Modeling, Enactment and Verification of Data-Aware Processes

Andrey Rivkin

Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy
rivkin@inf.unibz.it

CEUR-WS.org/Vol-2420/paperDA9.pdf

In contemporary organizations, the integrated management of business processes (BPs) and master data (MD) is being increasingly considered as a fundamental problem, both by academia and industry. From the practical point of view, it has been widely recognized that the lack of systematic synergy between BPs and MD causes fragmentation and redundancies in the organizational structure and its underlying IT solutions, with experts and tools solely centered around data, and others only focusing on process management [16, 11, 15]. This isolation falls short, especially when it comes to knowledge-intensive and human-empowered processes [12, 14].

Modern management systems (BPMSs), such as Bizagi BPM, Bonita BPM, Activiti, Camunda, and YAWL, actually provide clean conceptualizations for the process control flow as well as the "touching joints" between control flow and data: *(i)* process instances (or cases) carry local data, *(ii)* a database (DB) backend is typically used to store global, persistent data, *(iii)* the decision logic queries local and persistent data to choose which path to select among multiple alternatives, *(iv)* the task logic defines how to update local and persistent data. However, as argued in [9], no well-established approach exists to express the decision and task logic, which is in fact handled in an ad-hoc way, usually combining tool-specific languages with general purpose programming languages such as Java. Like that the interaction of the process and its data becomes a sort of "procedural attachment" that is exploited during the process enactment, but that is not conceptually well-understood [6]. As an effect, the verification tasks offered by such systems become either disabled when data are present, or produce misleading answers, since they do not take into account that the presence of data subtly affects the behaviors described by the process [9]. For example, seemingly concurrent behavior in the process may be in fact sequenced due to the presence of data constraints, implicitly inducing an order on the allowed data updates. More generally, non-executable paths and deadlocks may emerge only when the interplay between the process and its data is considered.

Foundational research witnesses a similar separation, with non-interacting areas of research either focused on data management or dynamic concurrent systems, with DB theory and Petri net theory being the two most prominent representatives of each field. Over the years, both fields entered into the problem of combining data and processes, with quite complementary approaches.

A first series of approaches stems from Petri nets, the reference formalism to represent the control-flow of BPs. All such models are more or less directly inspired by Colored Petri nets (CPNs) [13, 2], where colors abstractly account for (typed) data values, and where the control threads (i.e., tokens) traversing the net carry colors. Verification in this setting is tackled by severely restricting the contribution of data: colors are required to come from finite domains, thus realizing a form of a-priori propositionalization of the data, or by limiting the way tokens can carry data. The latter has led to the discovery of several CPN fragments that are amenable to formal analysis even in the case of infinite color domains, ranging from nets where tokens carry single data values (as in data- and ν-nets), to nets where tokens are associated to more complex data structures such as nested relations, nested terms, or XML documents. However, the common limitation of all such approaches is that data are still subsidiary to the control-flow dimension: data elements are "locally" attached to tokens, while no native support for global, persistent relational data is provided. In this light, CPNs naturally only support cases and case data through the abstraction of colored tokens [18], but does not let itself to adopting CRUD operations over DBs, which are typical for enterprise information systems. For this reason, they are unable to impact contemporary BPMSs, which, as argued above, all support the explicit linkage of BPs and an underlying persistent DB [9].

The second group of foundational approaches to data-aware processes has emerged at the intersection of database theory, formal methods and conceptual modeling, and specularly mirrors the advantages and lacks of CPN-based solutions. Such proposals go under the umbrella term of data-centric approaches [6], and gained momentum during the last decade, in particular due to the development of the business artifact paradigm [7], leading to concrete languages and implementations [8, 14]. The common denominator of all such approaches is that processes are centered around an explicit, persistent data component maintaining information about the domain of interest, and possibly capturing also its semantics in terms of classes, relations, and constraints. Atomic tasks induce CRUD operations over the data component, in turn supporting the evolution of the MD maintained therein. Proposals then differ in terms of the adopted data model (e.g., relational, tree-shaped, graph-structured), and on the nature of information (e.g., whether it is complete or not). The main downside of data-centric process models is that they disregard an explicit representation of how tasks have to be sequenced over time, only implicitly representing the control flow via (event-)condition-action rules [10, 8, 5]. Hence, they are too distant from contemporary BPMSs, which all rely on Petri net-inspired languages to define the process control flow.

