Dynamic Switching of Perspectives on Business Processes

Florian Krenn and Christian Stary

Johannes Kepler University Linz, Department of Business Information Systems-
Communications Engineering, Altenberger Straße 69
4040 Linz, Austria
{Florian.Krenn, Christian.Stary}@ jku.at

Abstract. Process models represent process specifications. They contain workflows that require execution, in order to achieve business objectives and support business operation effectively. With the advent of Subject-oriented and Social Business Process Management, communication and stakeholder interaction have become major perspectives on how to design and implement processes. Since such a perspective does not seem to be very common when executing processes, stakeholders, including organizational developers and IT specialists, can be supported looking at process execution from either perspective, namely from a traditional one, targeting the flow of functions, and from an interactional perspective, focusing on interaction among stakeholders or system components encapsulating behavior. In this paper, we introduce the meta-model and architecture required for a respective dual mode support tool. The workflow execution engine UeberFlow allows checking the completeness of process specifications from either perspective. Consequently, stakeholders can start modeling with a perspective they are familiar with and proceed with the other one by switching dynamically to the alternate mode of modeling and execution.

Keywords: Polymorph execution, SBPM, S-BPM, function flow, interaction flow.

1 Introduction

Although Business Process Management (BPM) is considered highly relevant for sustaining on volatile markets, many organizations are still struggling to understand, and thus fully implement its concepts [18]. Consequently, BPM technologies develop towards interactive stakeholder support through easy-to-capture and model features allowing the generation of executable models [14], [4]. When aiming to understand barriers hindering organization-relevant stakeholders to utilize BPM system capabilities, two relevant and complimentary perspectives when executing process specifications can be taken, namely function-based and interaction-based workflows. Function-based workflow specifications allow focusing on technical steps required for task accomplishment, whereas interaction-based execution puts the exchange of messages or business objects between self-contained behavior entities (role actors or applications) to the center of interest. Figure 1 shows a simple process from both perspectives, in order to illustrate the different points of view. The left side of the figure shows the process from a subject-oriented perspective. The three process participants (Requester, Head of Department, and Cost Center) are depicted including their interaction and information exchange. The other perspective shows the same process from a functional point of view, focusing on the sequence of tasks in this business process.

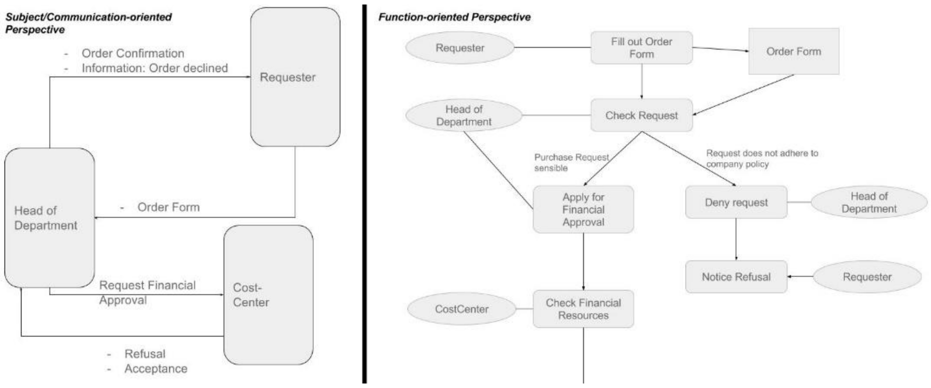
Subject/Communication-oriented Perspective
- Order Confirmation
- Information: Order declined
Requester
Head of Department
- Order Form
Request Financial Approval
Cost-Center
- Refusal
- Acceptance
Function-oriented Perspective
Requester
Fill out Order Form
Order Form
Head of Department
Check Request
Purchase Request sensible
Request does not adhere to company policy
Apply for Financial Approval
Deny request
Head of Department
Notice Refusal
Requester
CostCenter
Check Financial Resources

Fig. 1. One process – two perspectives: interaction and function flow.

The development of such a dual approach to execution has been triggered by Social BPM developments (SBPM) which take into account the social nature of executing and re-designing work processes [2]. Besides embedding Social Media into the BPM lifecycle, such as enriching process models and tagging process elements [12], [9], or annotating social interactions [1], the required interaction among stakeholders for achieving business objectives can be encoded directly into process models [4], [13]. Such an endeavor provides the opportunity to view functional behavior from the social perspective in an integrated way, as stakeholders interact when accomplishing their work tasks and thereby, complete processes. A corresponding workflow engine enables stakeholders (project managers, developers, users) to experience processes live and reflect on interaction and functional behavior when implementing them in an organization.

