solution is then to find all processes and their subprocesses within the dependency tree that are needed to create the final product. These are then merged into one final process that brings the same results as the original process.

The main idea of the algorithm is to connect the 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.

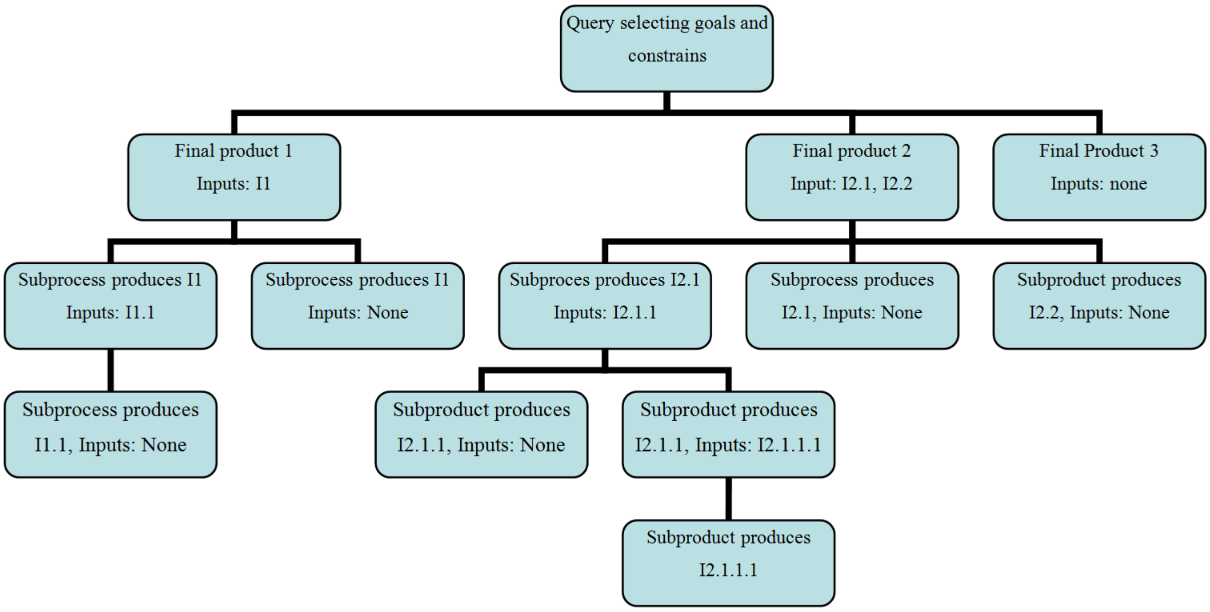

Figure 1: Example of construction of Product Dependency Tree

4.8.2 Types of the Processes

- 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.
- Regular process – the process that has sub processes but it is not the root process at the same time.

4.8.3 Steps of the Algorithm

- Select the desired products.
- Find all processes that can be used to produce the products selected.
- You will get a set of processes as the result. Search for all activities 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).
- Repeat until the activity is not a leaf activity.

4.9 Searching the Process Definition in Product Dependency Tree

The dependency tree covers all possible ways of getting the desired product. Each node of the tree represents one process. The relationship between nodes represents the relationship between the input and output products of processes. The nodes in the tree are synchronization points and contain additional data about the historical process performances. The algorithm follows the route from the root to leaf process. The route is marked according to the additional properties of the process. For example, there are two routes representing two options. One option is more cost-effective but slower, another is faster but more expensive. If the slow way meets the constraints, both the routes will be followed to lower levels until the leaf processes are reached. The higher level of the tree also covers related sub processes so that there are already aggregated values in the higher level of the tree due to the workflow management system. The aggregated values can be used in the decision process. If several solutions are available fitting the query, the solution has to be chosen that has 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.

4.9.1 Example of a Query

The query can look like as follows – select a process in which the desired product is a 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. To accomplish this aim, these steps are 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.

