Baltic J. Modern Computing, Vol. 12 (2024), No. 1, 408-419

A New Approach on Rule and Context Based Dynamic Business Process Simulation

Diana KALIBATIENE[1], Olegas VASILECAS[1], Titas SAVICKAS[1], Tadas VYSOCKIS[1], Vjačeslavs BOBROVS[2]

[1] Vilnius Gediminas Technical University, Sauletekio al. 11, Vilnius, Lithuania
[2]Riga Technical University, Azenes iela 12 - 201, Riga, Latvia

{diana.kalibatiene, olegas.vasilecas, titas.savickas, tadas.vysockis}@vgtu.lt, vjaceslavs.bobrovs@rtu.lv

Abstract. Nowadays business is rapidly changing to fulfil the customers' needs and to meet the requirements of a changing environment. All those changes of a business have to be implemented in business processes (BP) and their maintaining information systems. Therefore, a necessity for a new approach of dynamic business process (DBP) modelling arises. Moreover, business needs a tool suitable for DBP modelling, execution and simulation. This gap triggered significant research efforts over the past decade and a number of approaches to DBP modelling were proposed. However, a question of implementation of the proposed approaches is left without an answer. Therefore, in the paper we emphasize DBP simulation and existing DBP simulation tools. We review existing DBP modelling approaches, propose a new DBP modelling and simulation approach and develop a prototype for DBP simulation. A DBP simulation experiment was carried out using the developed prototype and obtained results are presented in the paper.

Keywords: dynamic business process, business rule, context, simulation, business process modelling.

1. Introduction

Traditional approaches used to model, simulate and implement business processes (BP) no longer cover the actual needs of the business, which should be more dynamic to fulfil the customers' needs, to meet the requirements of a changing environment and due to never before seen conditions. Moreover, business, i.e. industrial and service, is faced with a problem of minimizing resources, like time and cost, needed to serve a client, to develop a product or to fulfil a demand. To fulfil this gap of a business we need, firstly, an approach for dynamic business process (DBP) modelling, and, secondly, a DBP simulation tool, which allows investigating different process instances and evaluating the impact of changes on a BP with accuracy and speed.

This paper analyses existing DBP modelling approaches and their implementation. The main emphasize is placed on possibility to simulate DBP. Therefore, related works are reviewed and a rule- and context-based DBP modelling and simulation approach is presented.

The rest of the paper is organized as follows. Section 2 presents related works on DBP and simulation of BP. Section 3 presents an approach of rule- and context-based DBP modelling and simulation. Section 4 presents implementation architecture of the proposed approach and a case study of ordering process simulation. Section 5 presents results and discussion. Finally, Section 6 concludes the paper.

2. Related Works

In the research, we emphasize on DBP, where content and the sequence of activities depends on context of the environment and could be changed at runtime. The idea of a DBP has been described in our earlier papers, i.e. (Kalibatiene et al., 2015; Rusinaite, Vasilecas, 2015; Rusinaite et al., 2015), and more extended literature review on DBP is presented in (Rusinaite et al., 2016). Contrary to a DBP, usual BP specifications have static properties that are defined before BP instance execution and only simple modifications are supported or the BP instance cannot be changed at all at runtime.

Different approaches are proposed to ensure dynamicity of a BP. In the simplest case, a BP has decision points where a human or an automated system decides next step based on predefined rules. Almost all current BP modelling and simulation tools support it.

In more sophisticated approaches, like in (van Eijndhoven et al., 2008), a BP is divided in a variable and non-variable segments. Non-variable segments stay constant in all cases. Variable segments vary according to the predefined rules. Authors of (La Rosa et al., 2013) made a survey on process variability and distinguished eleven main approaches and eight subsumed approaches. In (Bui et al., 2013; Hermosillo et al., 2010; Yao et al., 2012) authors use join points, point-cuts or variation points to insert additional activities into BP according to the predefined rules. Context, describing external environment or resources of a system, can also be used to facilitate BP dynamics. In (Bui et al., 2013; Yao et al., 2012), a context is defined through variables and rules to present user's needs and to adopt BP based on them. In (Milani et al., 2016), authors propose to identify the main process, variations of each process and construct variation map to model families of BP variants.