We have structured the paper as follows: Section 2 discusses various perspectives on processes and process specifications. The results allow arguing for capturing and following functional and interactional angles when looking at process specifications, and finally, executing them. Section 3 details the conceptual meta-model when representing a functional and an interactional viewpoint on process specifications, and the architecture of a system enabling the interactional and functional flow of execution. Section 4 demonstrates how such a tool can be implemented and used in BPM practice. Section 5 closes the paper, wrapping up research objectives and results, and revealing topics of further research.

2 Perspectives on Processes Specifications

In this section, we review selected BPM approaches that refer to or encode perspectives on business processes. It allows arguing for developing or preserving perspectives on business processes as they facilitate understanding behavior patterns of organizations when tasks are accomplished or strategic objectives need to be met.

User roles play a crucial role on how to look at process specifications, as they do not only encapsulate targeted and self-contained behavior sequences, but also refer to qualification profiles of human actors or to requirements applications need to meet. Recently, Trkman et al. identified roles as essential context of activities [15]. The acquisition and representation method the authors propose should increase the understanding of the execution order of process steps, and thus integration dependencies for effective execution support. Thereby, user stories are put into relation to business process model activity elements. A user story is a brief statement involving stakeholder roles and activities in the form of *I as a <user role> perform <function>*. In principle each functional activity of a process can be contextualized by such a statement. However, a set of such statements does do not lead to complete or cohesive functional sequences. In the study, the standard notation for modeling business processes, the Business Process Model and Notation (BPMN) (www.bpmn.org) has been used to represent processes. The authors considered the activity element as 'the most important element since it is associated with a user story' [15]. An activity can be either an atomic or non-atomic (compound) unit of work that an organization performs in its business processes. It provides insights into the "What" an organization is doing. This prominent denotation of an activity indicates the primary perspective on processes, namely a function-oriented one.

However, 'its surrounding elements such as swim lanes, XOR gateways, and flow arrows' were considered 'important for understanding the dependencies' (*ibid*.). Thereby, swim lanes or pools serve as a graphical container for partitioning a set of activities from other activities. A pool represents a participant in a collaboration (e.g. a department), whereas a lane is a partition that is used to organize and categorize activities within the pool (e.g. a specific organizational role within a department). In this way, swim lanes provide insight into "Who" performs a specific activity, and thus, a constitutive element of taking an interactional perspective.

A BPMN business process model is a set of activities that represent the steps required to achieve a business objective. Its name should refer to the "Why" of the represented business operation. The use of these BPMN elements facilitates the integration of swim lane (Who) and activity (What), according to the coupling of the user story items 'user role' and 'function'. Hence, the functional and role perspective are complimentary when developing a contextual model of business processes.

Since the researchers aimed at intelligibility of models for stakeholders, a level of abstraction was selected that allowed for the analysis of the organization's business processes rather than for executing the models automatically. Nevertheless, this case shows how developers take a function-centered perspective of model representations when working with stakeholders. This finding is in line with the results presented in [10]. When working with students, the researchers could identify the initial approach novices are likely to take, namely to represent processes are flowcharts. 72% of the novices conveyed process information in a diagram detailing the steps as boxes, and giving their order by connecting them with arrows. These empirical findings allow concluding that activity-centered descriptions, as already provided by ARIS [11] are constitutive elements of business process specifications.

Mattos et al. investigated factors influencing activities in each instance of a business process [7]. They referred to several context elements of business operations: environment, people, technologies organization of work, external factors. The authors were interested in difficulties caused by these factors when business processes are executed. They intended to identify and group contextual elements, enabling the description of a situation of an activity performed in a business process. Such a specification could support the adaptation of a running process, in order to achieve better results for the organization. Rather than giving a formal model for that, Pinggera et al. refer to the situation when specifications come into being [8]. The perception of a task difficulty by a modeler seems to correlate with the probability of a modeler expecting difficulties in the course of modeling, and the necessity to rework a model. However, the time spent for creating a model seems to be largely independent of the perceived complexity of the task. Overall, the complexity of a modeling task and the way a modeler perceives it seem to be essential influence factors. For both parameters, focusing on a certain modeling perspective could influence the effectiveness or efficiency of modeling. Hence, the complexity of a modeling task could be reduced for a modeler by switching from a fully loaded functional flow model to a communication-oriented flow of control, simplifying the overall behavior specification. Hereby, the modeler could find a particular viewpoint on modeling more convenient than others, and thus more effective to handle a certain case.

