Modeling and Executing Batch Activities in Business Processes (Extended Abstract)

Luise Pufahl

Hasso Plattner Institute, Digital Engineering Faculty, University of Potsdam
Potsdam, Germany
luise.pufahl@hpi.de

1 Motivation

Organizations strive to provide useful products and services to their customers in an efficient manner. For reaching this, the underlying business processes need to be designed in an effective and efficient manner. Business process management (BPM) provides methods and techniques for the documentation, digitalization, enactment and evaluation of business processes [19, 2]. The main artifact being produced thereby is a process model serving as a blueprint for several process instances; a process instance represents the execution of one business case [19].

A common assumption in BPM is that the process instances have an independent existence and are executed autonomously [16]. However, the consolidated execution of cases is a common phenomenon in operational business processes. For instance, in healthcare, it is more time-efficient to firstly collect a set of blood samples taken from patients to deliver them to the laboratory instead of bringing each one of them separately. Or, when calling a software service in a business process, bundling several service requests can help to reduce the number of service calls. In these use cases, batch processing [5] is applied. It allows business processes acting on a single item to bundle the execution of a group of process instances for particular activities to improve their performance.

Batch processing of products [7, 13] or services [8, 10] is deeply investigated in operations management, but an integration of its techniques in process models was not considered so far. In existing business process modeling languages, such as BPMN (Business Process Model and Notation) [11], and BPM systems, the explicit design and execution of batch activities is not yet supported. Also, the often referenced control flow patterns [1] do not discuss batch processing. Multi-instance activities [1] are presented where multiple instances are created and synchronized at the end of their execution, but they are still independently executed. Thus, batch processing is often organized manually by practitioners where the rules of the batch work are not documented and cannot be enforced.

Regarding the given problem statements, the doctoral thesis in [14] provides a comprehensive concept for integrating batch activities into business process models. The thesis followed the design science process [12] starting with a requirements analysis, developing and evaluating a designed artifact, and improving that in several cycles. In the remainder of this extended abstract, the contributions of the referenced thesis, its implications, and future research will be described.

2 Contributions

In the discussion of the related work, it was observed in the thesis that batch processing is considered in other domains, such as computer science and operations management. Whereas batch processing in computer science is used to efficiently process a large amount of data with no user interaction, discussing the design and implementation of such systems, in operations management batch processing is used to process similar products or groups of customers for being more efficient, studying the balance between reduced costs and increase in cycle time. Only a small set of research works in the BPM domain exists on the integration of batch processing in business process models. The presented solutions in those works focus on specific scenarios and lack of a complete understanding of requirements. Therefore, the thesis started with a requirements analysis based on which the design objectives were set. Then, the main contribution was presented, the concept on how to explicitly represent batch activities in business process models and to execute them automatically. Aspects of flexibility during the batch execution by different means (i.e. flexible design of the batch activation rule, user involvement strategies, and flexible batch configuration based on events) were presented and discussed. Additionally, the basic concept was extended to batch processing across multiple different process models. In the following, the main four contributions of this thesis are presented in detail:

I. Requirements Framework In the referenced doctoral thesis, a requirements framework for integrating batch processing in business processes was presented. It was developed based on related work and complemented by requirements from collected industry examples, taken from different domains. The requirements framework gives insight into the aspects which need to be considered for developing a batch processing concept for business process models. Additionally, it fosters also the comparison of existing solutions. In this thesis, the requirements framework was used to structurally compare the requirements of the collected real-world scenarios whereby two preliminary types of batch activities – *automated* and *user-involved* batch activities – were identified. Further, it was used to set the design objectives, and to compare the developed batch activity concept to other related work.