Contributions. In this work, we aim at attacking two central challenges present in the contemporary approaches for integrating data and processes: *lack of concretisation* and *lack of cross-fertilization and balance*. The first one focuses on the gap between existing foundational frameworks and their incorporation into actual systems for modeling, enactment, and analysis. The second challenge concerns the lack of *(i)* interaction between different approaches that are paradigmatically data-aware as well as *(ii)* frameworks and methodologies in which integrated models for data and processes can be understood in terms of already well-established formalisms and concepts.

We tackle the ***first challenge*** by focusing on the framework of *data-centric dynamic systems* (DCDSs) [5], coming from the family of data-centric approaches. DCDSs tackle modeling and verification of data-aware processes running over a full-fledged relational DB with integrity constraints, that, however, operates over a single abstract object domain. On top of this relational DB, a process modifies and evolves the data by executing (update) actions, possibly injecting external data retrieved through service calls.

From the modeling point of view, we enrich DCDSs towards end-user oriented functionalities, obtaining the concrete setting of Relational Dynamic Systems (RDSs). Differently from DCDSs, RDSs support concrete datatypes (and their corresponding rigid predicates) as well as so-called data acquisition functions. The latter formally account for different types of external sources of data to be injected into the system, such as interaction with external web services, user forms, generation of fresh values, and numerical computations. We show that the decidability results obtained for DCDSs for model checking properties expressed in first-order extensions of μ-calculus carry over to RDSs.

From the conceptual point of view, we propose a general, pristine approach to model data-aware processes (DAPs) operating on top of the standard relational technology. Specifically, we propose a language called $\mathbb{dapSL}$ that is based on the formal framework of RDSs and that incorporates SQL for conceptual modeling of control-flow conditions and of persistent data updates with external inputs. We then show how $\mathbb{dapSL}$ can be automatically translated into a concrete procedural SQL dialect, consequently providing in-database process execution support.

From the implementation point of view, we introduce daphne, an engine for RDSs, whose back-end consists of a relational storage with corresponding stored procedures to manage the action-induced updates, and whose JAVA front-end provides APIs and functionalities to inspect the current state of the process and its underlying data, as well as to interact with different concrete systems for acquiring external data. daphne also offers the basis for explicit model checking of RDSs, which is realized by constructing a transition system capturing the execution semantics of the input RDSs that is succinctly represented in the DB using an improved variant of the abstraction technique originally developed for DCDS in [4, 5]. All in all, daphne provides, at once, the basis for modeling, enactment, and verification of DAPs, *all applied on the same input model.*

As for the ***second challenge***, we create a bridge between foundational frameworks for data and processes, and corresponding models that are closer to actual systems and implementations. Here we try to focus separately on two alternative approaches, namely data-centric and process-centric approaches, and work on the touching joints between them. Then we propose a combined model aiming at a suitable equilibrium between these two modeling styles.

Petri nets with data. So far, our focus was on DCDSs – a framework respecting the data-centric paradigm. While looking for touching joints among numerous frameworks from the process-centric camp, we opted for the formalism of Petri nets with name creation and management (ν-PNs) [17]. ν-PNs have been recently introduced as an expressive model for dynamic (distributed) systems, whose dynamics are determined not only by how tokens flow in the system, but also by the pure names (or abstract objects) they carry. On the one hand, this extension makes the resulting nets strictly more expressive than P/T nets. On the other hand, fundamental properties like coverability, termination and boundedness are decidable for ν-PNs. The last property together with the fresh name generation make ν-PNs an interesting formalism that could be compared in different ways against DCDSs. We first study the problem of formal verification of ν-PNs against data-aware temporal logics (namely, a first-order variant of μ-calculus) and obtain its decidability via a translation from ν-PNs to DCDSs. Our approach shows that interesting, novel results can be obtained by cross-fertilizing the research areas of formal methods for concurrent systems and that of foundations of data-aware processes, which have not been extensively related so far. Second, we show how ν-PNs can be used to enrich the well-known paradigm of workflow nets with explicit process instances and global resources, and how a suitably revised notion of soundness [1] can be formalized and checked with our approach. The devised extension can be used for modeling resource-aware workflows with multiple cases, where the number of resources is bounded a-priori. We also argue, that ν-PNs can be used to model the life-cycle of one case in isolation by implicitly incorporating the data flow into the model. While such an approach is not novel (e.g., in [3] places represent the data object states and transitions represent the activities performed on it), ν-PNs also allow for generating fresh data that can be used to abstractly account for, e.g., user input.