The diversification of process specifications is already supported by several concepts, as they correspond to perspectives on specifications [3]. A core concept is called Instantaneously Available Organization. It promotes templates for detailing process specifications. Enriching traditional reference modeling approaches it also provides role descriptors, which shed a different light on specifications. Another concept also corresponds to a perspective on processes. It is termed Organizational Aspect and concern orthogonal characteristics an organization can exhibit, e.g., making decisions transparent to all concerned stakeholders. In this way, strategic objectives and cultural issues can be represented and assured through specifications.

The concept of Orchestrated Business Object concerns the execution, as it refers to the software implementation of a business entity and its associated functionalities including operational data. These pieces of software implement business entities inside some specific business ontology, as they expose functionality and data to the software implementing these concepts. In accordance to subject orientation [4] each Orchestrated Business Object (subject/service) is a black-box, and should be interchangeable with other services implementing the same business entity. This concept enables to implement a process utilizing different software applications that are orchestrated in terms of services as specified in the process model. Such an abstraction decouples modeling the organization of work from technical implementation capacities and complement functional specifications with the interactional perspective for process execution. Not only the resources for implementing processes become interchangeable, but also the exchange process itself does not cause any disruption in the organizational behavior. Hence, coupling the functional perspective on business processes with an interactional one is of twofold benefit: It reduces the complexity of the modeling process and allows high organizational agility when executing the specifications.

3 Meta-Modeling a Dual-Perspective Engine

In this section, we introduce a meta-model called "UeberFlow Lang" which facilitates the capturing of the functional and interactional perspective on executable process specifications.

3.1 UeberFlow Lang – The Meta-Model

Generally, a workflow model comprises of a set of actions or tasks, which are ordered in a certain sequence and performed by workflow participants having certain roles. Workflow specifications in UeberFlow Lang incorporate these workflow elements using three basic building blocks. UeberFlow Lang Workflow specifications, as illustrated in Figure 2, comprise of *WorkflowUnits*, *WorkflowSteps* and *WorkflowFunctions*. Each component defined in the meta-model can be understood as an actor according to the actor model adhering to the defined mechanisms and regulations [17].

WorkflowSpecification: The WorkflowSpecification represents an entirely executable model of the workflow. It acts as container for the WorkflowUnits and stores the relevant meta-data like creation date or access permissions.

WorkflowUnits: WorkflowUnits group process steps according to the responsible role similar to lanes in BPMN or subjects in subject-oriented workflow specifications. For each role participating in the specified workflow a WorkflowUnit is created which contains and supervises all WorkflowSteps the corresponding role is responsible for. Additionally, WorkflowUnit functions as a data space for the underlying WorkflowSteps and WorkflowFunctions. In the course of execution all data accessible by the associated role are made available to the WorkflowSteps via the WorkflowUnit.

WorkflowSteps: WorkflowSteps represent the activities a workflow comprises. The actual execution logic of a WorkflowStep, its prerequisites and results, are solely defined by its WorkflowFunctions. Each WorkflowStep contains a sequence of WorkflowFunctions which are executed sequentially, when the corresponding step is triggered. WorkflowFunctions can be assigned to a WorkflowStep without any limitations concerning order or quantity. The execution of a step is complete, once all its WorkflowFunctions have been executed successfully.

WorkflowFunctions: WorkflowFunctions are the most fine-grained units of execution in the UeberFlow Lang meta-model, and the only truly active component from the workflow's perspective. Each WorkflowFunction represents an atomic action of workflow execution. In order to define the workflow execution logic on a very fine-grained level, for each WorkflowFunction an optional condition can be specified which limits the set of situations the WorkflowFunction is triggered based on current instance data.

Since WorkflowFunctions encapsulates all actions of a workflow specification, different types of WorkflowFunctions are needed for runtime purposes. In the herein presented basic version of UeberFlow Lang six WorkflowFunction-types are defined:

- *RequireFunction.* The RequireFunction allows specifying a set of values required for the execution of subsequent functions defined in the WorkflowStep. These values can either represent an event triggered during the workflow execution or a set of data. The execution of the process step stops until the required values are available for the process unit. The RequireFunction has an optional convert expression, which allows modifying the data before it is made available in the context of the WorkflowStep. Since the RequireFunction completely abstracts from the source of the data it is agnostic to whether the incoming (or already available) data was provided via a message or by the previous step.

