Rule- and context-based dynamic business process modelling and simulation

Olegas Vasilecas, Diana Kalibatiene and Dejan Lavbič

Abstract

The traditional approach used to implement a business process (BP) in today's information systems (IS) no longer covers the actual needs of the dynamically changing business. Therefore, a necessity for a new approach of dynamic business process (DBP) modelling and simulation has arisen. To date, existing approaches to DBP modelling and simulation have been incomplete, i.e. they lack theory or a case study or both. Furthermore, there is no commonly accepted definition of BDP. Current BP modelling tools are suitable almost solely for the modelling and simulation of a static BP that strictly prescribes which activities, and in which sequence, to execute. Usually, a DBP is not defined strictly at the beginning of its execution, and it changes under new conditions at runtime. In our paper, we propose six requirements of DBP and an approach for rule- and context-based DBP modelling and simulation. The approach is based on changing BP rules, BP actions and their sequences at process instance runtime, according to the new business system context. Based on the proposed approach, a reference architecture and prototype of a DBP simulation tool were developed. Modelling and simulation were carried out using this prototype, and the case study shows correspondence to the needs of dynamically changing business, as well as possibilities for modelling and simulating DBP.

Keywords

Dynamic business process, Business rules, Context, Simulation, Business process modelling

1 Introduction

Nowadays business processes (BPs) are dynamic by nature and affected by a dynamically changing environment. Examples of such dynamic changes of environment include regulatory adaptations (e.g. change in raw material prices), market evolution (e.g. stock price change), changes in customer behaviour (e.g. rapid change in customer needs), process improvement, policy shifts and exceptions.

Traditional approaches used to model, simulate and implement BP no longer cover the actual needs of business, which should be more dynamic in order to be able to compete. There is thus a necessity to conduct research in the field of DBP modelling, simulation and their automation in information systems (IS), and to propose new solutions that correspond to a dynamic business's needs. Moreover, a business needs tools which support such DBP and allow it to find answers for the question of what-if.

Consequently, these business needs can be transformed into the requirements for research as follows:

- proposing a DBP modelling approach,
- proposing a DBP simulation approach,
- proposing an architecture and implementing it into a BP simulation engine.

Existing approaches to DBP modelling and simulation are incomplete because they are lacking in theory and in terms of implementation. Furthermore, there is no commonly accepted definition of DBP. Current BP modelling and simulation tools are suitable for modelling and simulation of a static BP that strictly predicts which activities, and in which sequence, to execute (e.g. Simprocess or ARIS 9.7). In the best case, today's tools and proposed methods allow for changing BP by executing different configurations of the BP, as in (Xiao et al., 2011), or using templates for modifying BP activity sequences, as in (Eijndhoven et al., 2008). This means that the majority of existing tools and approaches require strict specification of a BP, and unexpected sequences of BP activities cannot be included during a BP execution. Therefore, modelling and simulation of a dynamically changing BP, e.g. dynamic processes, is a topical, relevant and challenging task.

As advocated in a number of papers, e.g. in (Mejia Bernal et al., 2010; Hermosillo et al., 2010b; Milanovic et al., 2011), business rules are applicable for ensuring the dynamicity of BP.

The main opportunities of using business rules to ensure dynamicity of BP lie in the following: each activity in a process is selected according to the defined conditions at BP runtime; choice of activity content; and representing the changing DBP context. However, as presented in the works related to this paper, there is no complete approach or tool for rule- and context-based DBP modelling and simulation – i.e. none of the analysed tools (IBM Websphere (v.7.0 2014)[1], Simprocess (v 2015)[2], Simul8[3], AccuProcess[4] and ARIS 9.7[5]), which are widely used for BP modelling and simulation, supports changing business rules during the simulation of DBP. Some approaches, like (Hermosillo et al., 2010a), describe BP dynamicity using BP pointcuts, where adaptations can be made, or changes of BP are available in new instances of the BP, but not at the same instance, like in (Xiao et al., 2011).

In our paper, we propose an approach for rule- and context-based DBP modelling and simulation. We define DBP as a process with a not-fixed sequence of activities, i.e. activities for execution in the BP are selected according to the rule conditions and changing context of the environment. Moreover, activities can be changed at the same BP instance runtime. We propose an approach for changing a sequence of BP activities, their content, rules for selecting activities in DBP and the context of a DBP at process runtime. Based on the proposed approach, the reference architecture and a prototype of a DBP simulation tool was developed. A case study was carried out using this prototype, and it shows correspondence between the obtained results and needs of the dynamically changing business, as well as possibilities for DBP modelling and simulating.

The balance of this paper is organized as follows. Section 2 presents related works on DBP. Section 3 presents our proposed requirements of rule- and context-based DBP. Section 4 describes our approach for rule- and context-based DBP modelling and simulation and reference architecture for the rule- and context-based DBP modelling and simulation system (DRBPSimul). Section 5 presents a case study of an ordering system. Section 6 presents results and discussion. Finally, Section 7 concludes the paper.

2   Related works

In our research, we emphasize DBP where content and the sequence of activities depends on the context of the environment and can be changed at runtime. In contrast to DBP, static BP has a strict specification, i.e. the content and sequence of activities are defined before BP instance execution and cannot be changed at runtime. The remaining part of the work discusses four main aspects of a DBP. They are: existing definitions, models, realizations and simulation of a DBP.

---

[1]http://www-03.ibm.com/software/products/en/modeler-advanced.

[2]http://simprocess.com

[3]http://www.simul8.com

[4]http://bpmgeek.com/accuprocess-business-process-modeler

[5]http://www.softwareag.com/corporate/products/new_releases/aris9/more_capabilities/default.asp

2.1 Concept of a dynamic business process

However, there is a range of approaches within DBP modelling and simulation, the difference of these approaches lies in the understanding of the concept of dynamicity. Therefore, we start our analysis with the concept of BP dynamicity. The analysis of the related works shows that four main concepts are popular for defining the ability of a BP to adapt to the changing environment. They are as follows:

- dynamicity,
- flexibility,
- agility and
- adaptability.

Gong and Janssen (2012) state that the concept of flexibility is more appropriate in comparison to agility within the context of a BP. According to them, the concepts of agility and flexibility overlap and represent a system's ability to respond to changes in the environment. Agility in this case has a stronger emphasis on the speed aspect than on flexibility. According to Pucher (2010), dynamic BP is a variant of agile process and enables a business user to make changes in the process at runtime, for example, by selecting a different sub-process at predefined decision points. Another kind of a flexible BP is an ad-hoc BP that has no underlying process definition and consists of a set of activities relating to possible contents (Pucher, 2010; Dustdar et al., 2005; Bizagi, 2016). In an ad-hoc process, a user decides what to do and when to do it. The most flexible BP, according to Pucher (2010), is adaptive BP, which is a process that changes according to the environmental conditions at runtime. Moreover, the execution of adaptive BP instance influences execution of the next BP instance, e.g. real-time knowledge from the last process execution can influence the execution of the next one.

In spite of the above, in many cases BP dynamicity, flexibility, agility and adaptability are used as synonyms. For example, organizations like (Wikipedia[6], Gartner, Inc.[7], WhatIs.com[8]) and research papers, like (Adams, 2010) and (Pesic and Aalst, 2006) define the dynamicity of BP as: the ability to react to changing conditions (internal and/or external) of operation, according to the client's individual needs, in an appropriate and timely manner at process instance runtime without having a negative impact on the process essence or its expected completion. Management of such DBP is understood as the ability to support process change by any role, at any time, with very low latency.

The differences in DBP definitions and modelling and simulation approaches can be found by analysing the level of dynamicity of BP. As proposed by WhatIs.com, there are three levels of DBP. The first, and lowest, level of dynamicity is described as using decision points, in which a human or an automated system decides what to do next according to predefined rules. Almost all of today's approaches and tools implement this level of dynamicity.

The second, and middle, level of dynamicity allows for automatic configurations of a BP, like choosing an alternative template for processing activities or changing the order of activities in a process if conditions change. A number of approaches, proposing dynamic adaptation using variability models, are presented. They are, like Hallerbach et al. (2008b), Alférez et al. (2014), Milani et al. (2016) etc. However, all those methods are based on the idea that business process consists of variable and not variable segments and only variable segments of a process could be changed, in some cases, like in (@ Alférez et al., 2014), at runtime.

Approaches presented by Eijndhoven et al. (2008), Hermosillo et al. (2010b), Hermosillo et al. (2010a) fall into this level of dynamicity, since their main idea is to define decision points in a process and describe those decisions through business rules. The approach, presented in (Xiao et al., 2011), also falls into this level of dynamicity. Although Xiao et al. (2011) suggest constructing a process from a number of reusable fragments, new process schemas generated according to the new conditions are applicable only at the next process instance execution (Berkane et al., 2012). In their approach (Hildebrandt and Mukkamala, 2011), the authors propose a Dynamic Condition Response Graphs (DCRGraphs) model based on an event-based process model to ensure flexibility and graphical notation of workflows.