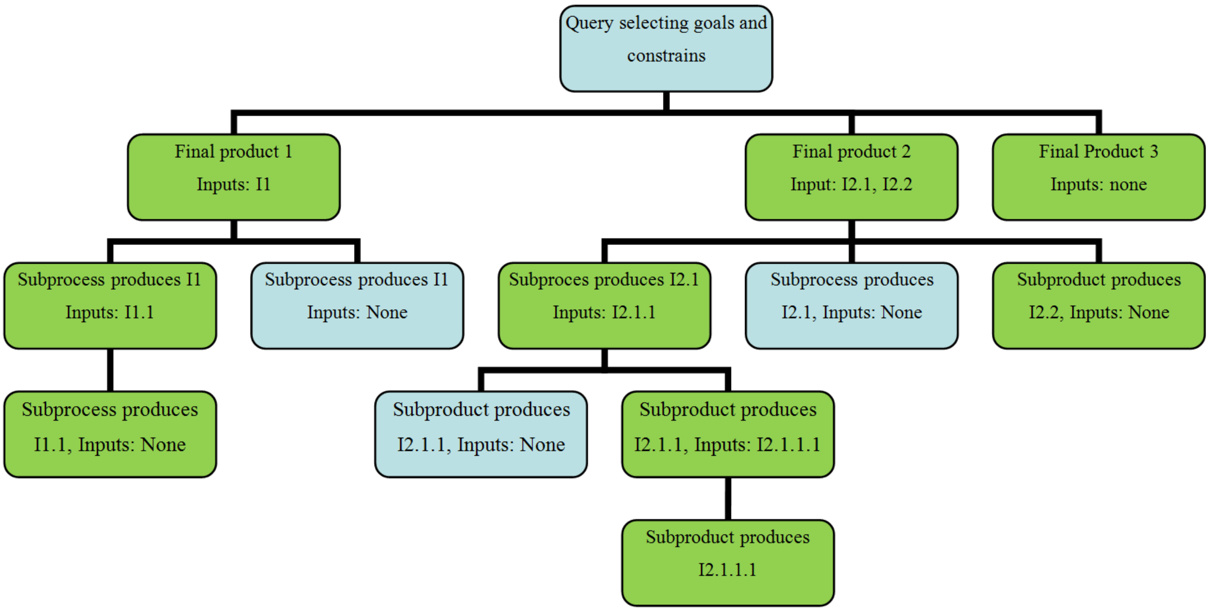

Figure 2: Example of searching in Product Dependency Tree

- Launch the modified process by the workflow management engine.

