Context-Based Adaptation of Process Definition

Vojtěch Mates
Department of Information Systems
BUT, Faculty of Information Technology
Brno, Czech Republic
e-mail: imates@fit.vutbr.cz

Tomáš Hruška
IT4Innovations Centre of Excellence
BUT, Faculty of Information Technology
Brno, Czech Republic
e-mail: hruska@fit.vutbr.cz

***Abstract*—There are several ways to reach a goal. The right way often depends on the context of the process that produces the desired product. The context can be understood as a set of conditions including changing requirements and resource properties. The method described in this paper is a possible way of dealing with a changing context of a process. Our solution uses a product dependency tree and a database of processes. The dynamic profiles of the process are maintained by a workflow management system that produces a valuable data stream that records past behaviour.**

Keywords – business process management; business process intelligence; workflow management system; business process optimizing

I. INTRODUCTION

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

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

There are more approaches for adaptive managing systems. The challenge is to provide flexibility and offer process support at the same time [19]. Another presented solution is the combination rule-based, case-based, and agent-based approach [20]. Our approach generates a structured process definition which can adapt to process context. Furthermore, the internal context variables are updated in runtime due to an existing workflow management system. Our approach is a hybrid between a declarative approach, rule-based approach, and a goal-based approach with the ability to monitor itself and use the information to optimize the process performance by the user's preferences, as defined by the users query.

The approach is also a combination of data-driven [14][15] and control flow approaches. It merges the subprocess definitions by using interfaces previously defined based on product compatibility. The algorithm uses dynamic profiles which describe the 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 to create the desired product. A route in the product dependency tree is used to build the final structure of a modified process. The product dependency tree represents a product hierarchy. In other words, it reflects which income products are needed to produce the desired product.

Our method has several advantages. It allows switching over to another way of producing the product according to changing conditions. This can be done due to the declarative approach. With this 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 that include process definitions and dynamic characteristics, it is possible to build new processes according to specific conditions. Thus, the process can be modified according to the current context, e.g., costs, time, success rate, resource problems, etc. This is a great advantage because information about the context of a process is often not available before runtime.

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

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-223-3

A. *Example*

The desired final product is lunch. There are several ways to get lunch, e.g., going to a restaurant, making lunch at home, or hiring a cook. Each of these options (like each product) is 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 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 the 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 to deliver to home), it is necessary to redefine all related processes which use the subprocess and make an evaluation of the processes using the changed subprocesser. The process can have a 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 the current context, but each kind of change has to be defined, for example for price changes, each supplier can use a different order process. Overloading due to continuous change can eliminate the advantage of case-based decisions.

This paper aims to show how to optimize the process definition by using data captured by the workflow management system for managing dynamic profiles of processes, and how to use the declarative skill for context-based online adaptation.

The rest of the paper is organized as follows. Section II provides an introduction to the Top-Down Declarative approach, Section III overviews the workflow management system used for creating dynamic profiles. In Section IV, the merging of processes using Petri nets is briefly presented. Section V introduces a query-based approach using the declaration of final products. Section VI covers prerequisites of the optimizing method based on The Product Dependency Tree. Section VII describes the database of dynamic profiles of processes used for optimal decision in The Product Dependency Tree. Section VIII describes the approach to managing dynamic profiles. Section IX introduces The Product Dependency Tree, section X explains searching in The Product Dependency Tree. Section XI overviews experiments related to rescheduling decisions comparing an algorithm to human decision. Section XII summarizes the paper.

II. TOP-DOWN DECLARARITVE 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 way to reach the goal. Thus, the final goal generates other goals. The approach is also frequently used in the field of production. The process of production of the final product is decomposed into several subprocesses with the ordering of 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. The declarative approach is useful for runtime 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.

The common declarative approach is described in [12][13][16][17]. However, our approach is focused more on hierarchy (hierarchy of goals). The approach is 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.

III. ARCHITECTURE OF THE WORKLOW MANAGEMENT SYSTEM

The workflow management system can manage the cooperation between the workflow management system participants (including the interaction between people and software) in order to perform a business process.

The main purpose of the workflow management system lies in managing resources according to the process definition, which is usually set by the company management. Making changes in a business process is easier because the process definition is not hard-coded into the system and the company is therefore more adaptable to changing conditions. The workflow management system also allows easier business process re-engineering, as well as provides valuable audit data (in addition to managing and monitoring of the processes) which can later be used for analysis. Information extracted from the audit data can be also used for better managing as will be illustrated further in the paper. The architecture of the workflow management system will be described very briefly using the reference model created by the Workflow Management Coalition.