---

[6]http://en.wikipedia.org/wiki/Dynamic_business_process_management
[7]http://www.gartner.com/it-glossary/dynamic-business-process-management-bpm
[8]http://whatis.techtarget.com/definition/dynamic-BPM-business-process-management

The third and the highest level of dynamicity is goal-driven BP that can be changed over time and at runtime according to the new conditions and the customer's needs. The approaches, suggested in Pesic and Aalst (2006) and Adams (2010), are the most refined among those analysed, since they allow for constructing BP from a number of possible and reusable tasks at runtime. However, the authors say nothing about the possible goal-orientation of the dynamic execution of BP.

As can be summarized from the previous analysis, the BP dynamicity can be viewed through vertical and horizontal dimensions. The vertical dimension represents the DBP level, which defines the ability of a BP to adapt to changes in the environment. At the lowest level we have the least DBP, like, for example, a BP with decision points, and at the highest level we have the most DBP. This would, for example, be the case for a BP, which is adaptable at runtime and is goal-driven. The number of levels of dynamicity of a BP depends on the definition of dynamicity as a feature. Therefore, in section 3 of this paper we define requirement for DBP. The horizontal dimension can present implementation perspective, which could be execution, simulation or automation.

2.2 Context-sensitive BP

As advocated in a number of papers, dynamicity and adaptability of a BP is ensured through being context-sensitive BP (Saidani and Nurcan, 2007; Mejia Bernal et al., 2010; Nunes et al., 2011). Such a process is able to adapt the execution of its instances to the changing context (Saidani and Nurcan, 2007). Authors of Laznik and Jurič (2013) uses context for exception handling in BP execution language. From the BP modelling viewpoint, authors Rosemann and Recker (2006) define context as: "the minimum of variables containing all relevant information that impact the design and execution of a BP". A context-aware modelling framework is introduced in (Coutaz et al., 2005), where the authors define context as a set of entities, a set of roles that entities may satisfy, a set of relations between the entities, and a set of situations that denote specific configurations of entities, roles, or relations. As presented in the framework proposed by Coutaz et al. (2005), a context-aware system should include the situation and context identification layer for identifying changes in situations or in a context. The authors Mejia Bernal et al. (2010) do not specify how they define, manage and adopt changes in a context.

As can be summarised from the papers analysed above, context can be internal and external. An internal context is represented by a state of system resources, and an external context is represented by state of system environment expressed by a set of variables and rules. When an external context changes, i.e. something important happens outside the system, a system should react to this change. Those external changes, however, can cause internal changes in a system, and it is also necessary to react to these changes. For example, an external change in a context might be a change in share price, to which a system may react by buying shares. Of course, after this action the amount of money can be insufficient for other purchases. Therefore, the system should react to this internal event and, for example, borrow money.

Therefore, two contexts (internal and external) should be taken into account when modelling dynamic system. And yet, the existing approaches, like those analysed in (Saidani and Nurcan, 2007; Mejia Bernal et al., 2010; Nunes et al., 2011), allow for the definition of one type of context: internal or external only, but not both. For example, the authors Hu et al. (2013) define context as the current state of system resources. That said, in some application domains, only one type of context is important.

Other researchers propose to ensure BP dynamicity through an event-based approach. Authors such as Hermosillo et al. (2010a), Hermosillo et al. (2010b), Hermosillo (2012) suggest using complex event processing to respond dynamically to the actual state of an environment. However, not all authors define the events, i.e. external or internal or both, to which their system reacts. Authors Chandy and Schulte (2009) propose to extend the BPM engine with an Event Processing engine, which can obtain information about the events that happen outside and inside the BP, giving the system complete context-awareness when making decisions.

The importance of separating external context from internal context lies in the fact that BP activities could the change internal context of a system, but not the external context of the system.

2.3 Rule-based BP

One more direction for DBP modelling is the involvement of business rules and business rule-based modelling. We believe that the main opportunities of using business rules to ensure dynamicity of BP are as follows: each activity in a process is selected according to the defined conditions at BP runtime, and the content of an activity is chosen based on the changing internal and/or external context. However, the proposed rule-based approaches – some of which are reviewed in this section – do not cover all these aspects of DBP. In (Hermosillo et al., 2010b) the authors use rules to define the relationship between complex events and to filter the interesting ones in their proposed CEVICHE architecture. When an event that is important for BP adaptation is detected, the Complex Event Processing (CEP) engine notifies the Aspect Manager component to adapt the BP (instance or model) with the corresponding activity (by authors, aspect) at runtime. It is in line with this that the authors Hermosillo et al. (2010b), Bastinos Šaša and Lavbič (2015), Šaša and Vasilecas (2011) concentrate on complex events processing in their research; they (Hermosillo et al., 2010b) say nothing about internal and external context description. The authors Milanovic et al. (2011) use business rule patterns to enrich BP by possible cases and to increase BP flexibility. However, their proposition satisfies only the first level of BP dynamicity according to WhatIs.com. In (Pesic and Aalst, 2006) the authors use constraints to define relations among tasks in the ConDec language. At every moment during the execution of a process model, there is a judgement about whether or not the model is correct. Authors of (Charfi and Mezini, 2004; Charfi et al., 2006) uses the idea of Aspect-Oriented Programming for combining existing web services into more sophisticated web services. Authors of (Alférez et al., 2014) propose dynamic adaptation of service compositions with variability models at process runtime, as discussed above.

In (Mejia Bernal et al., 2010) the authors propose an approach, which consists of two main steps: decomposing process into ECA (Event-Condition-Action) rule set and adapting process to the context data. During the first step the authors describe how to express transitions between activities in the form of ECA rules, where an event is generated when an activity has finished its execution, a condition used to verify which workflow part is enabled, and where a rule action determines the next activity that has to be executed. During the next step the adaptation at runtime is performed. Mejia Bernal et al. (2010) use a Control Flow Checking (CFC) table to represent the dependencies between activities. When some modification is performed, a new version of the CFC table is generated and used to update workflow. However, as proposed in (Mejia Bernal et al., 2010), all processes have a strictly defined sequence of activities and this sequence of activities can be changed according to the defined set of rules.

2.4 Implementation aspects of BP

A particular part of the papers analysed is aimed at proposing approaches for dynamic workflow[9] modelling and simulation, like in (Traverso and Pistore, 2004; Rao et al., 2006; Döhring and Zimmermann, 2011). According to the levels of BP dynamicity proposed by WhatIs.com, almost all the studies fall into the first, and rarely into the second, level of dynamicity. The authors Sadiq et al. (2000) discuss different issues of dynamic workflow modelling according to change and time. They do not, however, present any approach or method for dynamic workflow modelling. In (Li et al., 2012) the authors use product structure trees for dynamic workflow modelling and visualization, but such an approach limits the dynamicity of workflows. In (Kwan and Balasubramanian, 1997) a framework for dynamic workflow modelling is presented. Though the framework has four perspectives in which where behavioural perspective is responsible for dynamic workflow modelling, the authors say nothing about implementation of the framework or its detailed application.

In this paper, we have briefly reviewed architectures proposed for implementing suggested approaches on rule- and context-based DBP modelling and simulation. However, in (Valença et al., 2013) only half of the analysed researches offered automated tool support or prototypes for DBP modelling and simulation. The authors Hermosillo et al. (2010a), Hermosillo et al. (2010b) propose the CEVICHE framework and deal with

---

[9]As defined by Muehlen (2004), a workflow is a specific representation of a process which is designed in such a way that the formal coordination mechanisms between activities, applications, and process participants can be controlled by an information system, by the so-called workflow management system - i.e. a workflow is a formal, or implementation-specific, representation of a BP.

architecture for DBP adaptation. As Pourshahid et al. (2012) state, the main drawback of the suggested approach is that the pointcuts and the adaptation rules should be predefined in the BP model and CEVICHE can only handle known specific cases. In other words, it is not a generic framework that can be used to improve the design of the process models. Rather, it executes already known alternative processes after detecting predefined situations. The authors Bui et al. (2013) propose a SOA-based architecture for their approach implementation. The main components of the architecture - implementing dynamic workflows - are described as follows. Aspect manager with Aspect database are responsible for determining advices from Advice repository, according to calculated values for variables in general processes. The main functions of Context server are context management, determining context changes from Adaptors and informing the Aspect manager about user-context changes. Service Discovery Manager is responsible for assisting care-receivers by searching for services, prioritizing them and selecting the most suitable ones. Other components are described in (Bui et al., 2013). However, the proposed architecture is strongly domain-specific, since it was developed for the healthcare domain. Another solution for the healthcare domain is presented in (Geebelen et al., 2010).

2.5   Analysis of existing tools on BP modelling and simulation

The next thing we analyse in this paper is the implementation of the proposed approaches. The concept of an implementation of the proposed approach could be viewed from three perspectives: case study, automation, or simulation. As the analysis shows, the biggest part of the approaches is implemented as a case study, like in (Mejia Bernal et al., 2010; Hermosillo et al., 2010b; Kwan and Balasubramanian, 1997), or an automation of the proposed approach at some level. For example, as presented in (Xiao et al., 2011), the proposed implementation (i.e. automation) of the approach allows for changing BP by presenting several different instances of the changed BP, or by presenting templates for BP execution, like in (Eijndhoven et al., 2008). However, not one of the previously analysed researches presents simulation of the DBP.