- *ProvideFunction.* Upon execution, the ProvideFunction sends a set of values to any WorkflowUnit defined in the workflow. Analogous to the RequireFunction, these provided values can either be a set of data (e.g., completed order form) or an event. Thus, the ProvideFunction can be used in combination with the RequireFunction. It implements asynchronous messaging/data exchange between WorkflowUnits. In this way, data become available for subsequent steps in the same unit.

- *ProceedFunction.* The ProceedFunction triggers the execution of another WorkflowStep. The execution of the current step has not to be complete, in order to trigger the next step, i.e., other functions can be executed after the ProceedFunction has been executed. AND-, OR- and, XOR-Gateways can be implemented by specifying multiple sequential ProceedFunctions and corresponding conditions within one WorkflowStep. Besides simply triggering the execution of a subsequent step, the ProceedFunction offers an alternative to the ProvideFunction. It is also possible to directly pass data to the triggered step. For example, the result of a calculation performed by step A can be passed on the subsequent step B without adding it to the data context of the WorkflowUnit.

- *JoinFunction.* JoinFunctions enable synchronizing two or more parallel execution flows by halting execution of the containing ProcessStep until all of the defined previous steps have been executed. It is also possible to define a subset of steps required in order to realize a partial join according to the workflow patterns as described by [16]. The JoinFunction does not distinguish between the synchronization of paths within a single WorkflowUnit or synchronizing parallel paths of different WorkflowUnits.

- *RequestInputFunction.* The RequestInputFunction is used to define required user input. Based on a specified input form, the current user (or users) associated with the role of the WorkflowUnit is requested to provide input. The execution of the WorkflowStep is halted until the required input is provided.

- *CallFunction.* The CallFunction allows extending the workflow capabilities by using external services. Such an extension can be achieved by defining a code snippet, which is interpreted at run-time, once the WorkflowFunction is executed.

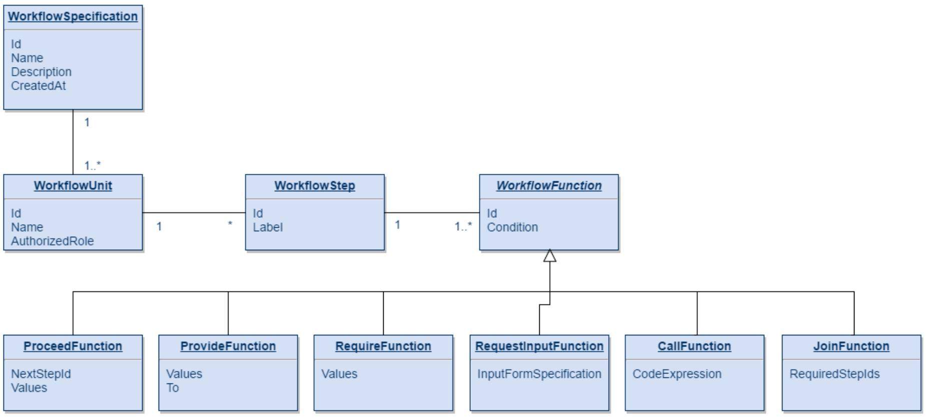

Fig. 2. Constructs defined in the UeberFlow meta-model

4 UeberFlow

The meta-model introduced in Section 3 has been implemented in a dual-perspective workflow engine and modelling tool called UeberFlow. UeberFlow supports creating, editing, validation, and execution of workflow definitions specified using the UeberFlow Lang meta-model. Furthermore, the transformation of other workflow model specifications to UeberFlow Lang is supported prototypically. Subsequently, the application of UeberFlow is shown for a scenario based on a business case. The application scenario deals with the elicitation, redesign, validation and execution of a notebook ordering process. It involves three process participants: an employee, who requires a new business notebook, the head of department, who needs to accept or refuse the request, and the IT department, which is in charge of ordering the new device and the first setup before it hands over the device to the employee.

A typical process of using UeberFlow includes creating a model using the dual-perspective editor, validating the result using the UeberFlow Validation mode, and adapting the model based on the validation results. When starting from an interactional or communication-oriented perspective one could start defining all roles (i.e., subjects) involved in the process and then model the encapsulated execution flow of a selected subject, focusing on one process participant at a time. This includes the communication with the other process actors and his or her tasks in this role. Figure 5 depicts the UeberFlow Editor showing the communication-oriented perspective of the sample workflow from the employee's viewpoint.