The workflow management system should consist of the Process Definition Tool, Workflow management engine, Work List Handler, and User Interface [5]. Creating a process definition is the first part of the process deployment in the workflow management system. This is usually done by the Process Definition Tool. A process described by the WFDL (Workflow Definition Language)

[10] is an output of this tool. A process definition should include information about tasks such as who can perform individual tasks (mostly the role-based approach), and information about routing between the tasks. The business process execution logic can be described as routing that will be dealt with in more detail in the following text.

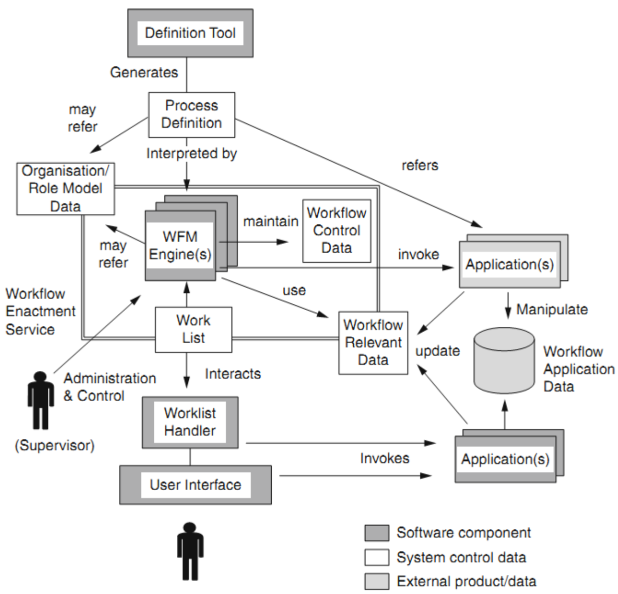

Figure 1. Architecture of the workflow management system [4]

Instances, i.e. particular processes, originate from the process definition. The process definition is transported to the workflow management engine where it can be launched as a process instance. Work lists are created by running particular instances according to the process definitions with specific data. For example, during the execution, applications or people can update a workflow's relevant data. Work list Handlers then assign tasks to resources, e.g., people or software (mostly web services) after launching a process instance. A workflow management system participant is an object in the workflow management system taking part in a process instance (usually people or web service). The user communicates with the system via several user interfaces like email, application interfaces, etc. All information about the performances of particular processes can be stored for monitoring and analysis purposes.

The workflow management system produces data that can be used as input data to dynamic profiles of certain parts of the process definition.

IV. Modelling a Workflow Using Petri Nets

The Petri nets [3] formalism is widely used for modelling the dynamic aspects in systems. Petri nets consist of places, transitions, and arcs. Places represent conditions in a workflow. Transitions represent the performances of tasks. Oriented arcs connect places with transitions. A process performance depending on certain conditions can be simulated by moving tokens from one place to another place within a Petri net. Specific patterns in Petri nets are used to model a business process such as: AND split, AND-join, OR-split, OR-join, and other patterns [11].

The High-level Petri net Formalism also supports colour (passing several attributes in a token), hierarchy, and time (time stamps). Colour extension enables tokens to hold several attributes of a certain type. Hierarchy allows the building of a process from a subprocesses. Time extension can be used for time related conditions [3]. Modelling business processes can be a very complex issue [1][6][9]. The mathematical basis for Petri nets is suitable to demonstrate the algorithm that makes several Petri nets merge into one Petri net which will be the final process definition. The dynamic change of Petri nets [2], the correctness criteria [18] has to be considered in merging Petri nets. Our approach uses the previously mentioned interface-based process merging which eliminates many problems by correcting the merged process.

V. Custom-made Process Based on the Final Products

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

A. *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 a particular purpose
- It is possible to modify the process structure according to temporary constraints which are only known in runtime (e.g., it is not possible to launch a particular task when the process is running, which may be caused by missing resources or a limited amount of time)

The problem can be defined as how to create the optimal process structure while keeping the process logic (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 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.

VI. PREREQUISITES OF THE OPTIMIZING METHOD

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. A requirement of the optimizing method is 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. The product created by the process can also be information, e.g., where to find a shop offering the lowest price. A process usually contains 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 to work with wood, some nails, etc.