The query is similar to regular database queries, thus a SQL-like language illustrate how it looks like.

```
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.time <= 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 a ready for 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 priority of a process then it should be done simply by changing the query. If the variant of making substitutable product is chosen the algorithm has to re-evaluate the process just before launching. The final process will be run-time generated according to the context values. Even if the number of variants of making 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 current situation.

4.10 Experiments

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 describing 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 the individual projects were similar and the following solutions were similar too.

The outcome of the research was 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 is to outsource a 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 there-fore 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 for dynamic selecting 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 haves a different productivity in making some operations. The work for a shift should be planned with respect to awareness of the particular individual person 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 an RSS reader that monitors the amount of submitting work done by the person by RSS reader. The method is more precise than a calculation based on standards. Another important side effect function of the method is is possible to monitoring abnormalities or trends in the behaviour of the particular person.

One of 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 huge dispersion of time for similar operation. The 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 more several steps depending on the input product to for the machine. Due to a manufacturing execution system, data about the operations tasks will be continuously processed thus so that the model of the process of model will be still kept up-to date with all the runtime characteristics. The dynamic profiles of tasks and processes also enable make it possible to possibility to report abnormalities immediately during production. The detection of a possible failure, before it happens occurs, saves a lot of money.

4.11 Conclusion

Process management should consider the context of a process. This article 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 is also adapted for the particular workflow management participant according to his 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 constant 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 work-flow management system.

The context-based approach is very helpful for dynamic rescheduling. The generated process is practicable 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 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 changes in the process definition were briefly illustrated via Hi-level Petri nets which support merging different nets into one process definition. The dynamic process structure allows higher flexibility of the process management in runtime. The online adaptation is important especially for processes which 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 the unique behaviour in manufacturing processes as was briefly described in [20].

The dynamic profiles of workflow management participants can be used for an extension of assignment tasks to 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. However, a description of all possibilities of the dynamic profiles of all objects is behind the scope of the paper. The aim of the article is to show how a context-aware system can look like and what its advantages are.

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

The goal of the section is to provide the description of a system capable of automatic adaptation of managing logic. The logic is continuously adapting according to the user-driven and environmental context. The system observes itself and based on these observations, it changes realizations of implemented business processes in order to keep optimal and fault-tolerant performance of the system. The concept is presented on service-oriented architecture.

5.1 Introduction

The context-adaptability is an important and desired feature especially for processes situated in dynamic environments [22]. The adaptability to the environmental changes allows 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) [9]. 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 easy runtime modifications of a composed system by changing its particular components [7].

However, realisation of the context-adaptive processes in SOA brings several issues that need to be addressed. The rescheduling of a business process require strong knowledge of its context at both business and implementation level, 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 paper is to provide a conceptual approach to description and realisation of context-adaptable business processes by service-oriented architecture. In the paper, we propose a model for both description of the adaptable business processes as well as description of their potential realisations and implementations including performance metrics. We also propose a method of automatic evaluation and assignment of process definitions to their best process realisations based on observation of the actual context and historical and current values of the performance indicators.

The rest of this paper is organised as follows. Section **??** introduces the context adaptability in more detail including related state-of-the-art work and an outline of our approach. The approach is described in-depth in Section 5.3 by conceptual models of context-adaptable business processes and the context adaptability itself. In Section 5.4, we describe a method to evaluate and assign process realisations by means of a product dependency tree. Section 5.5 continues with the description of implementation of context-adaptable business processes as SOA. Finally, Section 5.6 describes evaluation and discussion of our approach and Section 5.7 concludes the paper with an outline of future work.

5.2 Outline of the Proposed Approach

In our approach, we propose the way to describe a business process 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, we propose *a hybrid modelling approach* to model business processes in both declarative and imperative way. The approach allows 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 imperative way (e.g. by work-flow or orchestrations). Our approach is also *a combination of rule-based and case-based process management systems*. The approach allows 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 *goal-based approach* when each realisation has to fulfill goals required by the business process.

