delta-BPMN: an Operational Framework for Verifiable Data-Aware Business Processes

Silvio Ghilardi [1], Alessandro Gianola [2], Marco Montali [2], Andrey Rivkin [2]

[1] Università degli Studi di Milano, Milan, Italy
[2] Free University of Bozen-Bolzano, Bolzano, Italy
silvio.ghilardi@unimi.it, {gianola,montali,rivkin}@inf.unibz.it

Overview

The integration between data and processes is a long-standing challenge in information systems engineering (Hull 2008; Reichert 2012; Steinau et al. 2019). The increasing recognition of the need for this integration, both at conceptual and system levels, raises a new demand in standard-friendly, verifiable data-aware process modeling languages. This problem comes with a large number of difficulties, which can be summarized by the fact that all the proposals in the literature have been largely focusing on either uncharted approaches or conceptual proposals that would lack in tool support. In order to deal with both these limitations, in (Ghilardi et al. 2021) we propose delta-BPMN – a verifiable operational framework for data-aware processes.

We now discuss in more detail why this problem is challenging. On the one hand, the adopted model for the representing this integration should be expressive enough to represent complex processes where data influence how the process control-flow routes cases, while the process tasks inspect and manipulate data. On the other hand, such expressiveness has to be suitably controlled toward enabling verification, execution, monitoring, and mining of such multi-perspective models. A third, orthogonal dimension concerns the choice of modeling constructs, which often depart from those offered by process and data modeling standards such as BPMN and SQL, in turn hampering the adoption of the resulting frameworks.

These dimensions can be recognized in their full complexity when it comes to the *verification* of the resulting integrated models (Calvanese, De Giacomo, and Montali 2013; Deutsch et al. 2018). Verification is of particular importance in this spectrum, as even data and process models that appear correct when analyzed in isolation may lead to errors once integrated (Polyvyanyy et al. 2019). "Verifiability" of models is thus typically obtained by using abstract languages that do not adhere to well-established standards when it comes to the data and/or process component: either the control-flow backbone of the process is captured using Petri nets or other mathematical formalisms for dynamic systems that cannot be directly understood using front-end notations such as BPMN, or the data manipulation part relies on abstract, logical operations that cannot be straightforwardly represented in concrete data manipulation languages such as SQL. At the same time, the repertoire of constructs used to model data-aware processes cannot cover these languages in their full generality, as verification becomes immediately undecidable if they are not suitably controlled (Calvanese, De Giacomo, and Montali 2013). A last crucial issue is that the vast majority of the literature provides foundational results that do not directly translate into effective verification tools.

These limitations are tackled by delta-BPMN, an operational framework, presented in (Ghilardi et al. 2021), at once supporting modeling and verification of BPMN enriched with data management capabilities. delta-BPMN comes with a threefold contribution. First, we introduce the front-end data modeling and manipulation language PDMML, supported by delta-BPMN, which instantiates the data-related aspects of the abstract modeling language studied in (Calvanese et al. 2019a) by using a SQL-based dialect to represent as well as manipulate volatile and persistent data, and show how it can be embedded into a (block-structured) fragment of BPMN that captures the process backbone. The features of PDMML are based on requirements for concrete, verifiable data-aware process modeling languages distilled from the literature. Second, we show how the delta-BPMN front-end can be realized in actual business process management systems, considering in particular Camunda[1], one of the most popular BPMN environments. Third, we report on the implementation of a translator that, building on the encoding rules abstractly defined in (Calvanese et al. 2019a), takes a delta-BPMN model created in Camunda and transforms it into the syntax of MCMT[2], a state-of-the-art SMT-based model checker for infinite-state systems that can then be used for verification of data-aware processes (Calvanese et al. 2020).

Requirement Analysis and Related Work

Since our focus is on verification, we circumscribe the relevant works to those dealing with the formal analysis of data-aware processes. As pointed out in the introduction, this is also crucial because the choice of language constructs is affected by the task one needs to solve.