For more comprehensive study, rule-based BP modelling and simulation tools are compared in Tables 1 and 2.

The results of the comparison of rule-based BP modelling tools are presented in Table 1, where, based on the related works, criteria for comparison are divided into four groups: business rules usage within BP, business rules structuring and versioning, ensuring quality of business rules, and DBP simulation. The summary of the comparison presented in Table 1 is as follows:

- **IBM Websphere Business Modeler Advanced (v.7.0 2014)** is a BP modelling and analysis tool. It allows for simulating of dynamic BP by choosing an appropriate execution branch. However, it is not suitable for changing rules during the suspension of the process instance. It is possible to apply changes at the next BP instance.

- **Simprocess (v 2015)** is a process modelling and simulation tool which integrates Process mapping, hierarchical event-driven simulation, and activity-based costing (ABC). However, it is not suitable for dynamic BP simulation.

- **Simul8** is a process modelling and simulation tool that allows for defining dynamic parameters which can be changed according to the defined rules. However, this tool is not suitable for dynamic BP simulation.

- **AccuProcess Modeler** is a visual BP modelling software product that helps businesspeople to document, simulate and improve their BP. However, it has no dynamic BP simulation functionality.

- **ARIS 9.7** is a visual BP modelling and simulation software product, with incorporated ARIS Business Rules Designer, full support of BPMN 2.0 for BP modelling, and static BP simulation included in this version of the software.

As can be seen from Table 1, existing tools are well developed to model, analyse and simulate static BP. However, those tools are not suitable for simulation of DBP.

Another comparison of nine process management systems (PMS) (YAWL (Hofstede et al., 2010), DECLARE (Aalst et al., 2009), OPERA (Hagen and Alonso, 2000), ADEPT2 (Göser et al., 2007), ADOME (Chiu et al., 2000), AgentWork (Müller et al., 2004), ProCycle (Weber et al., 2009), WASA (Weske, 2001) and SmartPM

+ Continuous Planning) in terms of managing adaptability is presented in (Marrella and Mecella, 2011). The authors define adaptability as the capability to face exceptional changes – triggered by foreseeable or unforeseeable events and – that may require that BP instances be adapted in order to be carried out. The majority of PMSs use a case-based reasoning approach[10] to support adaptation of workflow specifications to changing circumstances. Adaptation remains manual, since users need to decide how to manage the events by choosing provided suggestions. Pre-planned approaches to exceptional changes (e.g. for each kind of failure that is envisioned to occur, a specific contingency process is defined a priori) are available in YAWL, DECLARE and OPERA. An unplanned adaptation is possible in SmartPM + Continuous Planning only.

To discern additional requirements of DBP and implementation of these requirements we review existing BRMS. A comparison of the five well-known BRMSs (Drools[11], OpenL Tablets[12], OpenRules[13], IBM ILog BRMS[14] and Blaze Advisor[15]) is presented in Table 2. For this comparison the set of attributes important for BP simulation was chosen (namely, Dynamic rules, documentation, possibility for use in simulation, open source, algorithm use, possibility of using several languages, and rules defined in a high level language).

[A] The engine monitors the state of the objects and is able to find out when the conditions of a rule become true, regardless of why this condition became true (Blaze Software, 1999).

During the process of choosing BRMS an important aspect is the possibility of defining rules for non-technical business users. A BRMS should have the feature to define business rules using high-level language, e.g. language that is closer to natural speech than programming language is. Unlike OpenL Tablets and OpenRules, Drools and IBM ILog BRMS allow users to specify business rules using low-level (e.g. close to programming language) languages. Therefore, those BRMS are more context-adaptable. Nevertheless, no BRMS analysed in Table 2 is suitable for dynamic simulation.

3   Requirements for a rule- and context-based DBP

According to the related works, we propose requirements for a DBP. To start, we present the definition of the context used in the requirements:

1. **Context** refers to internal context and external context.
2. An **external context** is a set of variables and context rules (i.e. business rules which define some policy of the environment) defining a particular state of the environment.
3. An **internal context** is a current state of system resources.
4. An **internal context change** is a change to a current state of system resources.
5. An **external context change** is a change to a current state of environment, like a change of variables describing environment or context rules.

The list of our proposed requirements for a DBP is as follows:

1. A DBP should not have a **predefined sequence of activities**.
    1. Every subsequent activity should be selected according to predefined rules and a context. Therefore, every subsequent BP instance may differ from the previous instance of the same BP.
    2. If there is no activity for further execution at a DBP runtime, it should be possible to do the following:
        1. to terminate the execution of a DBP instance;
        2. to define a new activity and concerning rules for a DBP instance execution.

---

[10]Case-based reasoning (CBR) is the way of solving new problems based on the solutions of similar past problems: users are supported to adapt processes by taking into account how previously similar events have been managed (Marrella and Mecella, 2011).

[11]http://www.drools.org/

[12]http://openl-tablets.sourceforge.net/

[13]http://openrules.com/

[14]http://www-01.ibm.com/software/integration/business-rule-management/jrules-family/

[15]http://www.fico.com/en/products/fico-blaze-advisor-decision-rules-management-system

Activities in DBP are selected according to the external and internal context and rules, i.e. DBP have no predefined sequence of activities. The advantage of implementing this requirement is that we increase dynamicity of a process. For example, in e-health process activities are selected according to the user's context, which describes user's health state. When a user's context changes, like blood pressure or temperature, necessary activities, like drug supply, should be selected for execution immediately. Moreover, in many cases, it is difficult or impossible to predict all possible changes of a context and predefine possible sequences of activities. Contrary to DBP, static processes have predefined sequence of activities and could be adopted only in predefined places of a process instance, like variation points.

2. **Context-based dynamicity**:
    1. It should be possible to define an external and internal context.
    2. A DBP should react to the change in a context.

As presented in Section 2.2, external context is expressed through variables and rules. Internal context describes state of system resources. A DBP should react to the changes of a context. The advantage of implementing this requirement is that at process instance runtime context can change and process should be adopted to those changes immediately at process instance runtime.

3. **Rule-based dynamicity**:
    1. It should be possible to define new business rules and to change or delete existing business rules at BP runtime.
    2. A DBP should react to the changes in business rules at process instance runtime.
    3. Every next activity in a DBP should be selected according to the predefined business rules.

Activities in DBP are selected according to the predefined rules. However, if it is necessary, additional rules can be defined or existing rules can be changed at process instance runtime. The advantage of implementing this requirement is that at instance runtime rules, expressing constraints of application domain, laws, etc., could change and, according to this requirement, it should be possible to implement those changes at process instance runtime.

4. **Any role** involved in DBP execution should **support DBP instance change**, i.e. change of activities or their sequence, according to the context and rules at runtime with possibly low latency.

According to this requirement, a DBP implementing system should react to changes immediately after those changes comes in force. Moreover, any role, involved in DBP execution, should support any process instance changes. The advantage of implementing this requirement is that the change of a context or business rules influences selection of the next activity immediately after a change comes in force at process instance runtime. Implementation of this requirement allows rising a process dynamicity. Otherwise, if we implement changes during next process instance, a process becomes static. The majority of analysed approaches (see Sections 2 and 6) do not meet this requirement.

5. Before selecting the next activity, the historical data of instances execution of the same DBP should be analysed and **the selected next activity should not cause execution of an unacceptable sequence of activities**, regarding to early gained experience.
    1. The historical data of each DBP instance execution should be stored in a log file.
    2. It should be possible to define executed instances of a DBP as a "good practice" and a "bad practice".
    3. It should be possible to select a suitable instance, labelled as a "good practice", from the historical data for repeated execution.
    4. Time, cost, etc. values should be calculated and stored for each executed DBP instance.
    5. Instances named "bad instance"" should not be executed.

This requirement describes that before executing any DBP instance, a historical data of this DBP should be analysed. The analysis of historical data allows determining so called "good" and "bad" instances of the same process. A "good instance" means the one that requires minimum resources, is completed within an optimal to this process period and the goal of the process is reached. A "bad instance" means that it utilise too many resources for its execution and/or do not reach a goal. Therefore, such instance is unacceptable. Moreover, during historical data analysis, it can be found what conditions, i.e. state of current context, leads

to such "bad instances". This analysis is necessary to prevent execution of "bad instances" and to propose suitable alternative sequence of activities for execution.

6. A DBP execution and selection of the next activity at DBP **runtime should be based on goal-oriented approach**.

In our paper, a goal-oriented approach means that we are focusing on achieving some state of the system, but not on the way of how to achieve some state of the system. Therefore, we implement goal through rules, which together with check, if we approaching to a goal or not, are used to select each next activity and/or define new activity for execution. Advantage of implementing this requirement is that we achieve more dynamicity and strong goal-orientation of BP.

We argue that a BP, which meets all presented requirements, is a DBP. Implementing all six requirements provides more freedom in BP modelling. However, this freedom can be constrained by adding business rules, which depend on the requirements of application domain. Moreover, the advantage of the implementing the proposed requirements is that it allows DBP being a goal-oriented.