In an even higher level of sophistication, authors, like (Mejia Bernal et al., 2010; Boukhebouze et al., 2011), propose to transform a BP into a set of event-condition-action (ECA) rules or some variation of ECA rules, like in (Boukhebouze et al., 2011), and after event arises, check the condition and perform a consequent action.

However, in all analysed approaches authors use an initial BP model. The most sophisticated approaches are based on the idea that there is *no initial process model*, e.g. each activity for execution is selected based on the rules and a context, the process is *goal-oriented*. In our previous researches (Kalibatiene et al., 2015; Rusinaite, Vasilecas, 2015; Rusinaite et al., 2015) we have analysed DBP and proposed that in a DBP it is possible to change BP rules, BP activities and their sequence at process instance runtime according to the new business system context and rules. Moreover, for the BP with same goal different sets of activities occur when different rules are applied and the context varies.

In this paper, we place emphasis on DBP *simulation*. van der Aalst (2010) defines simulation as "attempts to "mimic" real-life or hypothetical behaviour on a computer to see how processes or systems can be improved and to predict their performance under different circumstances". The advantages of a simulation are advocated in a number of papers. Main advantages are as follows (van der Aalst, 2010, 2013; Kellner et al., 1999): getting ideas on how to reduce costs; using in "what-if" analysis; analysing the correctness of a new model design; computing expected performance, etc.

There are a number of tools for process simulation. These tools are suitable to model and analyse the real-life work of organizations. However, the tools are closed-source and it makes difficult to analyse them. For example, SimCAD[1] and FlexSim[2] can be used to simulate manufacturing, healthcare and other processes. SimCad supports model changes at runtime; but does not support formal BP modelling semantics and notation. Some other tools used for BP simulation are overviewed in Section 5 (Table 1). However, we need a tool for DBP simulation. As stated in (van der Aalst, 2013), simulation is used less often than implementation due to the additional assumptions needed. Therefore, further in this section we present a short overview of what we need for simulation, i.e. we formulate criteria for BP simulation tools comparison.

Firstly, the main thing we have to do for simulating a BP is to create a BP simulation model. According to (Kellner et al., 1999), a *simulation model* is a computerized BP model, which possesses the BP characteristics and represents some dynamic system or phenomenon. A BP simulation is used to assess the dynamic behaviour of processes over time, i.e. the development of process and resource performance in reaction to changes or fluctuations of certain environment or system parameters (ARIS, 2016).

A standard approach to create a simulation model consists of three steps: domain analysis for domain model creation (O'Donnell, 2013), process execution analysis to model data distribution, and finally combination of model with distributions to create simulation models (Rus et al., 2016). This facilitates simulation, but it is not suitable in changing environments (Azab, AlGeddawy, 2012). DBP cannot be simulated using the static models.

To automate simulation model creation, authors of (Rozinat et al., 2009) proposed to exploit data existing in information systems and create a simulation model from event logs using process-mining techniques. Authors applied process mining techniques as follows: 1) a control-flow discovery algorithm to create a process model by examining *activities* and *their relationships* in the log; 2) a decision point analysis to discover *decision rules* for the choice points; 3) a performance analysis to enhance information about *execution times* and *waiting times* for the activities, and probabilities for taking alternative paths; and 4) a role discovery algorithm to group *resources* into roles, and to associate the discovered *roles with the activities* in the process. Finally, the mining results enhanced with data, performance, and organizational characteristics are integrated in one comprehensive simulation model. For more details, see (Rozinat et al., 2009). Consequently, a simulation model consists of BP model, resource model, environment model, behavioural model and rules.

As stated by (van der Aalst, 2010), the main problem of the poor quality of simulation is oversimplified simulation models. This problem can be divided into three parts: 1) correctness of the process model; 2) enough data to parameterize the model, and 3) enough detailed model. Training people and a better validation of the model, e.g., using process-mining techniques, learning from events logs, etc., can address the first

---

[1] *http://www.createasoft.com/process-simulation-software*

[2] *https://www.flexsim.com/flexsim/*

two parts. Authors of (van der Aalst, 2010) believe that the biggest problem of current BP simulation approaches is that *human resources* are modelled poorly. Therefore, a comprehensive resource model should be proposed to ensure the quality of the BP simulation. However, authors of (van der Aalst, 2010) provide only guidelines to the human resource model.