The next sections describe business process modelling and management in 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 looked 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 work-flow or an orchestration of IT services in the case of IT analysts or IT architects.

The basic idea can be represented by an 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 processes 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 and should be described at designtime while context information is provided at runtime by an existing work-flow management system.

Finally, the process management system in our 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). Our approach allows the 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.

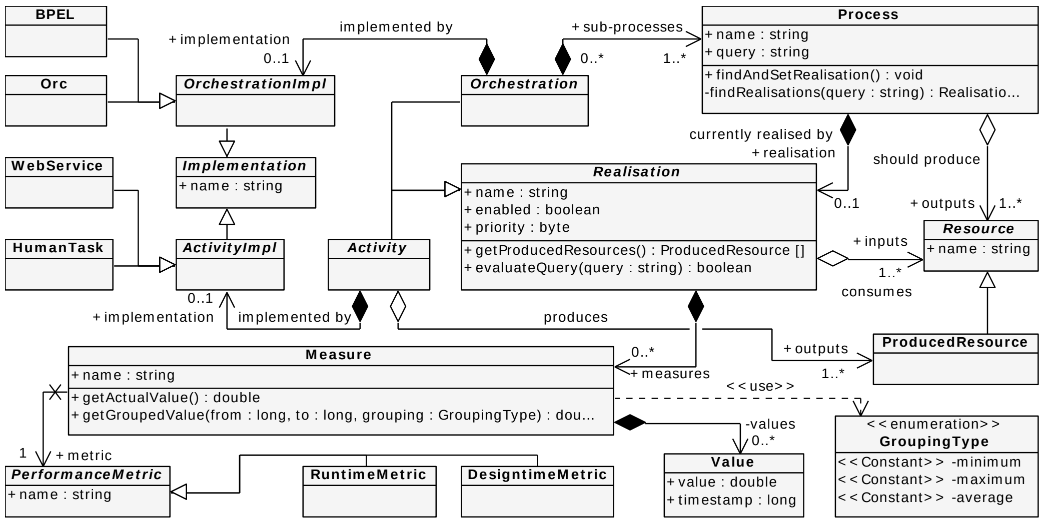

Figure 3: The conceptual model of the process definition, realisation, implementation, and measures, for context-adapable business process modelling.

5.3 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 (e.g. current values of performance metrics). This section serves as an introduction and a conceptual base for Sections 5.4 and 5.5 dealing with the context-adaptability and the implementation, respectively, in more detail.

5.3.1 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 3 and it 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 assigned one process realisation.

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

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

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

5.3.2 Modelling of Context Adaptability

In the conceptual model from the previous section and Figure 3, the context adaptability is represented by 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 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 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 proceed as follows. At first, all convenient process realisations are found by 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.

5.4 Process Realisations and a Product Dependency Tree

In the case of more variants how to achieve a business process's goals by its various process realisations, which is typical especially for project-oriented processes, one of the variants has to be selected according to desired conditions. However, it is often 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 possibility of mapping of 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 delay in the process performance.

In this section, we describe 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 5.2 and 5.3.2) and on process realisations defined by business process engineers. The best realisation is selected by its produced resources (the desired products) which meet goals of a given business process and by preferences stated in the query.

5.4.1 A Product Dependency Tree

The *product dependency tree* is defined as an 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 can not be decomposed, contrary to orchestrations).

The root node of the tree contains an empty set of the activities, $n(a) = \emptyset$, and an elementary set with the abstraction of a main business process only, $|n(p)| = 1$, i.e. the process, which is producing output resources consumed directly by its user, not by another 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 the 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 and informally, 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 a 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".

5.4.2 Building the Product Dependency Tree Based on a Query

The query declares the final products of a 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 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 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).

To build the dependency tree, each business process participating in hierarchy of decomposition of a 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, at least information that the process is finished). Moreover, each process realisation which may participate in the 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 process providing inputs as their products. Finally, each process realisation has assigned its performance profile with metrics (e.g. a 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 a 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 a main business process. The tree $G = (N, E)$ is build from a query by 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 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 its all sub-processes are added into set $n_c(p)$. Moreover, each process realisation which is activity is added into set $n_c(a)$.
3. The second step is repeated until each leaf node n in the tree has $n(p) = \emptyset$.

5.4.3 Evaluating the Query

To find the best realisation of a main business process through its whole hierarchy of process decomposition, we need to evaluate the previously build product dependency tree and the query together. In the tree, each branch, i.e. a 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 their value and a 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 user-defined priority of the realisations or randomly.

The above mentioned evaluation criteria are related mostly to the process performance and to 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 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 neither all branches nor every single process realisation in a branch. We need to re-evaluate only such 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 another nodes related to the change. Therefore, continuous re-evaluation of a query, which is necessary for business process context adaptability, may be considerably optimised.

5.4.4 An Example of the Query

A general business process usually contains a human task and Web services. A user who needs a 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, a cost, and a quality of the process product.

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

To demonstrate our 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

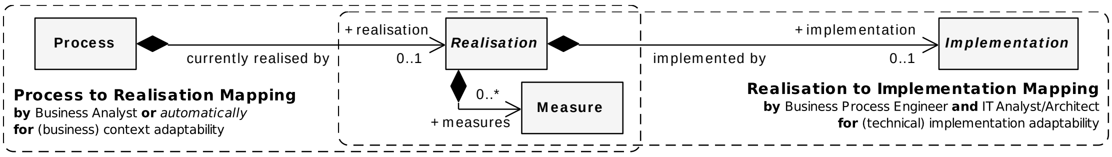

Figure 4: Context adaptability and implementation adaptability in the conceptual model for context-adaptable business process modelling.

implemented in several different ways, by different distributed algorithms orchestrating many services of different providers. For example, a simulation can be performed by a group of commodity low-cost computers, however, the 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 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 second) AND (p.cost <= 2 Euro)
/* order the results according to costs and return the first */
ORDER BY p.cost LIMIT 1;
```