4 A rule- and context-based DBP modelling and simulation approach

According to the DBP requirements defined in the previous section, we propose an approach for rule- and context-based DBP modelling and simulation. The approach is based on the fact that at BP runtime it is possible to change rules, BP actions and their sequence, depending on the new business system context.

The main idea of the approach is presented in Figure 1. The DBP simulation approach consists of these steps:

1. The historical data of the same BP, e.g. already executed instances of the same BP, are analysed. This activity is useful for distinguishing successful and unsuccessful BP simulation instances and using them for the current instance simulation.
2. The context of the DBP simulation is analysed.
3. According to the context, the set of rules are chosen for further selection of BP activities.
4. According to the rules, an activity is selected for execution. Here we may have three cases, as follows:
   1. one activity – in this case the selected activity is executed and we go to the first step;
   2. several activities – in this case the activity with greater priority is executed and we go to the first step; and
   3. no activities – in this case the process is finished or we have to define a new or select another activity for execution.

Moreover, an activity uses resources during its execution. For more about resources see (Vasilecas et al., 2015).

In accordance with the proposed rule- and context-based DBP simulation approach, a system architecture – DRBPSimul – was developed. It is presented in Figure 2.

The proposed architecture (Figure 2) consists of these components:

- **Simulation Interface** is responsible for the collaboration of the user with the Simulation engine;
- **Simulation engine** is responsible for the simulation execution. It works as described in the model presented in Figure 1. The Simulation engine consists of these components:
  - **The Activity engine**, which is responsible for the processing of the selected activities;
  - **The Rule engine**, which is responsible for the checking of the selected rules that correspond to the existing context. The Activity engine uses the Rule engine to select the activities for execution;
  - **The Context engine**, which is responsible for the context analysis and definition of the process instance context. The Rule engine uses the Context engine to determine which rules correspond to the existing context and should thus be used for the selection of the activities for future execution;
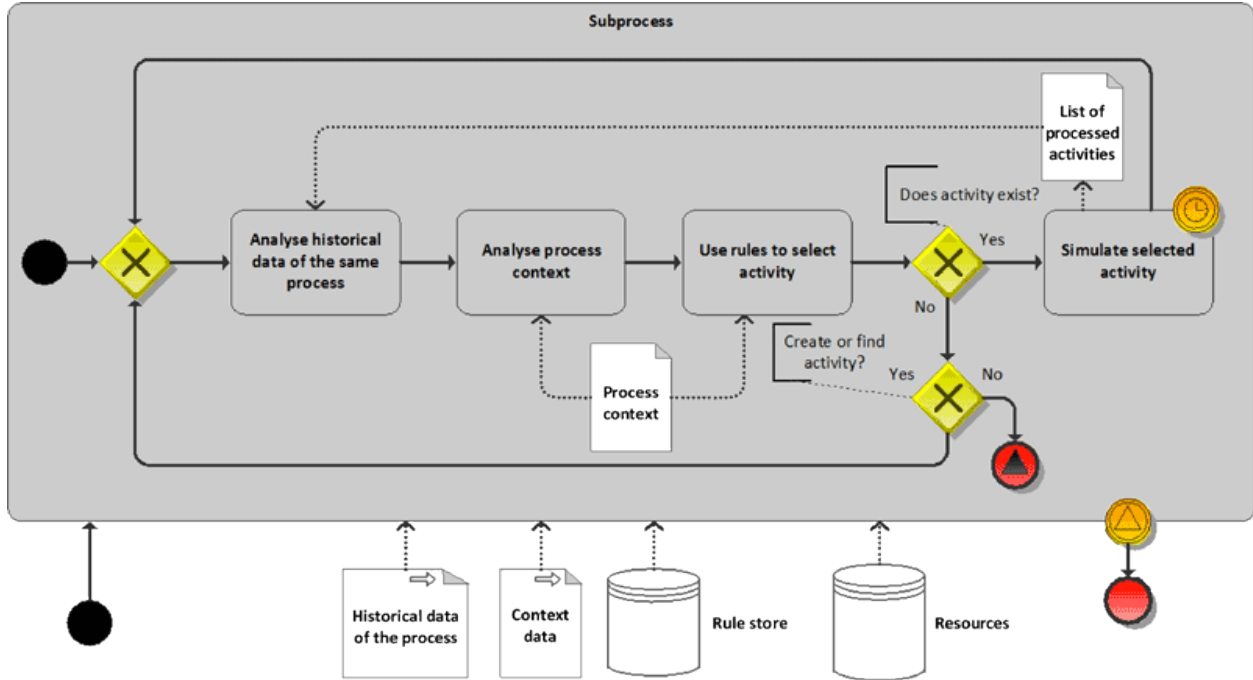
Figure 1: An approach for rule- and context-based DBP simulation

- **The Results analyser**, which is used to analyse historical data of the executed BP instances. The activity engine uses the Results analyser to check if the selected activity for the execution causes execution of a not-acceptable sequence of activities.
- **Storages** (Rule base, Activities storage, Context storage, Historical data storage and Resources) are responsible for storage of rules, activities, context, historical data and resources used for simulation execution and results analysis. Storages handle the persistence factors of entities (e.g., rules, activities, context, historical data and resources) by providing an interface to create, delete, and obtain entities and their templates for new entities by reference or query.
- **Manager** (Resources Manager, Rules Manager and Activities Manager) – responsible for managing (i.e. creating, deleting and getting entities their templates for new entities by reference or query) of entities through Management Interface.
- **External context** is a sensor used by Context storage to determine an external context for process instance simulation.
- **Resources** are resources used for the execution of BP. This component is used to determine an internal context state and save it in Context storage. Moreover, Resources are used by the Activity engine to process selected activities. For more about resources see (Kalibatiene et al., 2015; Vasilecas et al., 2015).