II. Batch Activity Concept The main result of the thesis is the *batch activity* concept. The concept describes the syntax of a batch activity with its batch configuration parameters which need to be specified by process designers. Specifically, we present in details the *groupedBy*-parameter for grouping instances in specific batches based on their data context and the *batch activation rule* being responsible for balancing the cost reductions with additional waiting time. Different types of activation rules were formalized, such as the *threshold rule* identified in operations management. Based on it, extensions were developed, such as the *MinMaxRule* to consider future instances. Further, it gives an operational semantics of the batch activity. It describes the life cycle of batch clusters (i.e. the actual representations of batch executions) and the interaction of batch clusters with process instances and the activity resources (i.e. task performers or services). The feasibility of the concept was shown by a prototypical implementation in Camunda – an existing, open-source BPM system. The application of the batch activity to different use cases shows that process cost can be reduced with acceptable or even positive influence on the cycle time if a suitable batch activation rule is selected. For supporting *automated* as well as *user-involved* batch activities, different levels of user involvement and the way they might be realized were presented in this thesis.

III. Integration of Flexibility in Batch Activities The thesis showed that flexibility aspects of batch activities are not discussed by existing related work in the BPM domain. In the work of this thesis, flexibility for batch activities was provided by different means: First of all, batch activation rules, which are using Event-Condition-Action rules [4], can be flexibly designed, such that also an immediate execution of special cases is allowed. The thesis presented here the so-called *FastTrackRule*. Further, the presented user involvement strategies allow task performers to react dynamically on changes and exceptions in the process environment (e.g. by starting batch clusters when needed, re-assigning instances to another batch cluster, or waiting for specific future instances). Finally, the thesis presents a concept based on event processing techniques to allow a flexible batch configuration. In this, batch adjustment rules are defined by process designers which specify for which event type which type of batch clusters need to be adapted and how. The application of the batch adjustment rules to a healthcare scenario in a simulated environment showed that they help to compensate the losses caused by the exceptional behavior in this use case.

IV. Multi-process Batch Processing Concept This thesis could show that batch processing across several different business processes is useful and was not yet discussed by existing related work. The proposed multi-process batch processing concept allows a centrally defined batch specification in an object lifecycle. Object lifecycles complement process models and describe allowed actions of business processes on data artifacts across the process-model boundaries. The basic concept of *batch transitions* was additionally extended in this thesis to multiply connected batch transitions (to allow also batching in process fragments) and to multiple similar batch transitions (to allow batching of activities having different data inputs but producing the same output). The requirement of *optional* batch processing in a multi-process setting was enabled by activating the batch processing only when similar activity instances are detected. Further, the concept includes a user approval where the task performer has to accept identified batches by the system. The feasibility of the concept was shown by a prototypical implementation in Camunda, an existing open-source BPM system.

3 Implications and Future Research

The results of the referenced thesis have relevant implications for practice and the BPM research. With regards to practice, it showed that batch work cannot be specified and automated with the current process modeling languages and technology. Therefore, a new process modeling element was provided, the *batch activity* that allows a specification of batch work with the help of several configuration items. Additional to that, this thesis presented a BPMN process simulator *Scylla* [15] extended with batch activities which can be used to validate a defined batch activity and its configuration to identify the most useful one. The extension of Camunda by the batch activity provides for BPM system providers an example how this new modeling element can be integrated into their system. Also, the given operational semantics in the thesis can support such efforts.

With regards to BPM research, the referenced doctoral thesis extended and advanced the first attempts of batch activities (e.g. in [5, 17]) with a comprehensive requirement analysis and a concept for batch activities building on this. The proposed batch activity concept stimulated or influenced further research, such as batch process mining (e.g. in [6]), new types of batch activities (e.g. in [9]), and that batching of instances is a relevant instance-spanning constraints [3]. The flexibility concept for batch activities shows how runtime changes can be handled in a process with the help of event processing; thereby it contributes to the area of flexible process management. The extension of batch activities to batching over several business processes showed that object life cycles are useful to define central aspects being relevant for several business processes.