Since we emphasize DBP simulation, it is necessary to perform a research on DBP also. However, here we present short overview of our previous researches on DBP. In our research (Kalibatiene et al., 2015; Rusinaite, Vasilecas, 2015; Rusinaite et al., 2015), we overviewed the concept of DBP and approaches on DBP and it was determined that DBP is a BP that has no predefined sequence of activities. The activity for the execution should be selected after evaluation of the external and internal context according to the predefined rules. Moreover, context and rules may change during the process instance execution. Therefore, it should be possible to make those changes at process instance runtime. The requirements for DBP are presented in (Rusinaite et al., 2015). In (Rusinaite et al., 2016), the authors present a more extended literature review on DBP. In Section 3, we present a refined context- and rule-based DBP simulation approach.

3. A DBP simulation approach

Based on our research and related works in Fig. 1 we present a context- and rule- based DBP simulation approach. The main steps of the approach are as follows:

1. After the process has been triggered, two parallel activities are performed:
   1.1. External and internal contexts of the process are analysed.
   1.2. Historical data of the same processes are analysed.
2. After the analysis results are obtained, it is determined what next should be done. There are two possible ways to simulate the process:
   2.1. Choose the best previously executed instance model of the process simulation from the historical data storage and simulate it according to the context. In this case, no more modifications are made.
   2.2. Simulate BP instance dynamically. According to the rules, select an activity from the non-simulated activities list. Here we may have three cases, as follows: a) one activity – in this case the selected activity is simulated and we go to the step 1; b) several activities – in this case collision solving is performed to select the activity with greater priority is simulated and we go to the step 1; and c) no activities – in this case the process is finished or we have to define a new or select another activity for simulation.

During the simulation, an activity uses resources. In our previous research (Vasilecas et al., 2015) we have proposed a resource model for DBP simulation. The main idea of the proposition is that resources are allocated to the activity before its simulation, but not before the simulation of all BP instance. Such dynamic allocation is necessary, since at the beginning of the BP simulation it is not clear which activities are going to be simulated.

Another contribution of our proposition of resource model is that resources can have different characteristics, as was presented and suggested in (van der Aalst, 2010), which allows detailed rule-based resource allocation and management during simulation.