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 dependency tree, as it is described in Section 5.4.2.
- Select a branch of the tree such that it satisfies the constraints of the response time shorter than 1 second and maximum cost is lower or equal than 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 order above.
- 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 it is described in Section 5.4.3.
- 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 a desired product). However, the most important are response time, cost, location, and quality of service.

5.5 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, we will take into account implementation details of business process realisations and discuss mapping of context-adaptive business processes to services of Service-Oriented Architecture.

5.5.1 Implementation of Context-Adaptive Business Processes

The context-adaptive business processes are implemented by *process realisations* which "realise" *process definitions* as it is described in Section 5.3.1. In the other words, each process realisation can have assigned a process implementation. 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 a 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 the adaptability, namely *(business) context adaptability* and *(technical) implementation adaptability*, as it is depicted in Figure 4 (for the context see also Figure 3). 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 our approach, we focus on the context adaptability to allow optimisation

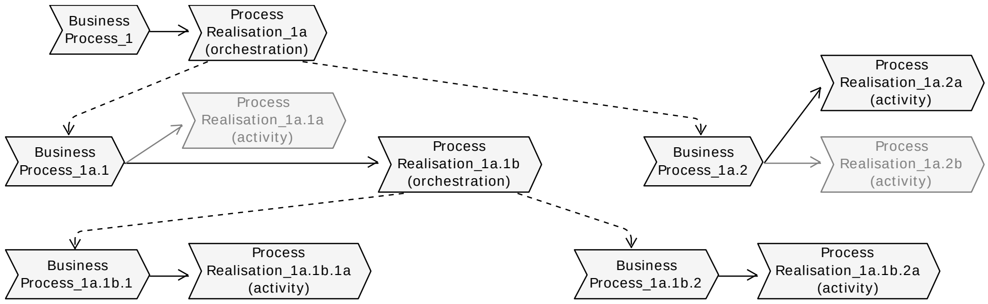

Figure 5: Hierarchy of business processes with mappings to process realisations (denoted by rightwards full arrows) and decomposition to orchestrations of sub-processes (downwards dashed arrows). Alternative mappings to realisations 1a.1a and 1a.2b are shaded in grey colour.

at the business level by automatic mapping of process definitions to process realisations. Moreover, at the business level, each realisation represents also its implementation which affects the 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 the most one implementation. In this way, the mapping of process realisations to their implementations need not to be considered.

The implementation of context-adaptive business processes proceed 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 (also see Section 5.3.2). The business process realisations can be found by decomposition of a main business process (a top-down approach) or by 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 these 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 to deployment, testing, and 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).

At each time, the resulting hierarchy of mappings of business processes to their realisations (for example, see Figure 5) 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 a process realisation is determined not only by 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.

5.5.2 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 maintenance or improvement of process performance. In our approach, we utilise *runtime metrics* and *designtime metrics*. The first are used to measure performance of process realisations at runtime and to evaluate their suitability as final realisations of defined processes, while the second allows to include into the evaluation also hidden costs, such as resource utilisation or set-up costs (i.e. changeover costs associated with switching from 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 designtime. Together with definitions of business processes, a business analyst define *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 an "automatic order processing" realisations because a manual order processing cannot meet 60 minutes 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 [6]: availability and reliability of the service, price, throughput, response time, latency, performance, security, accessibility, regulatory, robustness/flexibility, accuracy, servability, integrity, and reputation.

5.6 Evaluation and Discussion

Potential applications providing fault tolerant and adaptive solution in 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, a resulting system has to implement both process realisations, different for each user group according to their needs, even if both realisations provide the same products. Our approach provides the solution which can automatically adapt to a changing environment based on user requirements.