A second important point is that the vast majority of the contributions in this line of research provide foundational results, but do not come with corresponding operational tools

[1] https://camunda.com
[2] http://users.mat.unimi.it/users/ghilardi/mcmt/

for verification. Hence, all in all, *we consider in this research only those approaches for the integration of data and processes that come with verification tool support*: VERIFAS (Li, Deutsch, and Vianu 2017), BAUML (Estañol, Sancho, and Teniente 2018), ISML (Polyvyanyy et al. 2019), dapSL (Calvanese et al. 2019b), and the delta-BPMN approach, relying on the foundational basis in (Calvanese et al. 2019a).

We use these approaches to distill a series of important requirements on languages for verifiable data-aware processes, indicating which provide full (+), partial (+/−), or no support (−) for that requirement. The first two requirements concern verifiability.

RQ 1. The language should be operationally verifiable with a tool. ◁

While the approaches above all come with an operational counterpart for verification, there are huge differences in how this support is provided. VERIFAS comes with an embedded, ad-hoc verification tool (+) that supports the model checking of properties expressed in a *fragment of first-order LTL*. BAUML encodes verification into a form of first-order satisfiability checking over the flow of time (+), defining a fixed set of *test cases* expressing properties to be checked as derived predicates. ISML relies on state-space construction techniques for Colored Petri nets, but in doing so it assumes that the data domains are all bounded (+/−); no specific verification language is defined, leaving to the user the decision on how to explore the state space. dapSL relies instead on an ad-hoc state-space construction that, under suitable restrictions, is guaranteed to faithfully represent in a finite-state way the infinite state space induced by the data-aware process; however, no additional techniques are defined to explore the state space or check temporal properties of interest (+/−). Finally, delta-BPMN encodes verification of *(data-aware) safety properties* into the state-of-the-art MCMT model checker (+).

The second requirement concerns the analysis of key meta-properties (such as completeness and termination) of the algorithmic techniques used for verification. This is crucial since, in general, verifying data-aware processes is highly undecidable (Calvanese, De Giacomo, and Montali 2013; Deutsch et al. 2018).

RQ 2. The verification techniques come with an analysis of key meta-properties. ◁

Since ISML and dapSL do not come with specific algorithmic techniques for verification, no such analysis is provided there (−). BAUML relies on first-order satisfiability techniques that come with semi-decidability guarantees. In (Estañol, Sancho, and Teniente 2018), it is claimed that for a certain class of state-bounded artifact systems, verification terminates; however, this is not guaranteed, as for that class only decidability of verification is known, not that the specifically employed satisfiability algorithm terminates (+/−). VERIFAS comes with a deep, foundational study on the boundaries of decidability of verification (Deutsch, Li, and Vianu 2016); the study identifies classes of data-aware processes for which finite-state abstractions can be constructed, guaranteeing termination of the verifier when analyzing such classes (+). Finally, delta-BPMN relies on the foundational DAB framework (Calvanese et al. 2019a), where soundness, completeness, termination of the algorithmic technique implemented in MCMT are extensively studied (+).

The third crucial requirement is about the type of language adopted, and whether it adheres to accepted standards or is instead rather ad-hoc.

RQ 3. The language relies on well-assessed standards for processes and data. ◁

Recall that, to carry out verification, the features supported by the language need to be carefully controlled. So we do not assess approaches based on their coverage of constructs, but rather focus on which notations they employ. On the one hand, approaches like VERIFAS adopt a language inspired by artifact-centric models but defined in an abstract, mathematical syntax (−). At the other end of the spectrum, BAUML comes with a combination of UML/OCL-based models to specify the various process components (+). In between we find the other proposals (+/−): ISML relies on Petri nets and employs data definition and manipulation languages defined in an ad-hoc way; dapSL instead defines the control-flow implicitly via condition-action rules, and uses a language grounded in the SQL standard for querying and updating the data. delta-BPMN relies on a combination of (block-structured) BPMN and SQL for data manipulation; while standard SQL is employed for data queries and updates, the language has to be extended with some ad-hoc constructs when it comes to actions and (user) inputs (+/−).