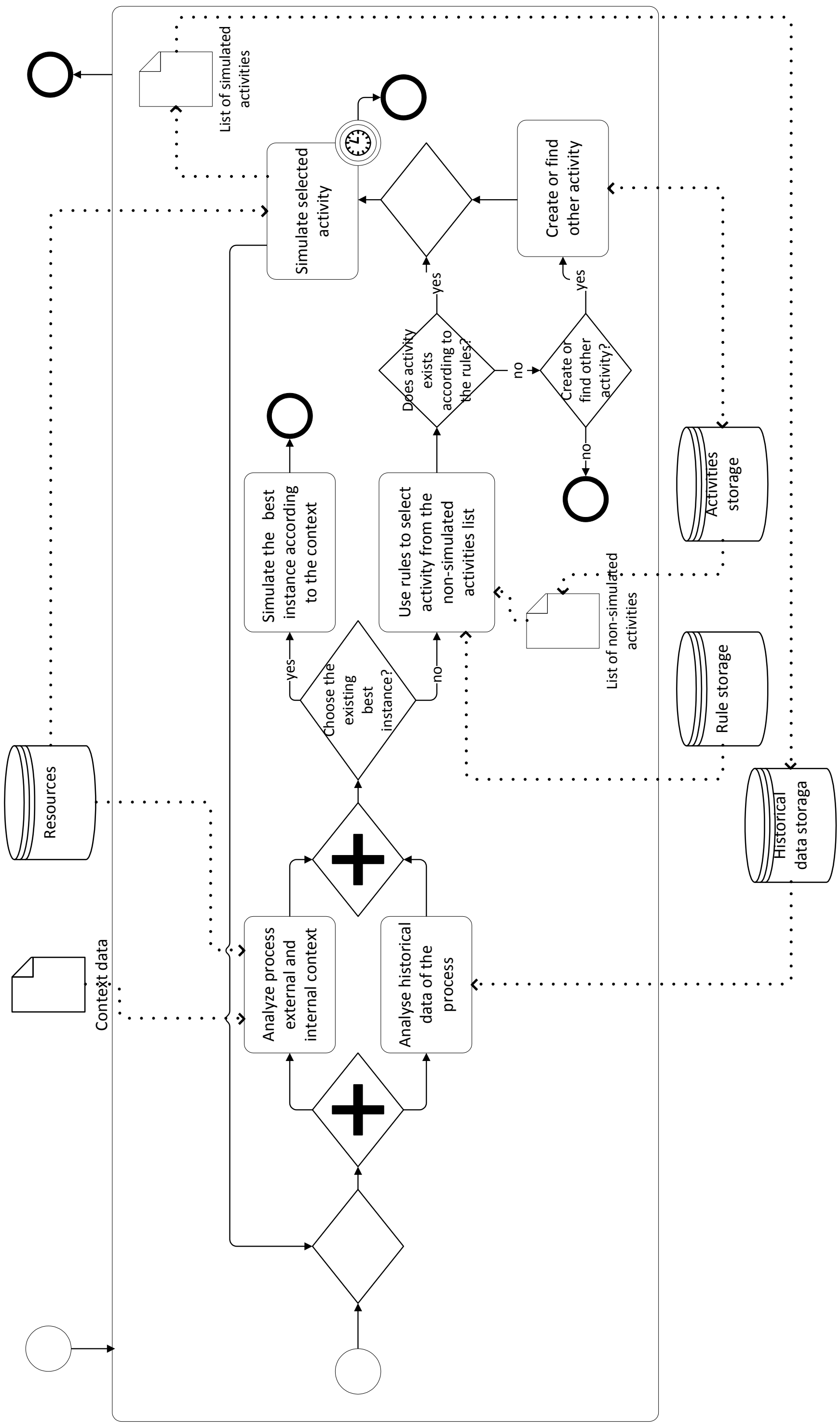

Figure 1. A context- and rule-based dynamic business process simulation approach

4. Case study on implementation of a DBP simulation approach

The proposed simulation approach was implemented using Microsoft technologies. An architecture of the implemented simulation prototype is presented in Fig. 2.

The user facing component is *GUI (Graphical User Interface)* and it is responsible for the interaction between the Analyst and system components. The *Simulation GUI* also allows to start, speed-up, step through or suspend the simulation. The *Analyser* visualizes simulation results and *Activity GUI*, *Event GUI*, *Resource GUI* components facilitate interaction between Analyst and *Manager*, which is used insert new activities, events or resources into Storages, modify or delete existing activities, events or resources.

The *Manager* is responsible for the management (insertion, deletion and modification) of activities, events and resources in Storages.

The *Simulation Engine* is responsible for the simulation execution. It works as described in the model presented in Fig. 1. The Simulation Engine consists of the following components:

- the *Simulation Controller*, which is responsible for simulation of a BP;
- the *Context Engine*, which is responsible for the definition of the context according to the resources in Resource DB;
- the *Activity Selector*, which is responsible for the selection of activities according to the conditions and current context and passing the selected activities to Activities Executor through Simulation Controller; and
- the *Activities Executor*, which is responsible for the execution of the selected activities.