Before the simulation, Manager (see Figure 2) should load activities, rules and resources into Storages and allocate resources to activities. When the loading is over, an Analyst or Business user can start the simulation process. The simulation process is managed through Simulation Interface. E.g., the simulation process can be started, stopped, continued and modified. The simulation process is executed and managed by Simulation engine, which gets commands from Simulation Interface. When Simulation engine gets a message from the interface to start a simulation process, it sends a message to Context Engine and Rule Engine to check existing context and to determine which rules correspond to it and should be used for the selection of the activities for future execution. After the corresponding rules are determined, Simulation engine sends a message to Activity Engine to choose an activity according to the rules. Moreover, Activity Engine uses Results Analyser to analyse the Historical data of the same DBP. When activity is selected, it is executed by Activity Engine. After executing a selected activity, Activity Engine sends results of an activity execution to Simulation engine, which in turn repeats the process from determining current context and selecting next
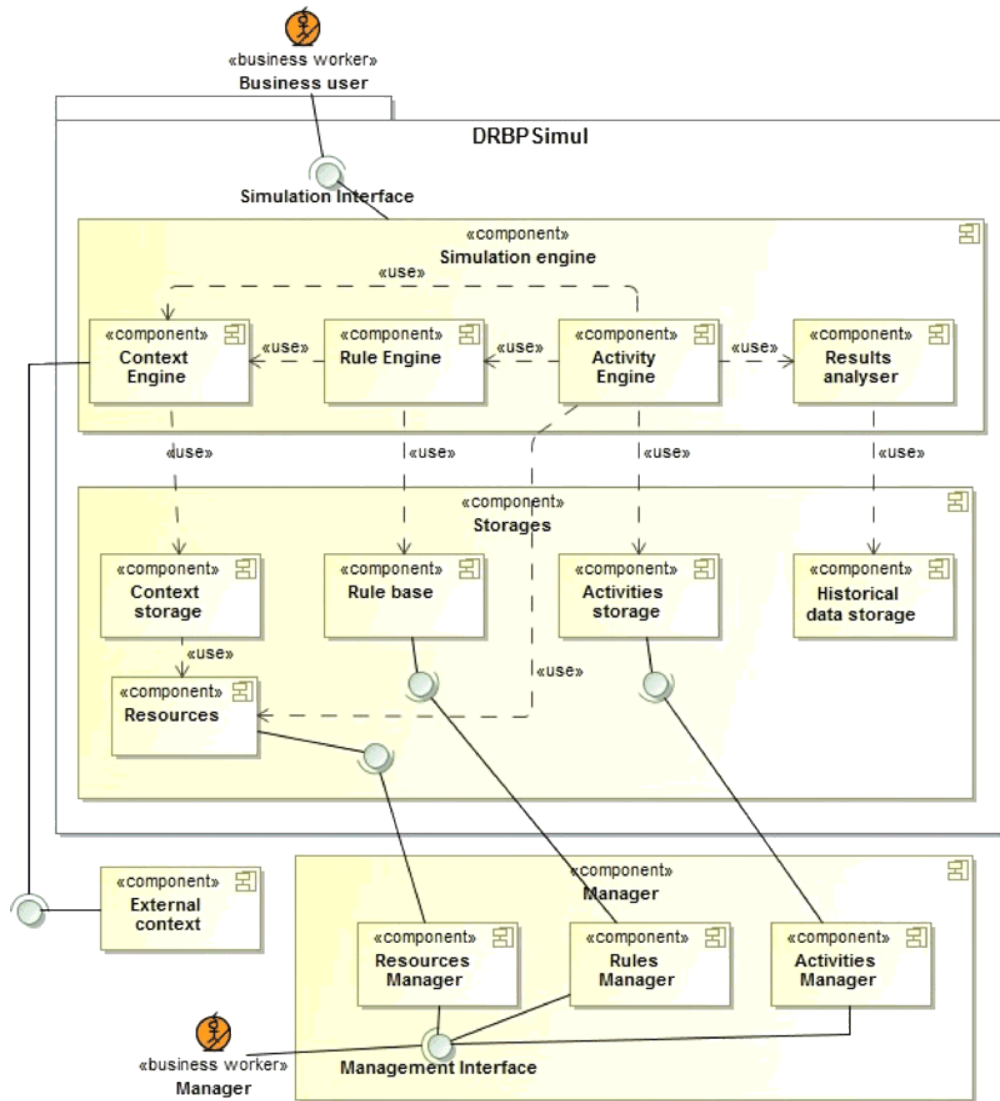
Figure 2: The architecture of the rule- and context-based DBP modelling and simulation system (DRBPSimul)
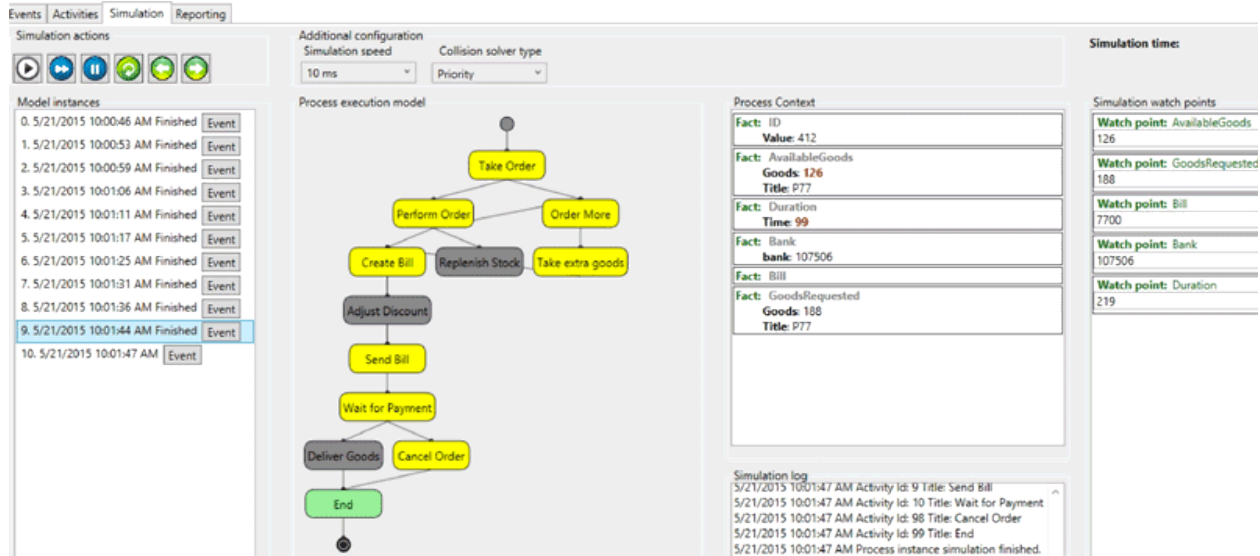
Figure 3: A simulation example of ordering process

activity for the execution.

The proposed architecture belongs to the reference level, i.e. it is independent of the implementation solutions, which could be used to implement it. For example, a service-oriented approach (SOA) may be used for the implementation of the architecture, like in (Hermosillo, 2012; Marrella and Mecella, 2011; Xiao et al., 2011; Berkane et al., 2012); the authors use a SOA for rules execution. This does not mean that other ways of implementing DBP are neglected.

5   A case study of rule- and context-based DBP simulation

The proposed rule- and context-based DBP simulation approach was realised using Microsoft technologies: Visual Studio 2012[16], C#[17] and .NET platform 4.5[18]. Windows Presentation Foundation[19] components were used to develop the user interface. Microsoft technologies were used because the university where the prototype of the system was developed has licenses for those Microsoft products. An implementation, named DRBPSimul, and an experiment were developed in Information Systems Research Laboratory[20].

An ordering system was used in this case study for demonstrating the rule- and context-based DBP simulation tool. A simulation example is presented in Figure 3.

The description of Figure 3 is presented as follows:

1. The first column on the left presents the list of events, which were detected and processed during the simulation. In the presented case, the event "Receive an order" and its triggered process are presented in Figure 3.

2. The second column on the left presents the graph of the simulated process instance. The activities presented in yellow are those executed in this process instance simulation; those in grey are activities which are not executed in this process instance simulation but which were simulated earlier in previous

---

[16]https://www.visualstudio.com/

[17]https://code.msdn.microsoft.com/site/search?f%5B0%5D.Type=ProgrammingLanguage& f%5B0%5D.Value=C%23&f%5B0%5D.Text=C%23

[18]https://www.microsoft.com/en-us/download/details.aspx?id=30653

[19]https://msdn.microsoft.com/en-us/library/aa970268%28v=vs.110%29.aspx

[20]Information Systems Research Laboratory, Institute of Applied Computer Science, Faculty of Fundamental Sciences, Vilnius Gediminas Technical University, http://fm.vgtu.lt/faculties/departments/institute-of-applied-computer-science/departaments/information-systems-research-laboratory/73139
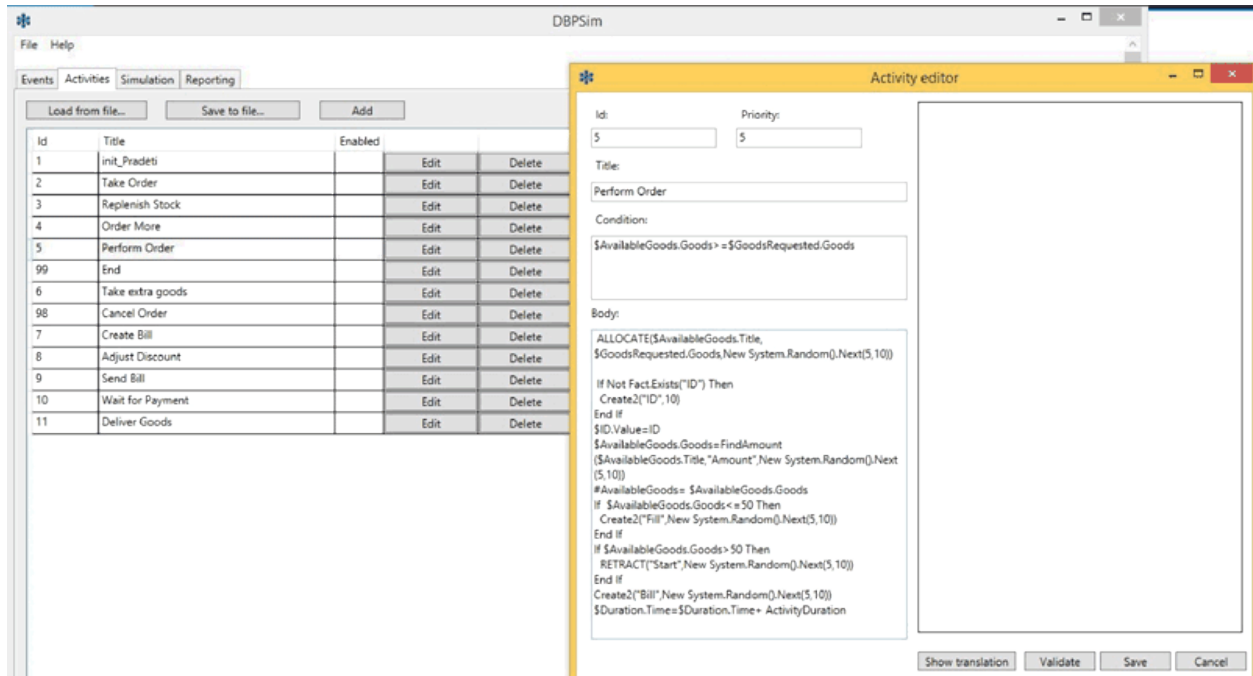
Figure 4: Example of changing the rules

process instances; those in green are activities that have just been simulated. As can be seen from Figure 3, the presentation of a graph depends on the executed activities. If all simulated instances are the same, we will have a sequential chain of activities. However, if simulated instances differ, we will have a graph that is coloured differently than the one presented in Figure 3.

3. The third column on the left presents the process context and its change during execution of activities. The process context can change after execution of each activity. The simulation log is located below the process context.

4. The fourth column on the left presents simulation "Watch points".

It is possible to change rules and context during the simulation. Rules could be changed through the Activities tab at the Body placeholder, as presented in Figure 4. After changing a rule, we can continue a simulation of a BP instance.

