Integrating Declarative and Imperative Process Modeling Paradigms in the Age of Generative AI

Timotheus Kampik, Gregor Berg, and David Eickhoff

Abstract: This brief paper summarizes a talk introducing and discussing the notion of process atoms, small facts or queries, each describing an organizationally relevant property or constraint of a business process that cannot be further split without losing its business meaning. An example of a process atom is: “only if an order with a purchase amount greater than 10,000€ is requested, management approval has to take place afterwards” (more abstractly: “only if A then eventually B”). As process atoms are executable as queries on data and allow for dynamic contextualization across process and organizational scopes, they complement and augment traditional process models, such as BPMN diagrams, particularly in the age of data-driven process analysis and generative AI-created process content.

Keywords: Modeling Paradigms, Process Modeling, Generative AI

Process Atoms

In the era of model-driven development, when process execution engines were expected to be the ultimate bridges between business and IT, the focus was on imperative models – the corresponding hope was that business users would easily maintain these models, alleviating the traditional communication overhead at the business-IT interface [CT12]. However, the dream of business user-friendly model-driven development of information systems has not fully materialized: few organizations manage to continuously practice model-driven development at a high maturity level, and for the ones who do, this incurs substantial costs and requires high expertise levels, e.g., in technical modeling skills among business (process) analysts. Still, imperative models are important, and frequently used to provide a broad, approximate specification of an actual or future state process, often for human consumption or social proof of compliance and not for direct deployment. In contrast, declarative approaches [CM22] play a crucial role in the data-driven era of business process management, most notably in the form of query languages [Po22]. Here, an industry assumption is that imperative models are – at least without further manipulation – not particularly useful for data analysis: they are often overly specific (too strict) and not easily

1 SAP Signavio, George-Stephenson-Str. 7-13, 10557 Berlin, Germany, timotheus.kampik@sap.com, gregor.berg@sap.com; david.eickhoff@sap.com

2 Note that an extended (yet more colloquial) version of this short paper is available as a blog post at https://community.sap.com/t/technology-blogs-by-sap/the-role-of-process-atoms-in-modern-business-process-management/b-13578185.

This work is licensed under Creative Commons Attribution 4.0 International License http://creativecommons.org/licenses/by/4.0/. https://doi.org/10.18420/modellierung2024-ws-005
Integrating Declarative and Imperative Process Modeling Paradigms in the Age of Generative AI

Executable by symbolic data management systems, such as traditional relational databases or OLAP-style data warehouses that utilize structured query languages. Crucially, this means that imperative models are not organically suitable for cross-organizational comparative analysis (e.g., in a benchmarking context).

Now, considering that imperative models are the lingua franca of human-centric BPM, whereas declarative models in the form of queries (or abstractions of queries) are what typically enables data-driven BPM, the question is how the two approaches can be fused in sufficiently elegant, yet practicable manner: imperative models provide a broad, high-level description that is somewhat intuitive for human experts, but typically neither precise on a technical process data-level, nor easily portable to different contexts, e.g., using generative AI approaches; this shortcoming can be addressed by augmenting imperative models with the most crucial facts about a process in a manner that is precise yet re-usable across process contexts and organizations. We claim that such a fusion can be achieved by introducing process atoms, declarative facts about a business process that are minimal in the sense that splitting them further would obfuscate their business meaning (analogously to the notion of a propositional atom in logics). Conceptually, process atoms can be thought of as constraints, i.e., as queries that are either satisfied or violated on the level of a specific process instance. From a practical perspective, process atom types can be defined as query templates for declarative process query languages and then be instantiated based on domain knowledge that can be formalized as queries for a specific process data analysis use case.

Given the role we expect process atoms to play in generative AI-augmented business process management [Du23], we introduce a series of open questions that we think are worthy of academic research attention:

1. How to best model process atoms at an abstraction level that is both technically effective and business user friendly?
2. How to best manage process atoms in process model repositories?
3. How to best integrate process atoms with existing process modeling approaches such as Business Process Model and Notation (BPMN)?

We hope that with the increased adoption of industry-scale process querying solutions, as well as with the application of generative AI to process querying, researchers and practitioners alike will set out to answer these questions.

---

3 Let us highlight that the general idea of fusing declarative and imperative modeling paradigms is not new [Ab20]; however, we claim it is understudied in the context of industry-scale process modeling and mining.

4 The term process atom is based on the notion of a propositional atom, i.e., our terminology is consistent with terminology used in computational logic.
References