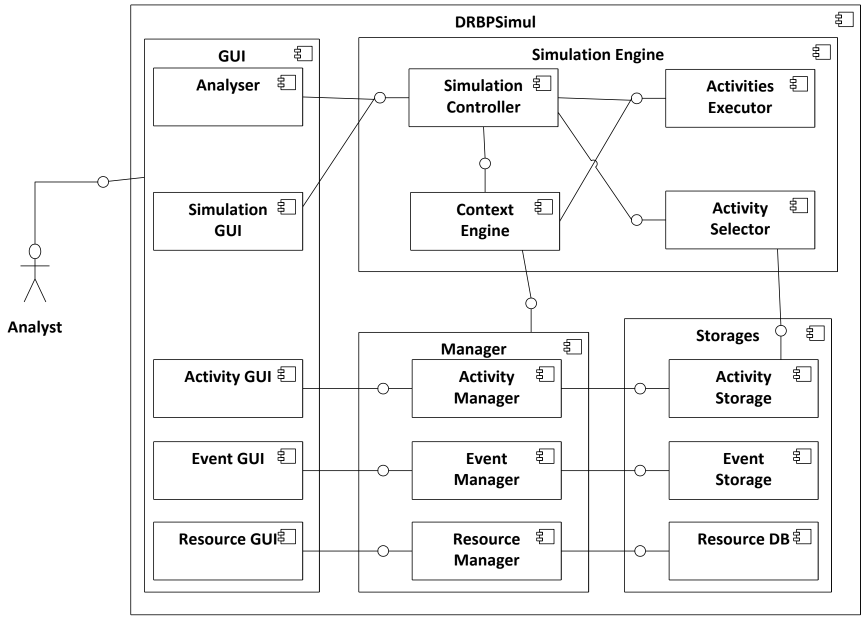

Figure 2. A simulation prototype DRBPSimul architecture

Storages (*Event Storage*, *Activity Storage* and *Resource DB*) are used to store events, activities and resources necessary for simulation execution. The *Activity Storage* saves all possible activities, which can be chosen for the execution. The following information is saved about activities: description of the activity on what should be done during the execution and condition under which the activity is initiated for execution. In *Event Storage*, main *external events*, which initiate execution of a BP, and *internal events* that should be processed, are defined. In the *Resource DB* resources used during the execution of activities are saved (Vasilecas et al., 2015).

Before the simulation, Analyst should load activities, events and resources into Storages and allocate resources to activities. When the loading is over, an Analyst can start the simulation process. The simulation process is managed through Simulation GUI. E.g., the simulation process can be started, stopped, continued and modified. The simulation process is executed and managed by *Simulation Controller*, which gets commands from *Simulation GUI*. When *Simulation Controller* gets a message to start a simulation process, it sends message to *Activity Selector* to choose an activity according to the context, which is defined in *Context Engine*, and detected Events. When activity is selected, it is passed to *Activity Executor* through *Simulation Manager* to execute it.

A pseudo code implemented of activity selection during the DBP simulation is presented as follows:

Algorithm 1: Activity Selection

```
begin
  AvailableActivities ← FilterEnabledActivities(AllActivities)
  ApplicableActivities ← {}
  ExecutionQueue ← ∅
  for each activity in AvailableActivities
    ConditionResult ← Evaluate(GetCondition(activity))
    If ConditionResult = true then
        ApplicableActivities ← ApplicableActivities ∪ {activity}
  Collisions ← IdentifyCollisions(ApplicableActivities)
  If Collisions ≠ ∅ then
      ExecutionQueue ← SolveCollisions(ApplicableActivities)
  Else
      ExecutionQueue ← ApplicableActivities
  Return ExecutionQueue
```

To demonstrate a DBP simulation an ordering process was chosen. The static structure using BPMN of an ordering process is presented in Fig. 3 and a simulation example of an ordering process using the developed prototype is presented in Fig. 4.

In the Fig. 4, the first column on the left presents the list of events, which were detected and processed during the overall simulation. The second column on the left presents the graph of the simulated process instance. The activities presented in yellow are those that were executed in this process instance simulation; those in grey are activities, which are not executed in this process instance simulation but were executed earlier in previous process instances; those in green are activities that were executed in previous step.

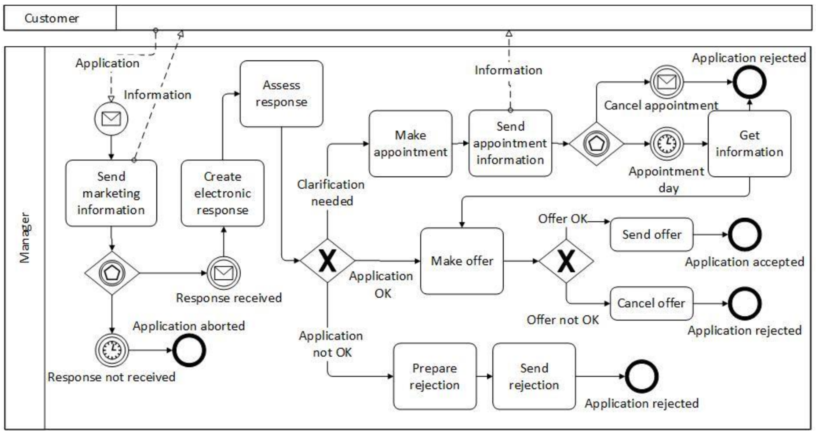

Figure 3. An ordering process model

As can be seen from Fig. 4, the presentation of a graph depends on the executed activities. If all simulated instances have the same execution path, 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 Fig. 4.