In data-aware processes, it is essential to capture the fact that while the process is executed, new data can be acquired.

RQ 4. The language supports the injection of data into the process by the external environment. ◁

All of the listed approaches agree on the need of equipping the language with mechanisms to inject data from the external environment. VERIFAS and BAUML allow one to non-deterministically assign values from value domains to (special) variables, ISML extends this functionality with an ability to guarantee that assigned values are globally fresh (but then it works by assuming a fixed finite domain for such fresh input), whereas dapSL supports all such functionalities using a language of service calls. In delta-BPMN we adopt a data injection approach similar to the one used in VERIFAS.

When executing process cases, one typically distinguishes at least two types of data: volatile data attached to the case itself, and persistent data that may be accessed and updated by different cases at once. This leads to our last requirement.

RQ 5. The language distinguishes volatile and persistent data elements. ◁

While BAUML, VERIFAS, and DAB natively provide distinct notions for case variables and underlying persistent data (+), ISML models conceptually account for token data and separate facts, but such facts are not stored in a persistent storage (+/−), while dapSL models all data as tuples of a relational database (−).

References

Calvanese, D.; De Giacomo, G.; and Montali, M. 2013. Foundations of Data Aware Process Analysis: A Database Theory Perspective. In *Proc. of PODS*.

Calvanese, D.; Ghilardi, S.; Gianola, A.; Montali, M.; and Rivkin, A. 2019a. Formal Modeling and SMT-Based Parameterized Verification of Data-Aware BPMN. In *Proc. of BPM*, LNCS, 157–175. Springer.

Calvanese, D.; Ghilardi, S.; Gianola, A.; Montali, M.; and Rivkin, A. 2020. SMT-based Verification of Data-Aware Processes: a Model-Theoretic Approach. *Mathematical Structures in Computer Science*, 30(3): 271–313.

Calvanese, D.; Montali, M.; Patrizi, F.; and Rivkin, A. 2019b. Modeling and In-Database Management of Relational, Data-Aware Processes. In *Proc. of CAiSE*, LNCS, 328–345. Springer.

Deutsch, A.; Hull, R.; Li, Y.; and Vianu, V. 2018. Automatic verification of database-centric systems. *ACM SIGLOG News*, 5(2): 37–56.

Deutsch, A.; Li, Y.; and Vianu, V. 2016. Verification of Hierarchical Artifact Systems. In *Proc. of PODS*, 179–194. ACM.

Estañol, M.; Sancho, M.; and Teniente, E. 2018. Ensuring the semantic correctness of a BAUML artifact-centric BPM. *Inf. Softw. Technol.*, 93: 147–162.

Ghilardi, S.; Gianola, A.; Montali, M.; and Rivkin, A. 2021. Delta-BPMN: A Concrete Language and Verifier for Data-Aware BPMN. In *Proc. of BPM 2021*, volume 12875 of *Lecture Notes in Computer Science*, 179–196. Springer.

Hull, R. 2008. Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges. In *Proc. of OTM*, volume 5332 of *LNCS*. Springer.

Li, Y.; Deutsch, A.; and Vianu, V. 2017. VERIFAS: A Practical Verifier for Artifact Systems. *PVLDB*, 11(3).

Polyvyanyy, A.; van der Werf, J. M. E. M.; Overbeek, S.; and Brouwers, R. 2019. Information Systems Modeling: Language, Verification, and Tool Support. In *Proc. of CAiSE*, LNCS, 194–212. Springer.

Reichert, M. 2012. Process and Data: Two Sides of the Same Coin? In *Proc. of OTM*, volume 7565 of *LNCS*. Springer.

Steinau, S.; Marrella, A.; Andrews, K.; Leotta, F.; Mecella, M.; and Reichert, M. 2019. DALEC: a framework for the systematic evaluation of data-centric approaches to process management software. *Softw. Syst. Model.*, 18(4): 2679–2716.