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

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

A. *Examples of the Properties*

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

These properties are changing according to the actual performance of the process. The time to accomplish the process can be changed as a result of a 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.

VII. 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.

A. *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.

Copyright (c) IARIA, 2012. ISBN: 978-1-61208-223-3

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 kinds of information. It will be described in more detail in the next section of the text.

VIII. Managing the Dynamic Properties of Activities Using the Workflow Management System

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.

A. 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 dynamic properties, it is very difficult to keep the correct values up-to-date. Examples of dynamic properties are the time to accomplish the process, process costs, or reliability of the process. 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 the performance of a particular process. They can reflect changes in the processes 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 [7]. Dynamic properties can be used in the decision process see [8].

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.

IX. Product Dependency Tree

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

A. 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.

B. 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 another 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

C. 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

X. Searching the Process Definition in the 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.

A. *Example of a query*

The query can look as follows – select a process in which the desired product is *1 bread or 10 rolls*, the time to perform the process is less than 10 days, and the maximum costs are less than 2 euro. Choose the cheapest process. 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
- Launch the modified process by the workflow management engine

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

```
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 the specialization of the process (e.g., data from sensors - temperature, humidity, etc.), but time, costs, location, and the role of person starting the process are general variables. If it is necessary to change the priority of a process then it should be done simply by changing the query. If the variant of making a substitutable product is chosen, 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 a desired product is the same and the query remains the same, the historical performance data is still changing. Each change means checking the process if it is optimal according to the current situation.

XI. Experiments

The currently implemented workflow management system will also support the adaptation of processes for changing conditions and is currently being implemented. Research has been made based on data describing changes in project scheduling. The retrieved data covers only 21 projects, however, many causes of rescheduling in the projects have been proven similar, and the resulting solutions have been similar, too. The outcome of the research is that almost all projects (over 80%) are rescheduled due to missing the deadlines of the sub processes of the projects or due to temporary unavailability of the resources. After detecting these kinds of problems, the solution is often to outsource some parts of the projects. The supplier is usually selected according to their success rate of meeting the deadlines, quality level, experience, and prices depending on the context and the priority of the projects. The integration of the adaptation into the workflow management system is therefore expected to reflect and react to natural changes in processes that occur in practice. The concept of the product dependency tree is currently being tested for the dynamic selection of the most suitable supplier. The concept of the dynamic profile is being tested for scheduling purposes and the data used for the testing include over one million records. The automatic selection behaves correctly (compatible with manager decision) in almost all projects (over 90%). In some projects (less than 10%), the decisions made by the algorithm were even better than decisions made by managers. The main reason was identified as the lack of time to collect and evaluate all information for decision. In some cases, the algorithm chose the solution that was not optimal because of a changing variable that was not taken into account. In order to fix the problem, the general process has to be confirmed by the manager. However, the tool enabled fast analysis based on user queries, using historical performance data.

XII. Conclusion and Future Work