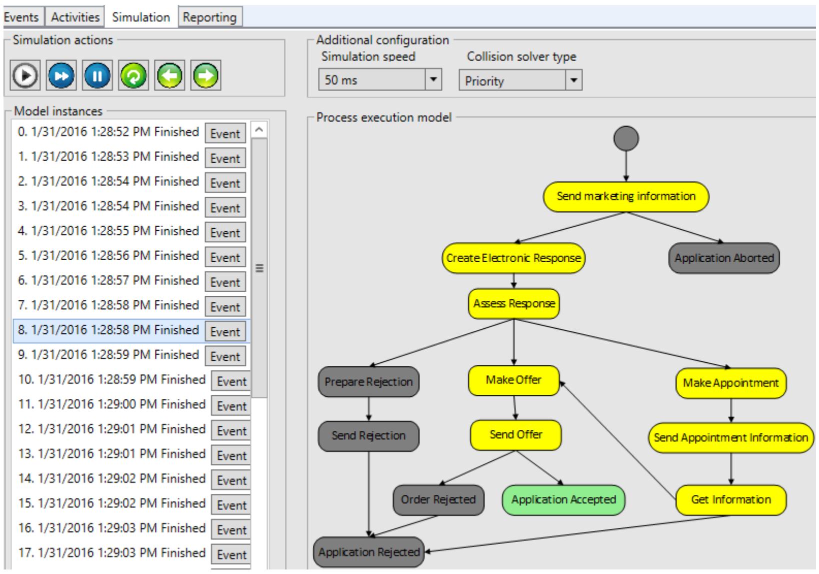

Figure 4. A simulation example of an ordering process

It is possible to change rules and context during the simulation. Rules could be changed through the Activities tab. After changing a rule, we can continue a simulation of a BP instance to see how those changes affect the process instance.

5. Discussion

5.1. Comparison criteria

According to the related works, the criteria for comparison of DBP approaches and tools are derived. Those criteria are as follows:

1. BP model – this criterion is divided as follows:
    1.1. Is initial model used at the beginning of the simulation?
    1.2. What type of dynamicity is implemented: 1) decision points, 2) variation points, 3) ECA rule model, or 4) choosing of activity according to rules and context?
2. Resource model – this criterion is divided as follows:
    2.1. Is it possible to define a resource model?
    2.2. What type of resources could be defined: utilized and employed?
3. Environmental model – is known as a context model or external context model:
    3.1. Is it possible to define an external context?
    3.2. How is external context defined?
4. Business rules – this criterion is divided as follows:
    4.1. Is it possible to integrate business rules into BP?
    4.2. Is it possible to manage (add, change and delete) rules?
    4.3. Is it possible to compare defined business rules?
    4.4. Is it possible analyse rules after and/or during their execution?
5. Historical data – this criterion is divided as follows:
    5.1. Is it possible to use historical data?
    5.2. Is it possible to analyse historical data?
6. Runtime – is it possible to make changes of rules, context and BP activities at BP runtime?

5.2. Comparison of the BP simulation tools

The results of the comparison of rule-based BP modelling and simulation tools, which are widely used for BP modelling and simulation, according to the defined criteria are presented in Table 1. Tools used for the comparison are as follows:

- **IBM Websphere Business Modeler Advanced**[3] (v7.0 2014) allows simulation of DBP by choosing appropriate execution branch. However, it is not suitable for changing rules during the suspension of the BP instance. It is possible to apply changes at the next BP instance.