Future work with regards to batch activities can be focused on the following directions: Batch activities are currently identified and configured based on expert knowledge by the process designers. The developed BPMN process simulator *Scylla* being able to simulate business process with batch activities could be extended to give recommendations for an optimal batch activity configuration. Another possibility, if execution logs exist, is to use process mining techniques for an automated discovery of batch activities from those event logs as proposed by [6, 18]. However, those techniques are not capable to give recommendations for batch activities and to mine a complete batch configuration, such that more research work in this direction is needed. Further, an application of the batch processing concepts presented in this thesis (e.g., in a user study or in a technical action research [20]) will give more insights into the usability and usefulness of the presented concepts. These might also lead to new requirements which extend or detail the introduced requirements framework. With regards to flexibility of batch activities, this thesis presented different means to support the batch activity adaptation, but the ability to allow variability of batch activities was not discussed yet. In future, a concept to allow variability of the batch activity configurations can be developed which might be supported by the batch processing discovery techniques. Going a step further, the current batch activity concept which is a design and implementation concept could be used as a basis for a dynamically batch processing concept where batches can be created at any activity if certain conditions are fulfilled.

References

1. van der Aalst, W.M., ter Hofstede, A.H., Kiepuszewski, B., Barros, A.P.: Workflow patterns. Distributed and Parallel Databases **14**(1), 5–51 (2003)
2. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of business process management, vol. 1. Springer (2013)
3. Fdhila, W., Gall, M., Rinderle-Ma, S., Mangler, J., Indiono, C.: Classification and formalization of instance-spanning constraints in process-driven applications. In: International Conference on Business Process Management (BPM). pp. 348–364. Springer (2016)
4. Laliwala, Z., Khosla, R., Majumdar, P., Chaudhary, S.: Semantic and rules based event-driven dynamic web services composition for automation of business processes. In: Services Computing Workshops (SCW). pp. 175–182. IEEE (2006)
5. Liu, J., Hu, J.: Dynamic batch processing in workflows: Model and implementation. Future Generation Computer Systems **23**(3), 338–347 (2007)
6. Martin, N., Swennen, M., Depaire, B., Jans, M., Caris, A., Vanhoof, K.: Retrieving batch organisation of work insights from event logs. Decision Support Systems (2017)
7. Mathirajan, M., Sivakumar, A.I.: A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. The International Journal of Advanced Manufacturing Technology **29**(9-10), 990–1001 (2006)
8. Medhi, J.: Stochastic models in queueing theory. Academic Press (2002)
9. Natschläger, C., Bögl, A., Geist, V., Biró, M.: Optimizing resource utilization by combining activities across process instances. In: European Conference on Software Process Improvement. pp. 155–167. Springer (2015)
10. Neuts, M.: A general class of bulk queues with poisson input. The Annals of Mathematical Statistics **38**(3), 759–770 (1967)
11. OMG: Business Process Model and Notation (BPMN), V. 2.0 (2011)
12. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Journal of Management Information Systems **24**(3), 45–77 (2007)
13. Potts, C.N., Kovalyov, M.Y.: Scheduling with batching: A review. European journal of operational research **120**(2), 228–249 (2000)
14. Pufahl, L.: Modeling and enacting batch activities in business processes. Ph.D. thesis, University of Potsdam (2018)
15. Pufahl, L., Wong, T.Y., Weske, M.: Design of an Extensible BPMN Process Simulator. In: Business Process Management Workshops (2017)
16. 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 (CAiSE). pp. 216–232. Springer (2005)
17. Sadiq, S., Orlowska, M., Sadiq, W., Schulz, K.: When workflows will not deliver: The case of contradicting work practice. International Conference on Business Information Systems (BIS) **5**, 69–84 (2005)
18. Wen, Y., Chen, Z., Liu, J., Chen, J.: Mining batch processing workflow models from event logs. Concurrency and Computation: Practice and Experience **25**(13), 1928–1942 (2013)
19. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Second Edition. Springer (2012)
20. Wieringa, R.J.: Design science methodology for information systems and software engineering. Springer (2014)