DB-nets. To reconcile data-centric and process-centric approaches, we obtain a novel, well-balanced formalism of DB-nets. In a DB-net: *(i)* MD are represented using full-fledged relational databases with constraints; *(ii)* the process logic is captured using a CPN extended with special places whose content corresponds to a view on top of the underlying database; *(iii)* the task logic conceptually defines how the underlying database is updated. We rigorously describe the abstractions offered by DB-nets, and formalize their execution semantics. Interestingly, the execution semantics of DB-nets is represented in terms of (possibly) infinite labeled transition systems, where each state simultaneously accounts for a net marking and a snapshot of the underlying DB. We then study different notions of boundedness that apply both to the DB and the net, and show that, using an encoding to DCDSs, analogous to the one used in the case of ν-PNs, one can prove decidability of reachability. To make the modeling and execution of DB-nets operational, we propose a prototype called *RelCPN* and that is based on CPN Tools. *RelCPN* uses CPN Tools to represent the process logic and Java extensions to make the net communicate with the outer world by allowing it to acquire possibly fresh data from external services and manipulate the underlying database using net elements. We also demonstrate how a very expressive class of DB-nets can be encoded into standard CPNs extended with priorities. This result is of particular interest since 1) it for the first time shows how full-fledged DBs with corresponding data manipulation operations can be encoded using Petri nets, and 2) the encoding formally justifies the CPN-based representation of DB-nets employed by *RelCPN*. At last, this encoding allows for the direct representation of DB-nets directly in CPN Tools and use it to simulate and analyse this DB-net models.

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.P., Stahl, C.: Modeling Business Processes - A Petri Net-Oriented Approach. Cooperative Information Systems series, MIT Press (2011)
3. Awad, A., Decker, G., Lohmann, N.: Diagnosing and repairing data anomalies in process models. In: Business Process Management Workshops. pp. 5–16. Springer Berlin Heidelberg (2010)
4. Bagheri Hariri, B., Calvanese, D., De Giacomo, G., Deutsch, A., Montali, M.: Verification of relational data-centric dynamic systems with external services. CoRR Technical Report arXiv:1203.0024, arXiv.org e-Print archive (2012)
5. Bagheri Hariri, B., Calvanese, D., De Giacomo, G., Deutsch, A., Montali, M.: Verification of relational data-centric dynamic systems with external services. In: Proc. of PODS. pp. 163–174. ACM (2013)
6. Calvanese, D., De Giacomo, G., Montali, M.: Foundations of data aware process analysis: A database theory perspective. In: Proc. of PODS (2013)
7. Cohn, D., Hull, R.: Business artifacts: A data-centric approach to modeling business operations and processes. IEEE Data Eng. Bull. **32**(3), 3–9 (2009)
8. Damaggio, E., Hull, R., Vaculín, R.: On the equivalence of incremental and fixpoint semantics for business artifacts with guard-stage-milestone lifecycles. Information Systems **38**(4), 561–584 (2013)
9. De Masellis, R., Di Francescomarino, C., Ghidini, C., Montali, M., Tessaris, S.: Add data into business process verification: Bridging the gap between theory and practice. In: Proc. of AAAI. pp. 1091–1099 (2017)
10. Deutsch, A., Hull, R., Patrizi, F., Vianu, V.: Automatic verification of data-centric business processes. In: Proc. of ICDT. pp. 252–267 (2009)
11. Dumas, M.: On the convergence of data and process engineering. In: Proc. of ABDIS. LNCS, Springer (2011)
12. Hull, R.: Artifact-centric business process models: Brief survey of research results and challenges. In: Proc. of ODBASE (2008)
13. Jensen, K., Kristensen, L.M.: Coloured Petri Nets - Modelling and Validation of Concurrent Systems. Springer (2009)
14. Künzle, V., Weber, B., Reichert, M.: Object-aware business processes: Fundamental requirements and their support in existing approaches. Int. J. of Information System Modeling and Design **2**(2), 19–46 (2011)
15. Reichert, M.: Process and data: Two sides of the same coin? In: Proc. of OTM (2012)
16. Richardson, C.: Warning: Don't assume your business processes use master data. In: Proc. of BPM. LNCS, vol. 6336, pp. 11–12. Springer (2010)
17. Rosa-Velardo, F., de Frutos-Escrig, D.: Decidability and complexity of petri nets with unordered data. Theor. Comput. Sci. **412**(34), 4439–4451 (2011)
18. Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow data patterns: Identification, representation and tool support. In: Proc. of ER. LNCS, vol. 3716, pp. 353–368. Springer (2005)