The concept of adaptation at business process level has been previously analysed in [13] as rescheduling of projects with 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. Our approach extends the concept of dynamic profiles from [13] to dynamic process realisations. For example, in 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 performance of these services and other user-defined metrics representing a context of the process allows to change possible process realisations, i.e. to switch service orchestrations and implementations. A resulting system would be very flexible (adaptation to the context, it easy to add new services at runtime, etc.) and would be able immediately find how to keep optimal performance based on a user-defined query which describes priorities for desired optimal behaviour.

5.7 Conclusion

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

The current approach can be improved. The future work includes an extension of the approach with the ability to analyse combinations of service performance properties and process input resources, because we expect that a service may have 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 of desired outputs, i.e. without any knowledge of the actual services inputs).

6 Conclution

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 Section **??**.

The architecture designed in thesis has several advantages compared to current business process management systems architecture. The architecture provides automatic fault-tolerant mechanism, which also support several modelling approaches. The architecture has natural support for alignment with strategic level of management, where there is possible to create to several possible strategies simply declaring the objectives and constrains related to them and the algorithm create the Product Dependency Tree. The architecture support context-awareness, the process logic can be re-evaluated in time of binding each subprocesses. The designed architecture support to usage historic data and use them for the heuristic analysis.

The architecture can be used as adaptive context-aware planner. This can be helpful especially in short time predictions where there is several context variant of performance.

6.1 Supporting several modelling approaches

The most used modelling approach so far is 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 kind of processes especially depending on level of looseness of managing. For example, there is very difficult to model process describing the treatment of illness, because there is so huge state explosion of combination of all activities, that using traditional imperative approach would be almost impossible. Because there is no simple answer 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 enables simple join one modelling approach on others into shared business process management architecture.

6.2 The fault-tolerant mechanism

In many situations, there is a need for backup plan especially if risk issue occurs. Thus, it is necessary to watch continuously the changing environment of the process. Currently, there is a situation solved by using exception handling, however these exception has to be implement in process definition which is time consuming. The unified strategy of managing changing logic of the process which 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 dynamic object, there will be a high potential of a good resource usage prediction in short future. It can automatically choose the most suitable subprocesses which are able to fulfill the goal. This could be compared to OSPF algorithm in network routing area.

6.3 Alignment with Strategic Objectives

The alignment with strategic objectives is very important issue from the whole system performance point of view. However, the most methodology used for alignment is often used in design time and do reflect changes of environment thus there is needed some time in order to make strategic changes, because the alignment is not done very frequently. However, the environment is changing even faster than it used to be a few years a goal, thus even the strategic objectives are need to change faster, however when the company change the strategy objectives it has a huge influence to overall organization because, the strategy objectives has direct relation to the processes.

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

The query selects the most suitable variants of process orchestration while the user declare the final products and constrains related to the producing the final products. The user does not need to know all process definition at the strategic level. However, defining the query influences the process orchestration not just in design time. The query can be change in every moment, the running instances of processes can be each in different state or it is possible to add new subprocess to the system while other process instances will be performing.

6.4 Context-Aware Adaptive Scheduler

The architecture is designed to deal with many aspects that other planning systems do not take into account. Therefore it there is a solid base for adding support of short time 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 changing context. Thus it can improve short time prediction of outgoing actions. Each action within system is connected to particular resources. Because, the system is context-driven and uses for heuristic analysis historical performance data, the prediction of outgoing action is much closer to reality comparing to systems that do not use context-aware approach. In papers [19][20], I tested influence of context dependencies no real data set from manufacture covering several millions records and we were able to decrease the deviation of prediction of performance time to approx. 50% when we predict according to the attributes of the product. Thus, there was a quite important context dependency even in production process where every action is supposed to be standardized which should imply the low level of performance time deviation. The future work will be focused especially on improvement in are Scheduling. Related to that topic the location dimension should be also taken in account. The presented approach is quite general, thus it could be transferred to different area with modification of heuristic analysis. It was applied to Service-oriented Architecture.

References

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

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

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

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

[5] 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.

[6] 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.

[7] 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.

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

[9] Fernando Lins, Julio Damasceno, Andre Souza, Bruno Silva, David Aragäo, 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.

[10] 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.

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

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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–, Washington, DC, USA, 2007. IEEE Computer Society.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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