The proposed conceptual architecture presented in Figure 2 is now partially realised as follows:

- Simulation Interface, Activity Engine, Rule Engine, Rule base, Activities storages and Manager are realised;
- Context Engine, Context Storage and Resources are realised partially;
- Results Analyser and Historical data storage are least realized.

6 Main results and discussion

During our research, we obtained the following results.

The analysis of the related works, including our research conducted on BP modelling and simulation, allows us to formulate a theoretical background of a dynamic business process (DBP), as well as to propose an extended definition and to specify requirements of the rule- and context-based DBP.

Based on the proposed DBP requirements, an approach for rule- and context-based DBP modelling and simulation is presented. The main idea of the approach is that during the simulation of the same BP instance a rule set and a context can be changed. Those changes can be observed within the same BP instance immediately after changes come into force, e.g. activities and their sequence can be changed at runtime. Consequently, DBP has no predefined sequence of activities, i.e. each subsequent activity for execution is selected after observing an existing context and rules. Moreover, new activities can be created and existing activities can be modified at DBP runtime.

To implement the proposed approach, a reference architecture for rule- and context-based DBP modelling and simulation tool was developed. The architecture presents the main components and their relationships, which are necessary to support the proposed approach. Our contribution is this: unlike in the majority of the analysed research, our proposed architecture is not domain-specific. Moreover, components and their relationships are organized in such a way as to support rule- and context-based DBP modelling and simulation. Additional components, such as Results analyser linked to Historical data storage, are also included into the architecture in order to add intelligent functionality to the approach.

The proposed architecture was implemented into the rule- and context-based DBP modelling and simulation prototype, and an experiment on ordering system business processes was carried out. An experiment shows that instances for the same BP differ in appearance of specific activities when a different context and rules are applied (see Figure 3). Moreover, we can observe the dynamicity of a BP.

The comparison of the proposed approach with the other analysed approaches according to the proposed requirements (see Section 3) is presented in Table 3. The first column lists the 11 approaches considered, including our proposed approach. Each approach is identified by a sequence number and a reference to the publication/s as follows:

- **A1** = CEVICHE, (Hermosillo et al., 2010a,b), where implementation includes some examples of code,
- **A2**, (Eijndhoven et al., 2008), where implementation includes prototype and case study with some examples of code,
- **A3**, (Bui et al., 2013), where implementation includes prototype with test cases from the homecare domain,
- **A4**, (Yao et al., 2012), where implementation includes prototype, print screens, from on-boarding customers to IT outsourcing,
- **A5**, (Döhring et al., 2010), where implementation includes prototype and print screens,
- **A6** = PROVOP, (Hallerbach et al., 2008c,b,a, 2010), where implementation includes prototype and print screens from automotive and healthcare industries,
- **A7** = DYPROTO, (Wörzberger and Heer, 2011; Heer et al., 2009), where implementation includes prototype, without examples,
- **A8** = RoDP, (Hu et al., 2013), where implementation includes example with Mathlab simulation experiment,
- **A9** = BPFAMA (Boukhebouze et al., 2011), where implementation includes prototype, print screens and example of purchase order,
- **A10**, (Mejia Bernal et al., 2010), where implementation includes case study with examples,
- **A11** = **Our proposed approach**, where implementation includes prototype and print screens.

The other columns indicate the requirements and to what extent the given approach covers each requirement.

[B] The authors Bui et al. (2013), Yao et al. (2012) identify user-context, which can be described by a set of variables characterizing the user's needs (i.e. time of taking medications, blood pressure, etc. in (Bui et al., 2013)). We make a presumption that it is an external context, since the most variables belongs on concrete situation (i.e. a user).

[C] The authors Döhring et al. (2010), Hallerbach et al. (2008c), Hallerbach et al. (2008b), Hallerbach et al. (2008a), Hallerbach et al. (2010) define context by context variables, and context rules are applied to define the behaviour of the process in concrete situations. They do not clearly exclude what type of context (internal or external) is defined. In most cases, it is the user's needs. Therefore, we make a presumption that it is an external context, since most variables pertain to a concrete situation (i.e. a user).

Reviewing column-by-column, we can observe these results: the analysed approaches (except for our proposed approach) deal with the initial BP model, where sequence of process activities can be changed only in variation points according to the predefined rules.

Regarding the context, the majority of authors describe one type of context, i.e. external or internal, since, in the authors' analysed cases, it suffices to have one type of context - though not all authors specify how to define context. The common description of a context, also used by Bui et al. (2013) and Yao et al. (2012), consists of variables and rules.

All the analysed approaches deal with rules, and their application differs. For example, in (Hermosillo et al., 2010b,a) the authors use rules to define complex events. The authors of the analysed approaches consider the definition of new rules, or changing or deleting existing rules, before process simulation (see **R3.1** in Table 3). Nothing, however, is said about the possibility of changing rules at runtime and about how to react to those changes (see **R3.2** in Table 3). Moreover, since the basic ideas of the approaches vary, the analysed approaches differ in terms of how each subsequent activity is executed in accordance with the defined rules (see **R3.3** in Table 3). For example, in (Eijndhoven et al., 2008) only variable segments of a process can be changed according to predefined rules. Such a view limits the degree of dynamicity of a process.

The analysis of the approaches according to process modification at runtime (see **R4** in Table 3) shows that all approaches meet this requirement. However, authors do not show clearly how to implement this requirement.

As can be seen from Table 3, historical data is used only in a few approaches, like in (Yao et al., 2012), PROVOP and (Boukhebouze et al., 2011). But the usage of historical data is weak. The advantage of selecting "good practice" and "bad practice" (see **R5** in Table 3) is not used in the analysed approaches.

Regarding the **implementation**, the analysis shows that the biggest part of the approaches is implemented as examples or case studies, like in (Mejia Bernal et al., 2010; Hermosillo et al., 2010a,b) or an automation of the proposed approach at some level, like in (Eijndhoven et al., 2008). In half of the analysed studies, the authors present a small piece of code and some diagrams or examples from which it is difficult or impossible to determine the level of implementation of the approaches. Therefore, it is not clear which part of the method/approach is implemented. Though authors like Bui et al. (2013), Yao et al. (2012), Döhring et al. (2010), PROVOP and Boukhebouze et al. (2011) present prototypes and some print screens that enhance the proposed approaches, no author presents a process simulation.

From the presented results, it can be summarized that the main difference between our proposed approach and those analysed is that ours is goal-oriented. In such a process, every next activity for execution is selected according to the context (variables and rules describing context) and business rules. The analysed approaches deal with an initial process model, which can be changed at variation points according to the new user's needs. The main contribution of our proposed approach is that we ensure dynamicity of a process and allow implementing goal-oriented approach. Contrary, other approaches are based on an initial process model that limits the dynamicity and BP can be changed only in predefined places in a process – e.g. variation points, like in (Eijndhoven et al., 2008; Bui et al., 2013) or pointcuts, like in (Hermosillo et al., 2010a,b).

Moreover, in our approach, during the process simulation, we can change rules and react to the changing of context at runtime – something that is not supported in all the analysed approaches.

Our proposed approach meets all defined requirements, unlike other approaches, which meet only a particular set of requirements. This can be explained by the fact that other studies aim to solve slightly different types of domain-specific problems – that is, the majority of approaches are targeted at meeting the requirements of particular application domains. In contrast, our approach is general and is not dependent on a specific domain.

Implementing all six requirements provides more freedom in changing the BP than is possible in the analysed approaches. However, this freedom can be constrained by rules. Our developed prototype and obtained results show that the proposed method is implementable. Moreover, as can be seen in Figure 3, for a BP with the same goal different sets of activities occur when different rules are applied when different contexts appear.

Below we try to explain each requirement in term of our presented case study.

1. Implementation of the first requirement can be seen in Figure 3. Our process has no predefined sequence of activities. Each activity is selected according to the context and rules. It can be seen that different instances of the same process has different sequence of activities.

2. Implementation of the second requirement can be seen in Figure 3 column three (Process Context). In this case, we are working with internal context, i.e. resources.

3. Implementation of the third requirement can be seen in Figure 4, where we present a window with rules, which can be changed during paused BP simulation.

4. The forth requirement is implemented manually, when it is necessary to implement changes, like a rule change, we manually pause the simulation, change the rule and continue the simulation of the same instance.

5. The fifth requirement is implemented by adding rules, which prevent execution of "bad instances".

6. The sixth requirement (goal-orientation) is implemented through defined rules, which example can be seen in Figure 4.

7 Conclusions and future works

This analysis of the concept of a dynamic business process (DBP) shows that there is no commonly accepted DBP definition and neither is there a complete approach or tool for DBP modelling and simulation. Current BP modelling and simulation tools are mainly suitable for modelling and simulating a static BP which strictly predicts activities and the sequence of their execution. Today's tools and proposed methods allow us, at best, to change BP by presenting several different instances of the changed BP or presenting templates for BP execution. This means that the majority of existing tools and approaches require strict BP specification of and that an unexpected sequence of BP activities cannot be executed.