The context of a process should be taken into consideration in process management. This article aims to describe the idea of the runtime context-based process definition adaptation based on replacing the substitutable parts of the process with its other parts. This paper shows how the original workflow management system is used for maintaining the dynamic process profiles which are used for making decisions in the product dependency tree. The extension can solve the problem of the automatic redesigning process according to constant changes in the context, which enables using advantages in rule-based and case-based approach. The context-based approach is very helpful in dynamic rescheduling in case of changing user priorities. For example, time is suddenly preferred more than costs or due to a problem with resources; the most optimal variant of the process cannot be performed. The solution also supports fault tolerant processes management and includes risk management into the common process management. The solution is suitable for business process reengineering because it can simulate and compare the old and new versions of a process. The changes in the process definition are briefly illustrated via high-level Petri nets that have good support for varying regions in the process definition. The dynamic process structure allows higher flexibility of the process management during runtime. An advantage of this approach is performing a process, which is accomplishable, and it is able to adapt to changing context. It can also manage flexibility at the same time. The case-based management system has the advantage of flexibility. The rule-based management system has the advantage of standardization of a process. The contribution at present is complex approach which provides the ability to make a dynamic change which is driven on by user queries and historical process performances.

Acknowledgment

This research was supported by the grants of MPO Czech Republic MPO TIP FR-TI3 039, by the grant FIT-S-10-2, by the research plan no. MSM0021630528 and by the IT4Innovations Centre of Excellence project CZ.1.05/1.1.00/02.0070.

References

[1] W. M. P. van der Aalst and K. van Hee, "Workflow management Models, Methods, and Systems", The MIT Press. Massachusetts 2004, ISBN 0-262-72046-9.

[2] C. Ellis, K. Keddara, and G. Rozenberg, "Change Within Workflow System", http://portal.acm.org/citation.cfm?id=224021, retrieved: April, 2012.

[3] C. Girault and R. Valk: Petri "Nets for System Enginnering", Springer; 1 edition 2002, ISBN-13: 978-354041217.

[4] D. Hollingsworth, "Workflow Management Coalition The Workflow Reference Model", UnitedKingdom 1995, http://www.wfmc.org, retrieved: March, 2012.

[5] D. Hollingsworth, "WfMC The Workflow Reference Model 10 Years On", United Kingdom2003, http://www.wfnc.org, retrieved: March, 2012.

[6] D. Marlon, W.M.P van der Aalst, and A. Hoftstede: "Process-aware information systems", JOHN WILEY & SONS, INC, New Jersey 2005, ISBN: 978-0-471-66306-5.

[7] V. Mates, "Using Workflow Management System pro Analysis Based on Properties of Resources", DATAKON 2010 Proceedings (Ed. Petr Šaloun), Mikulov, CZ, Ostravsk univerzita, Ostrava, 2010, pp. 161-168,

[8] V. Mates, "Modification of Workflow Management System Focused on Human", Information Systems Development, , New York, US, Springer, 2011, ISBN 978-1-4419-9645-9, pp. 479-490.

[9] A. Sharp and P. McDermott: "Workflow Modeling: Tools for Process Improvement and Application Development", Second Edition, ARTECH HOUSE, INC., MA 02062 USA 2009, ISBN-13: 978-1596931923.

[10] Workflow Management Coalition, "Workflow Management Coalition Terminology & Glossary", United Kingdom 1999, http://www.wfmc.org, retrieved: March, 2012.

[11] Workflow Control-Flow Patterns : "A Revised View, BPM Center Report BPM-06-22", http://www.workflowpatterns.com/documentation/documents/BPM-06-22.pdf, retrieved: April, 2012.

[12] I. Barba, B. Weber, and C. Valle, "Supporting the Optimized Execution of Business Processes through Recommendations", Business Process Management Workshops, Bd. 99, F. Daniel, K. Barkaoui, S. Dustdar, W. M. P. van der Aalst, J. Mylopoulos, M. Rosemann, M. J. Shaw, and C. Szyperski, Hrsg. Springer Berlin Heidelberg, 2012, pp. 135–140.

[13] R. Hull, F. Llirbat, E. Siman, J. Su, G. Dong, B. Kumar, and G. Zhou, "Declarative workflows that support easy modification and dynamic browsing", Int'l Joint Conference on Work activities coordination and collaboration, New York, NY, USA, 1999, pp. 69–78.

[14] V. Künzle and M. Reichert, "Towards Object-aware Process Management Systems: Issues, Challenges, Benefits", Proc. 10th Int'l Workshop on Business Process Modeling, Development, and Support (BPMDS'09), Amsterdam, The Netherlands, 2009.

[15] V. Künzle and M. Reichert, "PHILharmonicFlows: towards a framework for object-aware process management", Journal of Software Maintenance and Evolution: Research and Practice, Bd. 23, Nr. 4, 2011, pp. 205–244.

[16] M. Pesic, H. Schonenberg, N. Sidorova, and W. M. P. van der Aalst, "Constraint-Based Workflow Models: Change Made Easy.", in OTM Conferences (1), 2007, pp. 77–94.

[17] M. Pesic, and W. M. P. van der Aalst, "A Declarative Approach for Flexible Business Processes Management", in Business Process Management Workshops, Bd. 4103, Springer Berlin / Heidelberg, 2006, pp. 169–180.

[18] S. Rinderle, M. Reichert, and P. Dadam, "Correctness Criteria for Dynamic Changes in Workflow Systems – A Survey.", Data & Knowl. Engineering, Bd. 50, Nr. 1, 2004, pp. 9–34.

[19] 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, 2008, pp. 16–30.

[20] B. Weber, W. Werner, and R. Breu, "CCBR–Enabled Adaptive Workflow Management.", in Proc. European Conf. on Case-Based Reasoning (ECCBR'04), Madrid, 2004, pp. 434–448