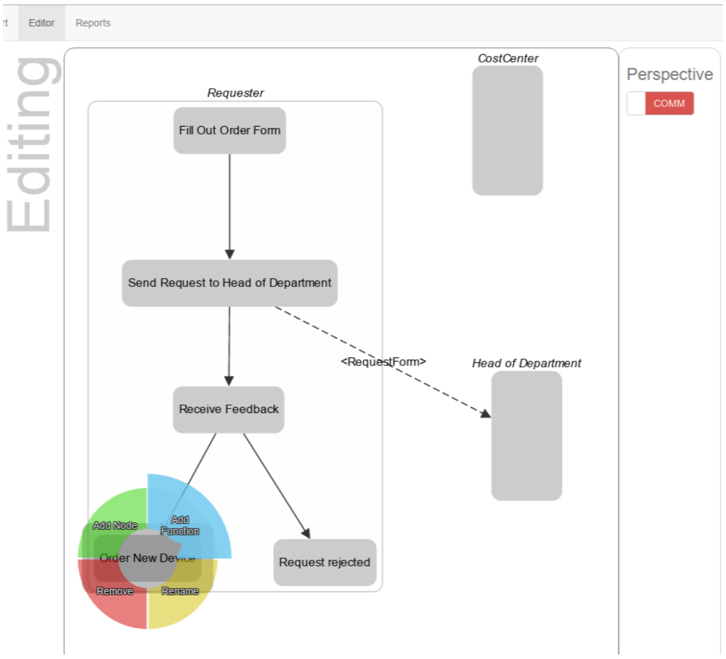

Fig. 3. The UeberFlow editor, showing the communication-oriented perspective

After having specified the behavior and communication paths of all subjects switching to the functional-perspective (cf. Figure 6) puts the focus on the overall sequence of tasks. This perspective provides an integrated view on the process compared to focusing on a single subject, whilst neglecting the communication aspects.

At any time in the modeling process, the created workflow specification can be executed in the so-called "Evaluation Mode". This execution mode allows a single user stepping through all of the tasks and communication paths, in order to check the semantic correctness of the model.

5 Conclusion & Further Research

Besides function and data flow orientation, the focus on communication and stakeholder interaction has taken hold as major perspective in the design and implementation of processes, Based on these developments, this paper argues for dual-perspective modeling support for workflows, in order to further support the creation of executable process specifications. By designing an actor-based meta-model which can be used as a foundation for providing tool support a first step towards a dual-perspective workflow specification approach has been made. The provided prototype implementing the designed meta-model and a corresponding editor showed the feasibility of the envisioned approach.

Although the potential of a dual-perspective modeling support is arguable by recent trends and studies in literature, there is yet no study providing empirical insights. Therefore, the next research step will target an empirical evaluation of the herein presented approach.