In this paper, we proposed a new approach for rule- and context-based DBP modelling and simulation. Our main contribution is that we have proposed the requirements for a DBP and according to those requirements developed a model for rule- and context-based goal-oriented DBP modelling and simulation. The main idea of the approach is that during the simulation of the same BP instance, rules and a context can be changed and those changes influence selection of the next activity for execution. Thus, activities and their sequence will be changed within the same instance immediately after rules and context changes come into force at runtime. Consequently, the DBP has no strict activities and their sequence, i.e. each subsequent activity for execution, is selected after observing an existing context and rules regarding to the defined goal.

According to the proposed approach, an architecture for rule- and context-based DBP modelling and simulation is put forth. This architecture was implemented into a prototype and the case study shows correspondence to the needs of the dynamically changing business, as well as possibilities for modelling and simulating DBP.

The topics for future research are as follows:

1. Refining and improving a set of requirements.

    1. Improving an implementation of a goal-oriented approach in DBP simulation.
    2. A requirement, that a DBP should meet requirements of a regular BP, should be analysed in more details and implemented. This type of requirements concerns with correctness and consistency of a DBP. Now, this kind of requirements is implemented manually. For example, Analyst is responsible for process correctness, since, he defines necessary rules and should ensure that deadlocks not arise.

2. Historical data of simulations should be saved in the historical data storage and appropriate algorithms should be created and used to automate analysis of the stored historical data. During historical data analysis, conditions, i.e. state of current context, which lead to "bad instances" should be found.

3. Improving the developed prototype and making its detailed description.

4. Simulating parallel DBP with shared resources.

References

Aalst, W. M. P. v. d., Pesic, M., and Schonenberg, H. (2009). Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development*, 23(2):99–113.

Adams, M. (2010). Dynamic Workflow. In *Modern Business Process Automation*, pages 123–145. Springer, Berlin, Heidelberg.

Alférez, G. H., Pelechano, V., Mazo, R., Salinesi, C., and Diaz, D. (2014). Dynamic adaptation of service compositions with variability models. *Journal of Systems and Software*, 91:24–47.

Bastinos Šaša, A. and Lavbič, D. (2015). SPARQL-based framework for semantically-based event processing. In *Proceedings of the 25th International Conference on Information Modelling and Knowledge Bases (EJC 2015)*, pages 282–296, Maribor, Slovenia. 2015/EJC.

Berkane, M., Seinturier, L., and Boufaida, M. (2012). A Pattern-based architecture for dynamically adapting business processes. *The Fourth International Conferences on Pervasive Patterns and Applications (PATTERNS 2012)*, pages 29–35.

Bizagi (2016). Understanding ad hoc processes.

Blaze Software (1999). Blaze Advisor. Technical White Paper, version 2.5.

Boukhebouze, M., Amghar, Y., Benharkat, A.-N., and Maamar, Z. (2011). A rule-based approach to model and verify flexible business processes. *International Journal of Business Process Integration and Management*, 5(4):287–307.

Bui, D. V., Iacob, M. E., Sinderen, M. v., and Zarghami, A. (2013). Achieving Flexible Process Interoperability in the Homecare Domain through Aspect-Oriented Service Composition. In *Enterprise Interoperability*, Lecture Notes in Business Information Processing, pages 50–64. Springer, Berlin, Heidelberg.

Chandy, K. M. and Schulte, W. R. (2009). *Event Processing: Designing IT Systems for Agile Companies.* McGraw-Hill Education, New York, 1 edition edition.

Charfi, A. and Mezini, M. (2004). Aspect-oriented Web Service composition with AO4bpel. In *Proceedings of European Conference on Web Services (ECOWS 2004)*, volume 3250, pages 168–182. WOS:000224322700013.

Charfi, A., Schmeling, B., Heizenreder, A., and Mezini, M. (2006). Reliable, Secure, and Transacted Web Service Compositions with AO4bpel. In *2006 European Conference on Web Services (ECOWS'06)*, pages 23–34.

Chiu, D., Li, Q., and Karlapalem, K. (2000). A Logical Framework for Exception Handling in ADOME Workflow Management System. In *Advanced Information Systems Engineering*, Lecture Notes in Computer Science, pages 110–125. Springer, Berlin, Heidelberg.

Coutaz, J., Crowley, J. L., Dobson, S., and Garlan, D. (2005). Context is Key. *Communications of ACM*, 48(3):49–53.

Dustdar, S., Hoffmann, T., and van der Aalst, W. (2005). Mining of ad-hoc business processes with TeamLog. *Data & Knowledge Engineering*, 55(2):129–158.

Döhring, M. and Zimmermann, B. (2011). vBPMN: Event-Aware Workflow Variants by Weaving BPMN2 and Business Rules. In *Enterprise, Business-Process and Information Systems Modeling*, Lecture Notes in Business Information Processing, pages 332–341. Springer, Berlin, Heidelberg.

Döhring, M., Zimmermann, B., and Godehardt, E. (2010). Extended workflow flexibility using rule-based adaptation patterns with eventing semantics. volume 1, pages 195–200.

Eijndhoven, T. v., Iacob, M. E., and Ponisio, M. L. (2008). Achieving Business Process Flexibility with Business Rules. In *2008 12th International IEEE Enterprise Distributed Object Computing Conference*, pages 95–104.

Geebelen, K., Kulikowski, E., Truyen, E., and Joosen, W. (2010). A MVC Framework for Policy-Based Adaptation of Workflow Processes: A Case Study on Confidentiality. In *2010 IEEE International Conference on Web Services*, pages 401–408.

Gong, Y. and Janssen, M. (2012). From policy implementation to business process management: Principles for creating flexibility and agility. *Government Information Quarterly*, 29:S61–S71.

Göser, K., Jurisch, M., Acker, H., Kreher, U., Lauer, M., Rinderle-Ma, S., Reichert, M., and Dadam, P. (2007). Next-generation Process Management with ADEPT2. In *BPM'07 Demo Proceedings*, CEUR-WS.org Workshop Proceedings, pages 3–6. CEUR-WS.

Hagen, C. and Alonso, G. (2000). Exception handling in workflow management systems. *IEEE Transactions on Software Engineering*, 26(10):943–958.

Hallerbach, A., Bauer, T., and Reichert, M. (2008a). Context-based Configuration of Process Variants. In *3rd International Workshop on Technologies for Context-Aware Business Process Management (TCoB 2008)*, pages 31–40.

Hallerbach, A., Bauer, T., and Reichert, M. (2008b). Issues in Modeling Process Variants with Provop. In *Business Process Management Workshops*, Lecture Notes in Business Information Processing, pages 56–67. Springer, Berlin, Heidelberg.

Hallerbach, A., Bauer, T., and Reichert, M. (2008c). Managing Process Variants in the Process Lifecycle. In *10th Int'l Conf. on Enterprise Information Systems (ICEIS'08)*, pages 154–161.

Hallerbach, A., Bauer, T., and Reichert, M. (2010). Capturing variability in business process models: the Provop approach. *Journal of Software Maintenance and Evolution-Research and Practice*, 22(6-7):519–546. WOS:000283478200006.

Heer, T., Briem, C., and Woerzberger, R. (2009). Workflows in Dynamic Development Processes. In Ardagna, D., Mecella, M., and Yang, J., editors, *Business Process Management Workshops*, volume 17, pages 266–277. WOS:000272076900026.

Hermosillo, G. (2012). *Towards Creating Context-Aware Dynamically-Adaptable Business Processes Using Complex Event Processing - Semantic Scholar*. PhD Thesis, Université des Sciences et Technologie de Lille, Lille.

Hermosillo, G., Seinturier, L., and Duchien, L. (2010a). Creating Context-Adaptive Business Processes. In Maglio, P. P., Weske, M., Yang, J., and Fantinato, M., editors, *Service-Oriented Computing - Icsoc 2010, Proceedings*, volume 6470, pages 228–242. WOS:000286691600016.

Hermosillo, G., Seinturier, L., and Duchien, L. (2010b). Using Complex Event Processing for Dynamic Business Process Adaptation. In *2010 IEEE International Conference on Services Computing*, pages 466–473.

Hildebrandt, T. T. and Mukkamala, R. R. (2011). Declarative Event-Based Workflow as Distributed Dynamic Condition Response Graphs. In *Electronic Proceedings in Theoretical Computer Science*, volume 69, pages 59–73. arXiv: 1110.4161.

Hofstede, A. H. M., Aalst, W. M. P., Adams, M., and Rusell, N. (2010). *Modern Business Process Automation: YAWL and its Support Environment*. Springer, Berlin, Heidelberg.

Hu, G., Wu, B., and Chen, J. (2013). Dynamic Adaptation of Business Process Based on Context Changes: A Rule-Oriented Approach. In *Service-Oriented Computing – ICSOC 2013 Workshops*, Lecture Notes in Computer Science, pages 492–504. Springer, Cham.

Kalibatiene, D., Vasilecas, O., and Rusinaite, T. (2015). Implementing a rule-based dynamic business process modelling and simulation. In *2015 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pages 1–4.