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

- **Simprocess**[4] (v2015) integrates Process mapping, hierarchical event-driven simulation, and activity-based costing (ABC).
- **Simul8**[5] allows defining of dynamic parameters, which can be changed according to the defined rules.
- **AccuProcess Modeler**[6] allows business people to document, simulate and improve their BP. However, it has no DBP simulation functionality.
- **ARIS 9.7** (ARIS, 2016) incorporates ARIS Business Rules Designer, full support of BPMN 2.0 for BP modelling, and static BP simulation into this version of the software.

Table 1. Comparison of rule-based BP simulation tools
("+" – has this feature, "+/-" – has this feature partially, "-" – has no this feature)

Comparison criteria	**IBM Websphere**	**Simprocess**	**Simul8**	**Accu Process**	**ARIS 9.7**	**Our developed simulation tool**
1. BP model						
1.1. Initial BP model	+	+	+	+	+	+/-
1.2. Type of dynamicity	1	1	1	1	2**	4
2. Resource model						
2.1. Could be defined	+	+	+	+	+	+
2.2. Type of resources*	+	+	-	-	-	+
3. External context						
3.1. Could be defined	-	-	-	-	-	+
4. Business rules (BR)						
4.1. BR integration into BP	+	+	+	+	+	+
4.2. BR management	+	+	+	+	+	+
4.3. Comparison of BR	+	-	-	-	+/-	+
4.4. Analysis of BR	+	-	-	-	-	+
5. Historical data						
5.1. Usage of historical data	+	+	+	-	+	+
5.2. Analysis of historical data	+/-	+/-	+/-	-	+	+
6. Runtime						
6.1. Changes of BR	-	-	-	-	-	+
6.2. Changes of context/resources	-	-	-	-	-	+
6.3. Changes of activities	-	-	+/-	+	+/-	+

* + is when not only cost and time attributes for the resource can be defined, but other attributes, like volume, colour, etc., also.
** Dynamicity is achieved by using Ad-Hoc process elements

[4] *http://simprocess.com*

[5] *http://www.simul8.com*

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

From Table 1 can be determined, that all analysed five tools need initial BP model (1). This limits the dynamicity of BP. All BP modelling and simulation tools allow allocation of resources to activities (2). In half of the analysed tools (IBM Websphere, Simprocess and our developed simulation tool) it is possible define not only cost and time attributes for the resource, but also other attributes, such as volume, colour, etc. It is not possible to define external context (3) in all five analysed tools. However, all tools supports feature of rules modelling within BP; neither of them supports analysis of rules at BP runtime (4). Uses of historical data (5) is supported in almost all analysed tools (except AssuProcess). As can be seen form Table 1, existing tools are well developed to model, analyse and simulate static BP (6). However, those tools are not suitable for simulation of DBP.

6. Conclusions

This analysis of the approaches on a dynamic business process (DBP) shows that there are a number of approaches for DBP modelling. However, all existing approaches are limited in their ability to model DBP and have different levels of support for DBP modelling. Therefore, the levels of dynamicity were analysed and a number of approaches were described according to those levels of dynamicity. The analysis shows that the majority of approaches falls into the middle level of dynamicity (e.g. using process variability or transforming a BP into a set of ECA rules). Moreover, not all approaches are implemented as an execution or simulation tool.

In this paper, we proposed a new approach for rule- and context-based DBP modelling and simulation. The main idea of the approach is that during the simulation of a BP its activities are selected according to the context and rules. Moreover, during the simulation of the same BP instance, a rule set and a context can be changed and those changes can be observed. Consequently, the DBP has no strict activities and their sequence.

According to the proposed approach, an implementation architecture was developed and implemented into a DRBPSimul prototype. An experiment of simulation was carried out and it shows the proposed approach suitability for modelling and simulating DBP.

Finally, the developed tool was compared with five widely used BP modelling and simulation tools (IBM Websphere, Simprocess, Simul8, Accu Process and ARIS 9.7). And it was determined that all five analysed tools need initial BP model, that limits the dynamicity of a BP. Half of the analysed tools (IBM Websphere, Simprocess and our developed simulation tool) are sophisticated enough to define different attributes for the resource. However, it is not possible to define external context in all five analysed tools. All tools allows rules modelling within BP; however, neither of them supports analysis of rules at BP runtime. Summarising all those results, existing tools are well developed to model, analyse and simulate static BP. However, those tools are not suitable for simulation of DBP.