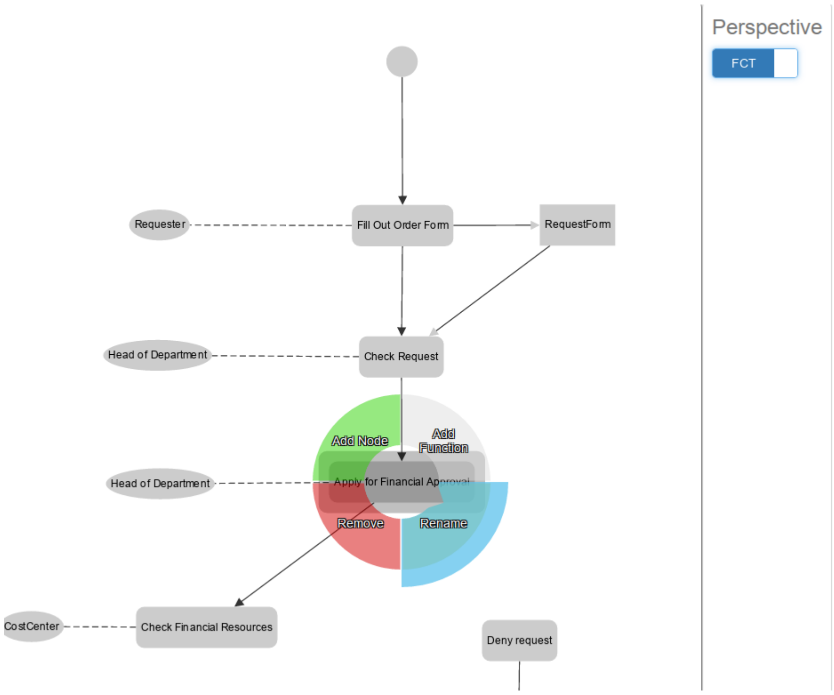

Fig. 4. Function-oriented perspective of the sample process

6 References

1. Argyropoulos, N., Mouratidis, H., & Fish, A. (2015). Towards the Derivation of Secure Business Process Designs. In *Advances in Conceptual Modeling* (pp. 248-258). Springer International Publishing.
2. Brambilla, M., Fraternali, P., & Vaca Ruiz, C. K. (2012, April). Combining social web and BPM for improving enterprise performances: the BPM4People approach to social BPM. In *Proceedings of the 21st international conference companion on World Wide Web* (pp. 223-226). ACM.

3. Duarte, F. J., Machado, R. J., & Fernandes, J. M. (2012). BIM: a methodology to transform business processes into software systems. In *Software Quality. Process Automation in Software Development* (pp. 39-58). Springer Berlin Heidelberg.
4. Fleischmann, A., Schmidt, W., Stary, C., Obermeier, S., & Börger, E. (2012). Subject-oriented business process management. Springer Publishing Company, Incorporated.
5. Gromoff, A., Kazantsev, N., Kozhevnikov, D., Ponfilenok, M., & Stavenko, Y. (2012). Newer approach to create flexible business architecture of modern enterprise. *Global Journal of Flexible Systems Management*, *13*(4), 207-215.
6. Hewitt, C., Bishop, P., & Steiger, R. (1973, August). A universal modular actor formalism for artificial intelligence. In Proceedings of the 3rd international joint conference on Artificial intelligence (pp. 235-245). Morgan Kaufmann Publishers Inc.
7. Mattos, T. D. C., Santoro, F. M., Revoredo, K., & Nunes, V. T. (2012, May). Formalizing the situation of a business process activity. In *Computer Supported Cooperative Work in Design (CSCWD), 2012 IEEE 16th International Conference on* (pp. 128-134).
8. Pinggera, J., Soffer, P., Fahland, D., Weidlich, M., Zugal, S., Weber, B., ... & Mendling, J. (2015). Styles in business process modeling: an exploration and a model. *Software & Systems Modeling*, *14*(3), 1055-1080
9. Rangiha, M. E., Comuzzi, M., & Karakostas, B. (2015). Role and Task Recommendation and Social Tagging to Enable Social Business Process Management. In *Enterprise, Business-Process and Information Systems Modeling* (pp. 68-82). LNBIP 214, Springer International Publishing.
10. Recker, J., Safrudin, N., Rosemann, M. (2012). How novices design business processes, Information Systems, Volume 37, Issue 6, September 2012, Pages 557-573, ISSN 0306-4379, http://dx.doi.org/10.1016/j.is.2011.07.001.
11. Scheer, A. W., (2000). *Architecture of Integrated Information Systems.Foundations of Enterprise Modeling*. Springer Science & Business Media, Berlin.
12. Silva, A. R., Meziani, R., Magalhaes, R., Martinho, D., Aguiar, A., & Flores, N. (2009, September). AGILIPO: Embedding social software features into business process tools. In *Business Process Management Workshops* (pp. 219-230). Springer Berlin Heidelberg.
13. Strecker, F., Gniza, R., Hollosy, Th., Schmatzer, F. (2016) Business-Actors as base components of complex and distributed software applications. In. Proceedings of the 8th International Conference on Subject-oriented Business Process Management, Erlangen, Germany, p.8, 10.1145/2882879.2882887, ACM
14. ter Hofstede, A. H., van der Aalst, W., Adams, M., & Russell, N. (Eds.). (2009). *Modern Business Process Automation: YAWL and its support environment*. Springer Science & Business Media.
15. Trkman, M., Mendling, J., & Krisper, M. (2016). Using business process models to better understand the dependencies among user stories. *Information and Software Technology*, *71*, 58-76.
16. van der Aalst, W. M., Ter Hofstede, A. H., Kiepuszewski, B., & Barros, A. P. (2003). Workflow patterns. Distributed and parallel databases, 14(1), 5-51.
17. Vernon, V. (2015). Reactive Messaging Patterns with the Actor Model: Applications and Integration in Scala and Akka. Addison-Wesley Professional, New York.
18. Wong, W. P. (2013). Business-process management: a proposed framework for future research. *Total Quality Management & Business Excellence*, *24*(5-6), 719-732.