Kwan, M. M. and Balasubramanian, P. R. (1997). Dynamic workflow management: a framework for modeling workflows. In *Proceedings of the Thirtieth Hawaii International Conference on System Sciences*, volume 4, pages 367–376 vol.4.

Laznik, J. and Jurič, M. B. (2013). Context aware exception handling in business process execution language. *Information and Software Technology*, 55(10):1751–1766.

Li, S., Shao, X., Zhang, Z., and Chang, J. (2012). Dynamic workflow modeling based on product structure tree. *Applied Mathematics and Information Sciences*, 6(3):751–757.

Marrella, A. and Mecella, M. (2011). Continuous Planning for Solving Business Process Adaptivity. In *Enterprise, Business-Process and Information Systems Modeling*, Lecture Notes in Business Information Processing, pages 118–132. Springer, Berlin, Heidelberg.

Mejia Bernal, J. F., Falcarin, P., Morisio, M., and Dai, J. (2010). Dynamic Context-aware Business Process: A Rule-based Approach Supported by Pattern Identification. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 470–474, New York, NY, USA. ACM.

Milani, F., Dumas, M., Ahmed, N., and Matulevičius, R. (2016). Modelling families of business process variants: A decomposition driven method. *Information Systems*, 56:55–72.

Milanovic, M., Gasevic, D., and Rocha, L. (2011). Modeling Flexible Business Processes with Business Rule Patterns. In *2011 IEEE 15th International Enterprise Distributed Object Computing Conference*, pages 65–74.

Muehlen, M. z. (2004). *Workflow-based Process Controlling. Foundation, Design, and Application of workflow-driven Process Information Systems.* Logos Verlag, Berlin.

Müller, R., Greiner, U., and Rahm, E. (2004). AgentWork: a workflow system supporting rule-based workflow adaptation. *Data & Knowledge Engineering*, 51(2):223–256.

Nunes, V. T., Werner, C. M. L., and Santoro, F. M. (2011). Dynamic process adaptation: A context-aware approach. In *Proceedings of the 2011 15th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pages 97–104.

Pesic, M. and Aalst, W. M. P. v. d. (2006). A Declarative Approach for Flexible Business Processes Management. In *Business Process Management Workshops*, Lecture Notes in Computer Science, pages 169–180. Springer, Berlin, Heidelberg.

Pourshahid, A., Amyot, D., Shamsaei, A., Mussbacher, G., and Weiss, M. (2012). A systematic review and assessment of aspect-oriented methods applied to business process adaptation. *Journal of Software*, 7(8):1816–1826.

Pucher, M. J. (2010). Agile-, AdHoc-, Dynamic-, Social-, or Adaptive BPM.

Rao, J., Küngas, P., and Matskin, M. (2006). Composition of Semantic Web services using Linear Logic theorem proving. *Information Systems*, 31(4):340–360.

Rosemann, M. and Recker, J. (2006). Context-aware process design: Exploring the extrinsic drivers for process flexibility. volume 236, pages 149–158.

Sadiq, S. W., Marjanovic, O., and Orlowska, M. E. (2000). Managing change and time in dynamic workflow processes. *International Journal of Cooperative Information Systems*, 09(01n02):93–116.

Saidani, O. and Nurcan, S. (2007). Towards Context Aware Business Process Modelling - Semantic Scholar. In *Workshop on Business Process Modeling, Development, and Support (BPMDS'07)*.

Traverso, P. and Pistore, M. (2004). Automated Composition of Semantic Web Services into Executable Processes. In *The Semantic Web – ISWC 2004*, Lecture Notes in Computer Science, pages 380–394. Springer, Berlin, Heidelberg.

Valença, G., Alves, C., Alves, V., and Niu, N. (2013). A systematic mapping study on business process variability. *International Journal of Computer Science & Information Technology*, 5(1):1–21.

Vasilecas, O., Kalibatiene, D., Rima, A., Birzniece, I., and Rudzajs, P. (2015). A resource model for the rule-based dynamic business process modelling and simulation. In *29th Annual European Simulation and Modelling Conference 2015, ESM 2015*, pages 36–41.

Weber, B., Reichert, M., Rinderle-Ma, S., and Wild, W. (2009). Providing integrated life cycle support in process-aware information systems. *International Journal of Cooperative Information Systems*, 18(01):115–165.

Weske, M. (2001). Formal foundation and conceptual design of dynamic adaptations in a workflow management system. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pages 10 pp.–.

Wörzberger, R. and Heer, T. (2011). DYPROTO - Tools for dynamic business processes. *International Journal of Business Process Integration and Management*, 5(4):324–343.

Xiao, Z., Cao, D., You, C., and Mei, H. (2011). Towards a Constraint-Based Framework for Dynamic Business Process Adaptation. In *2011 IEEE International Conference on Services Computing*, pages 685–692.

Yao, W., Basu, S., Li, J., and Stephenson, B. (2012). Modeling and Configuration of Process Variants for On-Boarding Customers to IT Outsourcing. In *2012 IEEE Ninth International Conference on Services Computing*, pages 415–422.

Šaša, A. and Vasilecas, O. (2011). Ontology-based support for complex events. *Elektronika ir Elektrotechnika*, (7):83–88.

Table 1: Comparison of rule-based BP simulation tools ("+" – has this feature, "+/−" – has this feature in part, "−" – does not have this feature).

Category	Comparison attribute	IBM WebSphere (v.7.0.2014)	Simprocess (v 2015)	Simul8	Accu Process	ARIS 9.7
Rules usage within BP	Business rule integration into BP	+	+	+	+	+
	Business rule inclusion into BP using decision tables	+	+	+	−	+
	Model generation using business rule templates	−	−	−	−	+
	Re-use of business rules and their templates	+	+	−	−	−
	Re-use of business rules and their templates in different projects	+	−	−	−	−
	Business rules modelling politics	+	+	−	−	−
	Trigger-based business rule calling	+	+	+	−	+
Rules structuring and versioning	Hierarchical business rule structure	−	−	−	−	−
	Rules versioning	+	+	+	−	−
	Changing the initial model data by rules during simulation	+	+	+	+	+
Ensuring quality of rules	Error checking in rules	+	+	+	+	+
	Comparison of business rules	+	−	−	−	+/−
	Automatic generation of reports	−	−	−	−	−
	Analysis of rules after their execution and/or during their execution	+	−	−	−	−
DBP simulation	DBP simulation	+/−	−	+/−	−	−
	Changing business rules during suspending the simulation of DBP	+/−	−	+/−	−	−
	Possibility to define a context (internal or external)	−	−	−	−	−
	Possibility to change context during execution	−	−	−	−	−
	Level of dynamicity according to WhatIs.com	first	first	first	first	first

Table 2: Comparison of BRMS ("+" – has this feature, "+/−" – has this feature in part, "−" – does not have this feature).

Comparison attribute	Drools	OpenL Tablets	OpenRules	IBM ILog	Blaze Advisor
Dynamic rules	−	−	+/−	+/−	Production rules
Documentation	+	+/−	+	+	+
Possibility to use for simulation	+	−	−	+	−
Open source	+	+	+	−	−
Algorithm used	Direct inference, Rete	Optimized sequential algorithm	Optimized sequential algorithm, Rete	Direct inference	Rules are triggered implic-itly$^{\text{A}}$
Possibility of using several languages	−	−	+	+	+/−
Rules defined in a high level language	+/−	+	+	+	+

Table 3: Comparison of the eleven approaches on rule- and context-based DBP modelling and simulation

Approach	R1	R2		R3			R4	R5					R6
		R2.1	R2.2	R3.1	R3.2	R3.3		R5.1	R5.2	R5.3	R5.4	R5.5	
A1	No	Yes, external, if defined in CEP. Does not detailed	Yes/No	Yes/No, in CEP	No	No, only when adaptation situation arises	Yes	No	No	No	No	Yes	No
A2	No	No	No	Yes/No	Nothind said	Yes	Yes	No	No	No	No	Yes	No
A3	No	Yes, external[B]	Yes	Yes/No	Nothind said	Yes	Yes	Yes	No	No	No	Yes	No
A4	No	Yes, external[B]	Yes	Yes/No	Nothind said	Yes	Yes	Yes	No	Yes	No	Yes	Yes
A5	No	Yes[C]	Yes	Yes/No	Nothind said	Yes	Yes	No	No	No	No	Yes	No
A6	No	Yes[C]	Yes	Yes/No	Nothind said	Yes	Yes	Yes	Yes	Yes/No	No	Yes	Yes/No
A7	No	No	No	Yes/No	Nothind said	Yes	Yes	No	No	No	No	Yes	No
A8	No	Yes, internal. Does not detailed	Yes	Yes/No	Nothind said	Yes	Yes	No	No	No	No	Yes	No
A9	No	No	No	Yes/No	Nothind said	Yes	Yes	Yes	Yes/No, diagnostic phase	No	No	Yes	No
A10	No	Yes, internal	Yes	Yes/No	Nothind said	Yes	Yes	No	No	No	No	Yes	No
A11	Yes	Yes, both	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes