RBPSim: A Resource-aware Extension of BPSim Using Workflow Resource Patterns

Nehal Afifi, Ahmed Awad and Hisham M. Abdelsalam

Faculty of Computers and Information, Cairo University, Giza, Egypt
n.afifi@grad.fci-cu.edu.eg,{a.gaafar,h.abdelsalam}@fci-cu.edu.eg

Abstract. One of the main limitations affecting business process simulation approaches is the incorrect modeling of human resources. The BPSim standard is acknowledged as a first step towards streamlining the experience of business process simulation and providing a tool independent exchange format for so-called simulation scenarios. Unfortunately, with respect to the human resource perspective, the standardization effort did not advance behind modeling resources as quantities required for the different process elements. Workflow resource patterns outline resources' distribution and utilization. This paper is taking a first step towards combining BPSim standard with the well-known workflow resource patterns through RBPSim: a resource-aware extension of BPSim standard.

Keywords: BPSim, workflow resource patterns, business process simulation

1 Motivation

Analysis of business process models has for a long time focused on verification, e.g. soundness [1], and validation, e.g. compliance checking [4]. Simulation as an important analysis approach for business processes has received very little attention from researchers [9]. Simulation attempts to predict how the real-world processes will operate through various "What-if" scenarios [3, 5].

For simulating business processes, we need to model at least three perspectives: control flow, data and resources [2]. One of the main issues affecting the current business process simulation approaches is modeling human resource in a naïve manner [2]. Resources are referred to either by quantities within a role or explicitly. To enrich simulation models with appropriate specifications of resource requirements, we need a standard definition of a resource model to be used in any Business Process Simulation(BPS) experiment [7].

The Business Process Simulation (BPSim) standard version 2.0 [16], developed by WfMC, allows business process models specified in either BPMN [10] or XPDL [15] to be augmented with simulation-specific parameters such as task durations, branching probabilities, case arrival rates, etc. The BPSim metamodel is not fully elaborated regarding the resource perspective. Oversimplified resource modeling and omitting workflow resource patters can cause unsuccessful simulations [9, 9, 12]. Resources require richer representation with respect to work

N. Herzberg, C. Hochreiner, O. Kopp, J. Lenhard (Eds.): 10th ZEUS Workshop, ZEUS 2018, Dresden, Germany, 8-9 February 2018, published at http://ceur-ws.org/Vol-2072

preference, speed and realistic allocation plan and working schedule regard to work items. BPSim limitations in modeling resources have been discussed in [6,8]. However, the standard is extensible as it defines a meta-model for its elements and extension points.

In this paper, we take a first step towards extending BPSim to enrich resources specification. We use the well-known workflow resource patterns [12] as the means to express resource's selection and allocation strategies during a simulation scenario. The rest of the paper is organized as the following: section 2 discusses extending BPSim standard with workflow resource patterns, an evaluation example is presented in Section 3. Section 4 concludes the paper with an outlook on future work.

2 Extending BPSim with Workflow Resource Patterns

In this section we discuss the level of support offered by BPSim regarding human resources. We present our extension of BPSim metamodel for more expressive resources representation in simulation scenarios. This extension is based on the well-known workflow resource patterns [12,13].

2.1 Revisiting Workflow Resource Patterns

Workflow resource patterns specify resources representation, selection and utilization within the process model and are divided into seven groups [12,13]. **1. Creation patterns** are concerned with *which resources are eligible?*. That is out of all available resources R, a Creation pattern cp is responsible for finding set $R_{cp} \subseteq R$ which represents the candidate resources where any of them can execute the respective task t. R_{cp} can be either specified by properties that each resource $r \in R_{cp}$ must possess to be able to execute t or it can be specified by explicitly enumerating its members; **2. Push patterns** are concerned with *How to pick one of the eligible resources?*. Push patterns are more on the execution or simulation time assignment of a work item wi to a resource $r \in R_{cp}$ where wi is the instance of task t within a specific process instance. So, the enforcement of a Push pattern should result in at most one specific resource being assigned to the work item whereas enforcement of a Creation pattern results in a set of candidate resources R_{cp}. Note that R_{cp} might be empty in case of none of the available resources possesses sufficient capabilities to perform t; **3. Pull patterns** the difference between Push and Pull patterns is that transitions in Push patterns are stated by the system while in Pull patterns resources may have the ability to initiate transitions, reorder their working queue and select the next work item to be executed; **4. Detour patterns** refer to interruptions to work items either by the system or by resources executing them; **5. Auto-Start patterns** refer to the triggering of work items by specific events either through creation or allocation; **6. Visibility patterns** determining work item visibility for a resource and **7. Multiple Resource patterns** are concerned with tasks that require more than one resource working on it concurrently.

Following a divide-and-conquer approach, in this paper, we are concerned with the first five groups, namely Creation, Push, Pull, Detour and Auto-Start patterns. We argue that building simulation models where those patterns can be employed is a first step towards getting more accurate simulation results. Multiple Resource patterns are currently out of scope and subject for future research. Visibility patterns are purely related to process enactment and building of process execution engines, like implementation of work item lists. Thus, they are not considered for process simulation.

2.2 BPSim Resources Representation Limitations

To accomplish an effective business process simulation experiment, process elements, e.g. tasks, involved in the experiment should include the following [6,14]: **1. Required resources:** to execute a task, one or more resources should be available to handle the task based on task-specific requirements; **2. Execution duration:** resources execution duration for each task is not constant and should follow a probabilistic distribution; **3. Resource share-ability:** resources are not dedicating all their time to one task and may divide time simultaneously between different tasks; **4. Resource availability:** resources may be unavailable to perform tasks which requires a value attached to the resources indicating availability; **5. Context switching overhead**: resources may require time intervals between different tasks execution so another timing interval with a probabilistic distribution should be included to specify time required between tasks execution and **6. Work item selection:** resource's working queue is necessary to organize work items and specify how resource will select the next work item (FIFO, LIFO or based on priority). Among the above-mentioned requirements for a simulation experiment, BPSim covers the following [16]:

1. ElementParameter indicates the reference to a process element and extended with a number of parameters: (a) time, (b) control that defines control flow of BPMN element, (c) cost, (d) priority contains "Interruptible Attribute" specifies if the execution is interruptible and "Priority Attribute" defines the resource allocation order based on element priority, (e) property (f) expressions are added functions such as *getResource* to select a collection of available resources, *getResourceByRole* to select a collection of available resources based on role and *Resource* to select an alternative list of available resources and (g) resource parameter. *2. ResourceParameter* specifies the resource's availability, quantity, selection based on defined role list or a number referring to a specific resourceID. Availability, quantity and role properties are only applicable for an individual resource element leading to the inability of selecting a specific resources based on other criteria. *3. TimeParameter and ControlParameter* are neither related to a resource nor resourcesRole elements although in some cases it might be needed, TimeParameter could be added to resource and resourceRole BPMN elements. *4. CostParameter* could be applied to the resource element (but not the resourceRole element) specifying the cost of resources either by fixed cost attribute based on resource usage or unit cost attribute based on a time unit [6]. *5. PriorityParameter and PropertyParameter* are not applicable for resource and

resourceRole elements. Condition attribute in ControlParameter is just a Boolean and only one condition could be applied to a BPMN element, this parameter should be modified to handle an expression parameters for filtering the resource required for task execution.

Several tools support business process simulation. They are either scientific prototypes, e.g., BIMP [1] and Desmo-J [11] or commercial tools, e.g., Bizagi [2], BonitaSoft [3], Visual Paradigm [4] and Trisotech Modeler [5]. Some of these tools are BPSim compliant. Freitas et al. [6], have assessed these tools with respect to their BPS capabilities. Concerning the resources perspective, all the selected tools support setting the number of available resources to execute work items, only Bizagi and Trisotech support allocation plan of resources, none of the tools supports unavailability of resources and all tools except Visual Paradigm define the resources working schedule and the resources usage cost. All tools define task cost while only Trisotech supports defining task execution priority and ability of interruption while running.

All mentioned tools that go beyond the resources support in BPSim still are not fully covering the resource's perspective. In Section 2.3 we discuss BPSim extended metamodel and provide a systematic way for more expressive resource representation in simulation scenario.

2.3 Realizing Workflow Resources Patterns in BPSim

In this section, we demonstrate our extension of BPSim metamodel showing how to address workflow resource patterns discussed in Section 2.1 based on the BPSim standard. Implementaion of the extended metamodel is left for future work. The extended metamodel is shown in Fig. 1. The newly added classes are highlighted with gray.

BPSim introduces the scenario entity containing all parameters needed to run a simulation. Each scenario represents one *what-if* case. Thus, it is defined for each business process element, e.g. a task, several parameters including, duration, resources, time unit etc. **Resources:** entity is the parent for both human and non-human resources, in this paper we are concerned with **HumanResources** to enumerate resources participating in a scenario and describe them with attributes. Those resources might be referred to later directly or indirectly by a resourceParameter. **ResourceQueue:** is used to handle resource's work items waiting to be executed. Work items in the resource's queue could be sorted based on preference of specific work items, FIFO, priority, LIFO. **Role:** A human resource is a member of role which generalizes over **OrganizationalGroup** and **Position** that has (**Privileges**). A human resource may have execution history of work items **History** and a **ShiftCalendar** indicating the availability. Shift extends BPSim CalendarParameter [16].

[1] http://bimp.cs.ut.ee
[2] https://www.bizagi.com
[3] https://www.bonitasoft.com
[4] https://www.visual-paradigm.com
[5] https://www.trisotech.com/release-notes/bpmn-modeler

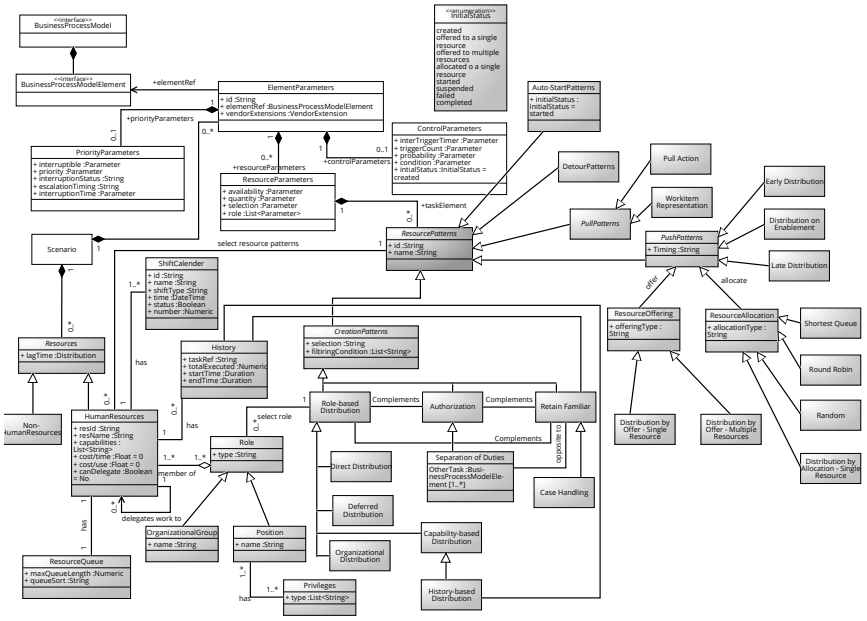

Fig. 1. New extended BPSim MetaModel

ResourcePattern: extends ResourceParameter defined in BPSim and specifies the required resource patterns as discussed in subsection 2.1 that should be applied on ElementParameter. **CreationPatterns:** specifies the requirements to execute a work item. **PushPatterns** include a timing parameter for work item distribution, work item offering to resources based on distribution by offer to single or multiple resources and work item allocation to a single resource based on shortest queue, round robin or random distribution. **PullPatterns** defines the awareness of resources with offered work items required for execution either from direct allocation or a shared work list. It includes two actions: 1. the pull action either from allocation or offering and 2. the sequence of work item representation in the resource waiting queue. **Detour Pattern:** defines the interruptions occurred to work items including escalation, redo, skip, etc. **Auto-Start Patterns** specifies the initial status of work item to *Started* so work item would start immediately. Table 1 identifies the defined classes and parameters in the extended metamodel and their equivalent resource patterns.

ElementParameter was modified in the following way: **1. ControlParameter modification.** based on work item life cycle [13], a state parameter indicating the work item transition from creation to offering then allocation. Work item is then started and moves either to completed, failed or suspended and then resumed. **2. PriorityParameter modification.** a new string attribute indicating the interruption status was added, also the interruption time attribute based on TimingParameter;

Table 1. Extended classes and related resources patterns

Class	**Parameter**	**Pattern Category**	**Pattern**
HumanResource	resName	Creation	Direct distribution
	capabilities	Creation	Capability-based distribution
	canDelegate	Detour	Delegation
ResourceQueue	queueSort	Pull	System-determined work queue content, Resource-determined work queue content
PullAction		Pull	Resource-initiated allocation, Resource-initiated execution- allocated work item, Resource-initiated execution- offered work item
Workitem Representation		Pull	Selection autonomy
Role	type	Creation	Role-based distribution
OrganizationalGroup	name	Creation	Organizational distribution
Privileges	type	Creation	Authorization
History	totalExecuted	Creation	Retain familiar
History	taskRef	Creation	History-based distribution Separation of duties
elementParameter	id (process)	Creation	Case handling
PushPatterns	Timing	Push	Early distribution, Distribution on enablement, Late distribution
ResourceOffering	offeringType	Push	Distribution by offer- single resource, Distribution by offer- multiple resources
ResourceAllocation	allocationType	Push	Distribution by allocation- single resource, Random allocation, Round robin allocation, Shortest queue
Auto-StartPatterns	initialStatus= "Started"	Auto-Start	Commencement on Creation

3 Example

The example explains a simple business process for "Car Maintenance", see Fig. 2. When cars arrive, an administration employee receives and records car information, the selection, offering and allocation of resource are following these patterns: 1. **Creation (selection):** *Role-based Distribution*, 2. **Push:** distribution timing based on *Distribution on Enablement*; offering based on *Distribution by Offer-Multiple Resources* and allocation based on *Round Robin*. The car is then sent to the mechanical department where an engineer is selected based on Role-based Distribution and Capability-based Distribution patterns with experience of 3 years and possession of a certificate. Finally, an accountant receives and records payments for the work done. The accountant pulls work items from his working queue based on work item priority. Listing .1 is an excerpt of the XML for the simulation scenario parameters based on the extended metamodel from Section 2.3

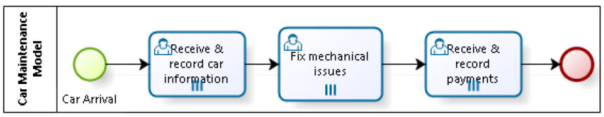

Fig. 2. Simple car maintenance process in BPMN

that realizes the process and patterns discussed above. ElementParameter of "Receive & record car information"task is defined in lines 4 to 13 including the new updates in ControlParameters (initialStatus) and PriorityParameters (escalation-Timing). ResourcePatterns are defined in lines 8 to 12 specifying creation pattern (role-based distribution), push patterns (resource offering, resource allocation and distribution timing). Parameters for the other tasks are defined in lines 13 to 31. Human resources participating in the scenario are defined in lines 32 to 49 specifying resource id, name, capabilities, lagTime, role, organizationalGroup, position, and resource queue properties.

```
1  <bpsim:Scenario id="S1" name="Scenario1:Car Maintenance Model" ...>
2    <bpsim:ScenarioParameters baseTimeUnit="min"/>
3    ...
4    <bpsim:ElementParameters elementRef="Receives & records car information">
5      ...
6      <bpsim:ControlParameters intialStatus="created"/>
7      <bpsim:PriorityParameters interruptible="True" escalationTiming="PT15M"/>
8      <bpsim:ResourcePatterns>
9        <bpsim:Role-basedDistribution roleName="Administration Employee" requiredResQty="1"/>
10       <bpsim:ResourceOffering Timing="Distribution on Enablement" OfferingType="Distribution by Offer-
             Multiple Resources"/>
11         <bpsim:ResourceAllocation AllocationType="round robin"/>
12     </bpsim:ResourcePattern>
13   </bpsim:ElementParameters>
14   <bpsim:ElementParameters elementRef="Fix mechanical issues ">
15       <bpsim:ResourcePattern>
16       <bpsim:Role-basedDistribution roleName="Maintenance Engineer" requiredResQty="1"/>
17       <bpsim:Capability-basedDistribution>
18         <bpsim:Capability experience="3years" certificate="yes"/>
19       </bpsim:Capability-basedDistribution>
20       <bpsim:ResourceOffering Timing="Distribution on Enablement" OfferingType="Distribution by Offer-
             Multiple Resources"/>
21         <bpsim:ResourceAllocation AllocationType="round robin"/>
22       </bpsim:ResourcePattern>
23   </bpsim:ElementParameters>
24   <bpsim:ElementParameters elementRef="Receive & record payments">
25     ...
26     <bpsim:ResourcePatterns>
27       ...
28       <bpsim:PullAction= "Resource-Initiated Execution-Allocated Work Item"/>
29       <bpsim:WorkitemRepresentation= "Resource-Determined Work Queue Content" queueSelection="
             Priority"/>
30     </bpsim:ResourcePatterns>
31   </bpsim:ElementParameters>
32   <bpsim:HumanResources>
33   <bpsim:HumanResource id="Res1" name="Adam" costUnit="$20/h" canDelegate="No">
34         <bpsim:Capabilities experience="3 years" certificate="yes"/>
35         <bpsim:LagTime>
36         <bpsim:UniformDistribution min="3" max="10">
37         </bpsim:LagTime>
38           <bpsim:Role Type="Administration Employee" >
39         <bpsim:OrganizationalGroup name="Administration Department"/>
40             <bpsim:Position name="Reception Employee">
41                 <Privileges/>
42         </bpsim:Position>
43       </bpsim:Role>
44       <bpsim:ShiftCalender/>
45       <bpsim:History/>
46       <bpsim:ResourceQueue maxQueueLength="20" queueSort="Priority"/>
47     </bpsim:HumanResource>
48         ...
49   </bpsim:HumanResources>
50     ...
51 </bpsim:Scenario>
```

Listing .1. RBPSim XML for the example

4 Conclusion and Outlook

In this paper, we have introduced RBPSim, a resource-aware extension for the BPSim metamodel, for better representation of resources within simulation scenarios. The extension is based on workflow resource patterns. RBPSim provides a tool independent exchange format for so-called simulation scenarios including the resources perspective.

In future, we plan to introduce the remaining resource patterns. Moreover, we aim at starting an implementation of the extended BPSim metamodel. The implementation may have two directions. The first is to seek an open source BPS tool that supports BPSim to apply the extended metamodel. The second direction is to implement the extension using a general-purpose simulation tool.

References

1. van der Aalst, W.M.P., van Hee, K.M., ter Hofstede, A.H.M., Sidorova, N., Verbeek, H.M.W., Voorhoeve, M., Wynn, M.T.: Soundness of workflow nets: classification, decidability, and analysis. Formal Asp. Comput. 23(3), 333–363 (2011)
2. van der Aalst, W.M.: Business process simulation revisited. In: Workshop on Enterprise and Organizational Modeling and Simulation. pp. 1–14. Springer (2010)
3. Van der Aalst, W.M., Nakatumba, J., Rozinat, A., Russell, N.: Business process simulation. In: Handbook on BPM 1, pp. 313–338. Springer (2010)
4. Awad, A., Weidlich, M., Weske, M.: Visually specifying compliance rules and explaining their violations for business processes. J. Vis. Lang. Comput. 22(1), 30–55 (2011)
5. Banks, J., Carson, J., Nelson, B., Nicol, D.: Discrete-event system simulation. Prentice Hall, 5th ed. edn. (2010)
6. Freitas, A.P., Pereira, J.L.M.: Process simulation support in bpm tools: The case of bpmn. In: Proceedings of 2100 Projects Association Joint Conferences (2015)
7. Januszczak, J., Hook, G.: Simulation standard for business process management. In: Simulation Conference (WSC), Proceedings of the 2011 Winter. pp. 741–751. IEEE (2011)
8. Laue, R., Müller, C.: The business process simulation standard (bpsim): Chances and limits. In: ECMS. pp. 413–418 (2016)
9. Nakatumba, J., Rozinat, A., Russell, N.: Business process simulation: How to get it right. In: Int. Handbook on Business Process Management. Springer (2009)
10. OMG: Business process model and notation (bpmn version 2.0) (2011)
11. Pufahl, L., Weske, M.: Extensible BPMN process simulator. In: BPM Demo. CEUR Workshop Proceedings, vol. 1920. CEUR-WS.org (2017)
12. Russell, N., van der Aalst, W.M., Ter Hofstede, A.H., Edmond, D.: Workflow resource patterns: Identification, representation and tool support. In: International Conference on Advanced Information Systems Engineering. pp. 216–232. Springer (2005)
13. Russell, N., Ter Hofstede, A.H., Edmond, D., van der Aalst, W.M.: Workflow resource patterns. Tech. rep., BETA Working Paper Series, WP 127, Eindhoven University of Technology, Eindhoven (2004)
14. Waller, A., Clark, M., Enstone, L.: L-sim: Simulating bpmn diagrams with a purpose built engine. In: Simulation Conference. pp. 591–597. IEEE (2006)
15. WfMC: Xml process definition language (xpdl version 2.2) (2012)
16. WfMC: Bpsim - business process simulation specification (2016)