References

ARIS. (2016). Business Process Simulation. *ARIS Community*. http://www.ariscommunity.com/business-process-simulation

Azab, A., AlGeddawy, T. (2012). Simulation methods for changeable manufacturing. *Procedia CIRP* 3, 179-184.

Boukhebouze, M., Amghar, Y., Benharkat, A. N., 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., van Sinderen, M., Zarghami, A. (2013). Achieving flexible process interoperability in the homecare domain through aspect-oriented service composition. *In: van Sinderen, M., et al. (eds.): Proc. of the IWEI 2013*. LNBIP 144, Springer Berlin Heidelberg, pp. 50-64.

Hermosillo, G., Seinturier, L., Duchien, L. (2010). Creating context-adaptive business processes. *In: Maglio, P.P., et al. (eds.): Proc. of the ICSOC 2010*. LNCS 6470, Springer Berlin Heidelberg, pp. 228-242.

Yao, W., Basu, S., Li, J., Stephenson, B. (2012). Modeling and configuration of process variants for on-boarding customers to IT outsourcing. *In: Proc. of the 9th International IEEE Conference on Services Computing (SCC 2012)*. IEEE, New York, pp. 415-422.

Kalibatiene, D., Vasilecas, O., Rusinaite, T. (2015). Implementing a rule-based dynamic business process modelling and simulation. In: *Proc. of the Electrical, Electronic and Information Sciences (eStream'2015)*. IEEE, New York, pp. 1-4.

Kellner, M.I., Madachy, R.J., Raffo, D.M. (1999). Software process simulation modeling: why? what? how?. *Journal of Systems and Software* 46(2), 91-105.

La Rosa, M., van der Aalst, W.M., Dumas, M., Milani, F.P. (2013). *Business process variability modeling: A survey*.

Mejia Bernal, J.F., Falcarin, P., Morisio, M., Dai, J. (2010). Dynamic context-aware business process: a rule-based approach supported by pattern identification. *In: Proc. of the 2010 ACM Symposium on Applied Computing*. ACM, pp. 470-474.

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

O'Donnell, J. (2013). SimModel: A domain data model for whole building energy simulation. *In: Proc. of the SimBuild 2011, Sydney, Australia*.

Rozinat, A., Mans, R.S., Song, M., van der Aalst, W.M. (2009). Discovering simulation models. *Information Systems* 34(3), 305-327.

Rus, I., Holger, N., Jürgen, M. (2014). A systematic methodology for developing discrete event simulation models of software development processes. *arXiv preprint arXiv:1403.3559*.

Rusinaite, T., Kalibatiene, D., Vasilecas, O. (2015). Requirements of dynamic business processes - a survey. *In: Proc. of the AIEEE 2015*. IEEE, New York, pp. 1-4.

Rusinaite, T., Vasilecas, O., Kalibatiene, D. (2016). A Systematic Literature Review on Dynamic Business Processes. *Baltic J. Modern Computing* 4(3), 420-427.

Rusinaite, T., Vasilecas, O. (2015). Dynamic business process model implementation solutions. *In: Proc. of eStream'2015*. IEEE, New York, pp. 1-4.

van der Aalst, W.M. (2010). Business process simulation revisited. *In: Barjis, J. (ed.): Proc. of the EOMAS 2010*. LNBIP 63, Springer Berlin Heidelberg, pp. 1-14.

van der Aalst, W.M. (2013). Business process management: A comprehensive survey. *In: Proc of the ISRN Software Engineering.*

van Eijndhoven, T., Iacob, M.E., Ponisio, M.L. (2008). Achieving business process flexibility with business rules. *In: Proc. of the EDOC'08*. IEEE, New York, pp. 95-104.

Vasilecas, O., Kalibatiene, D., Rima, A., Birzniece, I., Rudzajs, P. (2015). A Resource Model for the Rule-Based Dynamic Business Process Modelling and Simulation. *In: Proc. of the ESM'2015*. Eurosis Publication, pp. 36-41.

Received April 28, 2016, accepted April 30, 2016